

AD-A243 651



AFIT/GE/ENG/91D-51

DTIC  
ELECTE  
DEC 26 1991  
S C D

①

*VERIFICATION AND VALIDATION OF THE  
SIMULATED RADAR IMAGE (SRIM) CODE  
RADAR CROSS SECTION PREDICTIONS*

THESIS

Dale Arden Stanley  
Captain, USAF

AFIT/GE/ENG/91D

**91-19052**



Approved for public release; distribution unlimited

**91 12 24 081**



INSTRUCTIONS FOR COMPLETING SF 298

1. This form is to be completed by the contractor or subcontractor who is responsible for the work covered by the contract. It is to be submitted to the Contracting Officer's Representative (COR) or the Contracting Officer (CO) at the time of the final invoice or payment request. It is to be submitted to the COR or CO at the time of the final invoice or payment request.

*VERIFICATION AND VALIDATION OF THE  
SIMULATED RADAR IMAGE (SRIM) CODE  
RADAR CROSS SECTION PREDICTIONS*

THESIS

Presented to the Faculty of the School of Engineering  
of the Air Force Institute of Technology  
Air University  
In Partial Fulfillment of the  
Requirements for the Degree of  
Master of Science in Electrical Engineering

Dale A. Stanley  
Captain, USAF

December, 1991



<b>Accession For</b>	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC Tab	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Special
A-1	

## *Acknowledgements*

This thesis would not have been possible without the help of many others. First, I would like to thank my advisor, Dr. Terzuoli, for his guidance throughout the project. I also want to thank the committee members, Capt Joseph, Lt Col Meer, and Maj Rogers, for the hours they devoted in reviewing the work and suggesting improvements. I will forever be indebted to my sponsor, Capt Jeffrey Fath, for his assistance, advice, and expertise. I must also thank Rick Beam for his help with UNIX, SRIM, and modem problems. Finally, but certainly not least, I thank my wife, Sue, and children, Arden and Sheila, for their understanding, patience, and willingness to undergo the entire AFIT experience with me.

Dale A. Stanley

## ***Table of Contents***

	<b>Page</b>
Acknowledgements .....	ii
List of Figures .....	vi
List of Tables .....	ix
List of Acronyms .....	x
Abstract .....	xi
<b>I. Introduction .....</b>	<b>1.1</b>
Background .....	1.1
Problem Statement .....	1.3
Research Objective .....	1.3
Research Questions .....	1.3
Definitions .....	1.3
Scope .....	1.4
Approach/Methodology .....	1.5
Materials and Equipment .....	1.6
Overview .....	1.6
<b>II. Literature Search and Theory .....</b>	<b>2.1</b>
Introduction .....	2.1
Software Verification and Validation Methodology .....	2.2
Conceptual Model Validation .....	2.4
Computerized Model Verification .....	2.7
Sphere .....	2.7
Flat Rectangular Plate .....	2.8
Flat Circular Plate or Disk .....	2.10
Right Circular Cylinder .....	2.12
Dihedral .....	2.13
Trihedral .....	2.16
Top Hat .....	2.21
Operational Validity .....	2.22
Data Validity .....	2.22
Similar Electromagnetic Software Validation Efforts .....	2.22
<b>III. SRIM Overview .....</b>	<b>3.1</b>
Introduction .....	3.1
General Operating Principles .....	3.1
Target Modeling .....	3.2
Ray Tracing .....	3.2

	<b>Page</b>
Radar Cross Section Calculations .....	3.4
Derivation of Equation 3.1 .....	3.4
Input File Requirements .....	3.7
Geometry File .....	3.7
Scattering Model File .....	3.8
Surface Assignments File .....	3.9
Radar System Data File .....	3.9
Code Automation Technique .....	3.9
IV. Results .....	4.1
Introduction .....	4.1
SRIM Verification .....	4.1
Internal Validity Tests .....	4.1
Results .....	4.2
Fixed Value Tests .....	4.4
Sphere Results .....	4.5
Rectangular Flat Plate Results .....	4.5
Circular Disk Results .....	4.10
Solid Right Circular Cylinder .....	4.13
Right Dihedral Results .....	4.16
Square Trihedral Results .....	4.17
Top Hat Results .....	4.22
Comparison to Other Models .....	4.24
DBSM and SRIM RCS Prediction Comparisons .....	4.24
SarTool and SRIM RCS Prediction Comparisons .....	4.29
Validation .....	4.40
Data Acquisition .....	4.40
Comparison of Measured Data and SRIM RCS Predictions .....	4.42
Summary .....	4.50
V. Conclusions .....	5.1
Introduction .....	5.1
Verification Conclusions .....	5.2
Validation Conclusions .....	5.3
Recommendations for Future Work .....	5.4
Appendix A: GIFT and RADSIM Input File Requirements .....	A.1
A.1 Geometry File, *.cg .....	A.1
A.2 Scattering Model File, Surface.dat .....	A.4
A.3 Surface Assignments File, *.sur .....	A.4
A.4 Radar Data File, Radar.dat .....	A.6
Appendix B: Actual SRIM Input Files Used .....	B.1
B.1 Input File Examples .....	B.1
B.2 Geometry and Surface Assignment Files .....	B.3

	<b>Page</b>
Appendix C: GIFT and RADSIM Prompts .....	C.1
C.1 GIFT Prompts .....	C.1
C.2 RADSIM Prompts .....	C.2
Appendix D: Automation Technique Programs .....	D.1
D.1 Rcs.script .....	D.1
D.2 Azimuth.f .....	D.2
D.3 Azimvary.f .....	D.5
D.4 Rcsplot.f .....	D.7
Appendix E: Closed Form PO Equations .....	E.1
E.1 Sphere Source Code, Sphere.f .....	E.1
E.2 Rectangular Plate Source Code, Rplate.f .....	E.2
E.3 Circular Plate Source Code, Cplate.f .....	E.4
E.4 Solid Right Circular Cylinder Source Code, Rccpo.f .....	E.6
E.5 Right Dihedral Source Code, Dihedral.f .....	E.10
E.6 Square Trihedral Source Code, Trihed.f .....	E.13
E.7 Top Hat Source Code, Tophat.f .....	E.21
Appendix F: Data Acquisition Programs .....	F.1
F.1 Data Transfer .....	F.1
F.2. Data Shift Source Code, Shiftpk.f .....	F.3
Bibliography .....	BIB.1
Vita .....	VITA.1



## **List of Figures**

<b>Figure</b>	<b>Page</b>
2.1. Sphere with Origin at Center .....	2.8
2.2. Flat Plate in XZ Plane .....	2.9
2.3. Disk in XZ Plane .....	2.11
2.4. Cylinder Oriented Along Y Axis .....	2.13
2.5. Right Dihedral .....	2.14
2.6. Square Trihedral .....	2.17
2.7. Top Hat with Disk in XZ Plane .....	2.21
3.1. Boolean Operations .....	3.2
3.2. Sideview of Target and Emanation Plane .....	3.3
3.3. Orientation of Source and Observer .....	3.5
3.4. GIFT and RADSIM Input/Output Files .....	3.8
3.5. RCS.SCRIPT Flow Diagram .....	3.10
4.1. Plate in XZ Plane - SRIM RCS Prediction Repeatability, X=10 cm, Z=15 cm, F=10 GHz, Elev=0°, RS=.5 cm .....	4.2
4.2. Sphere - SRIM RCS Prediction as a Function of Ray Spacing, Radius=5 cm, F=10 GHz, Elev=0° .....	4.4
4.3. Sphere - SRIM & PO RCS Predictions, Radius=5 cm, F=10 GHz, Elev=0° .....	4.5
4.4. Plate in XZ Plane Showing Effect of Absorbing Edges, X=10 cm, Y=1 mm, Z=15 cm, F=10 GHz, Elev=0°, RS=1 mm .....	4.7
4.5. Plate in XZ Plane - SRIM & PO RCS Predictions, X=10 cm, Z=15 cm, F=10 GHz, Elev=0° .....	4.8
4.6. Plate in XZ Plane - SRIM & PO RCS Prediction Differences, X=10 cm, Z=15 cm, F=10 GHz, Elev=0° .....	4.8
4.7. Plate in XZ Plane - SRIM & PO RCS Predictions, X=10 cm, Z=15 cm, F=10 GHz, Elev=35° .....	4.9
4.8. Plate in XZ Plane - SRIM & PO RCS Prediction Differences, X=10 cm, Z=15 cm, F=10 GHz, Elev=35° .....	4.9
4.9. Disk in XZ Plane - SRIM & PO RCS Predictions, Radius=5 cm, F=10 GHz, Elev=0° .....	4.11
4.10. Disk in XZ Plane - SRIM & PO RCS Prediction Differences, Radius=5 cm, F=10 GHz, Elev=0° .....	4.11
4.11. Disk in XZ Plane - SRIM & PO RCS Predictions, Radius=5 cm, F=10 GHz, Elev=35° .....	4.12
4.12. Disk in XZ Plane - SRIM & PO RCS Prediction Differences, Radius=5 cm, F=10 GHz, Elev=35° .....	4.12
4.13. RCC Along Y Axis - SRIM & PO RCS Predictions, Radius=10.16 cm, Length=20.32 cm, F=10 GHz, Elev=0° .....	4.14
4.14. RCC Along Y Axis - SRIM & PO RCS Difference, Radius=10.16 cm, Length=20.32 cm, F=10 GHz, Elev=0° .....	4.14
4.15. RCC Along Y Axis - SRIM & PO RCS Predictions, Radius=10.16 cm, Length=20.32 cm, F=10 GHz, Elev=35° .....	4.15

<b>Figure</b>	<b>Page</b>
4.16. RCC Along Y Axis - SRIM & PO RCS Differences, Radius=10.16 cm, Length=20.32 cm, F=10 GHz, Elev=35° . . . . .	4.15
4.17. Right Dihedral - SRIM & PO RCS Predictions, X=Y=10 cm, Z=15 cm, F=10 GHz, P=HH, Elev=0° . . . . .	4.18
4.18. Right Dihedral - SRIM & PO RCS Predictions, X=Y=10 cm, Z=15 cm, F=10 GHz, P=VV, Elev=0° . . . . .	4.18
4.19. Right Dihedral - SRIM & PO RCS Predictions, X=Y=10 cm, Z=15 cm, F=10 GHz, P=HH, Elev=35° . . . . .	4.19
4.20. Right Dihedral - SRIM & PO RCS Predictions, X=Y=10 cm, Z=15 cm, F=10 GHz, P=VV, Elev=35° . . . . .	4.19
4.21. Square Trihedral - SRIM & PO RCS Predictions, X=Y=Z=10 cm, F=10 GHz, P=HH, Elev=0° . . . . .	4.20
4.22. Square Trihedral - SRIM & PO RCS Predictions, X=Y=Z=10 cm, F=10 GHz, P=VV, Elev=0° . . . . .	4.20
4.23. Square Trihedral - SRIM & PO RCS Predictions, X=Y=Z=10 cm, F=10 GHz, P=HH, Elev=35° . . . . .	4.21
4.24. Square Trihedral - SRIM & PO RCS Predictions, X=Y=Z=10 cm, F=10 GHz, P=VV, Elev=35° . . . . .	4.21
4.25. Top Hat - SRIM & PO RCS Predictions, Disk Radius=20.32 cm, Cyl Radius=10.16 cm, Cyl Length=20.32 cm, F=10 GHz, P=HH, Elev=0° . . . . .	4.23
4.26. Top Hat - SRIM & PO RCS Predictions, Disk Radius=20.32 cm, Cyl Radius=10.16 cm, Cyl Length=20.32 cm, F=10 GHz, P=VV, Elev=0° . . . . .	4.23
4.27. Plate in XZ Plane - SRIM & DBSM RCS Predictions, X=10 cm, Z=15 cm, F=10 GHz, Elev=0° . . . . .	4.25
4.28. Plate in XZ Plane - SRIM & DBSM RCS Predictions, X=10 cm, Z=15 cm, F=10 GHz, Elev=35° . . . . .	4.25
4.29. Disk in XZ Plane - SRIM & DBSM RCS Predictions, Radius=5 cm, F=10 GHz, Elev=0° . . . . .	4.26
4.30. Disk in XZ Plane - SRIM & DBSM RCS Predictions, Radius=5 cm, F=10 GHz, Elev=35° . . . . .	4.26
4.31. Cylinder Along Y Axis - SRIM & DBSM RCS Predictions, Radius=10.16 cm, Length=20.32 cm, F=10 GHz, Elev=0° . . . . .	4.27
4.32. Cylinder Along Y Axis - SRIM & DBSM RCS Predictions, Radius=10.16 cm, Length=20.32 cm, F=10 GHz, Elev=35° . . . . .	4.27
4.33. Right Dihedral - SRIM & DBSM RCS Predictions, X=Y=10 cm, Z=15 cm, F=10 GHz, P=HH, Elev=0° . . . . .	4.28
4.34. Plate in XZ Plane - SarTool RCS Prediction, X=10 cm, Z=15 cm, F=10 GHz, Elev=.1° (28:4-3) . . . . .	4.31
4.35. Plate in XZ Plane - SRIM RCS Prediction, X=10 cm, Z=15 cm, F=10 GHz, Elev=0° . . . . .	4.31
4.36. Plate in XZ Plane - SarTool RCS Prediction, X=10 cm, Z=15 cm, F=10 GHz, Elev=35° (28:4-4) . . . . .	4.32
4.37. Plate in XZ Plane - SRIM RCS Prediction, X=10 cm, Z=15 cm, F=10 GHz, Elev=35° . . . . .	4.32
4.38. Disk in XZ Plane - Sar Tool RCS Prediction, Radius=5 cm, F=10 GHz, Elev=.1° (27:4-7) . . . . .	4.33

<b>Figure</b>	<b>Page</b>
4.39. Disk in XZ Plane - SRIM RCS Prediction, Radius=5 cm, F=10 GHz, Elev=0°	4.33
4.40. Disk in XZ Plane - SarTool RCS Prediction, Radius=5 cm, F=10 GHz, Elev=35° (28:4-8)	4.34
4.41. Disk in XZ Plane - SRIM RCS Prediction, Radius=5 cm, F=10 GHz, Elev=35°	4.34
4.42. Right Dihedral - SarTool RCS Prediction, X=Y=10 cm, Z=15 cm, F=10 GHz, Elev=.1° (28:4-12)	4.35
4.43. Right Dihedral - SRIM RCS Prediction, X=Y=10 cm, Z=15 cm, F=10 GHz, Elev=0°	4.35
4.44. Right Dihedral - SarTool RCS Prediction, X=Y=10 cm, Z=15 cm, F=10 GHz, Elev=35° (28:4-15)	4.36
4.45. Right Dihedral - SRIM RCS Prediction, X=Y=10 cm, Z=15 cm, F=10 GHz, Elev=35°	4.36
4.46. Square Trihedral - SarTool RCS Prediction, X=Y=10 cm, F=10 GHz, Elev=.1° (28:4-18)	4.37
4.47. Square Trihedral - SRIM RCS Prediction, X=Y=Z=10 cm, F=10 GHz, Elev=0°	4.37
4.48. Square Trihedral - SarTool RCS Prediction, X=Y=Z=10 cm, F=10 GHz, Elev=35° (28:4-21)	4.38
4.49. Square Trihedral - SRIM RCS Prediction, X=Y=Z=10 cm, F=10 GHz, Elev=35°	4.38
4.50. Top Hat - SarTool RCS Prediction, Disk Radius=20.32 cm, Cyl Radius=10.16 cm, Cyl Length=20.32 cm, F=10 GHz, Elev=.1° (28:4-26)	4.39
4.51. Top Hat - SRIM RCS Prediction, Disk Radius=20.32 cm, Cyl Radius=10.16 cm, Cyl Length=20.32 cm, F=10 GHz, Elev=0°	4.39
4.52. Measured Data Target Orientation - A) Square Plate; B) Circular Disk; C) Solid Cylinder; D) Top Hat; E) Cone; F) Prolate Spheroid; G) Generic Missile (2 fins in YZ Plane and 2 fins in XY Plane)	4.41
4.53. Square Plate in XZ Plane - SRIM RCS Prediction & Measured Data, X=Z=15.24 cm, Y=.159 cm, F=12 GHz, P=HH, Elev=0°	4.44
4.54. Circular Disk in XZ Plane - SRIM RCS Prediction & Measured Data, Radius=15.24 cm, Y=.159 cm, F=12 GHz, P=HH, Elev=0°	4.44
4.55. Circular Cylinder Along Y Axis - SRIM RCS Prediction & Measured Data, Radius=3.8 cm, Length=31.27 cm, F=8 GHz, P=HH, Elev=0°	4.45
4.56. Top Hat Ohio State University RCS Measurements, Disk Radius=20.32 cm, Cyl Radius=10.16 cm, Cyl Length=20.32 cm, F=10 GHz, P=HH, Elev=0° (11:858)	4.47
4.57. Top Hat - SRIM RCS Prediction, Disk Radius=20.32 cm, Cyl Radius=10.16 cm, Cyl Length=20.32 cm, F=10 GHz, P=HH, Elev=0°	4.47
4.58. Cone - SRIM RCS Prediction & Measured Data, Base Radius=3.8 cm, Axial Length=8.5 cm, F=18 GHz, P=HH, Elev=0°	4.48
4.59. Prolate Spheroid - SRIM RCS Prediction & Measured Data, Major Axis Length=18 cm, Minor Axis Length=7.7 cm, F=10 GHz, P=HH, Elev=0°	4.48
4.60. Generic Missile - SRIM RCS Prediction & Measured Data, Length=53.3 cm, Radius=3.81 cm, F=8.5 GHz, P=HH, Elev=0°	4.49
A.1.1. Primitive Geometry	A.3

## *List of Tables*

<b>Table</b>	<b>Page</b>
4.1. Run Times as a Function of Ray Spacing .....	4.4
4.2. Measured Data Target Dimensions .....	4.41
4.3. Target $L/\lambda$ Ratios .....	4.42
4.4. Comparisons Used to Evaluate SRIM .....	4.51
A.1.1. Format for Geometry.cg File .....	A.2
A.1.2. Primitive Definition Description .....	A.3
A.3.1. Surface Identification Numbers .....	A.5
B.2.1. Correlation of Geometry Files and Chapter 4 Figures .....	B.9

## *List of Acronyms*

AFIT	Air Force Institute of Technology
BOX	Box Primitive
CAGES	Cross-Section and Glint Evaluation Simulation
CSG	Constructive Solid Geometry
dBsm	Decibel with reference to one square meter
DBSM	Denmar Backscatter Signatures Method
EHY	Elliptical Hyperboloid Primitive
ELEV	Elevation Angle
ELL	Ellipsoid of Revolution Primitive
EMCC	Electromagnetic Code Consortium
ERIM	Environmental Research Institute of Michigan
F	Frequency
GO	Geometrical Optics
GTD	Geometrical Theory of Diffraction
HH	Horizontal Polarization for Transmit and Receive
LOS	Line of Sight
MOM	Method of Moments
PO	Physical Optics
PSM	Polarization Scattering Matrix
PTD	Physical Theory of Diffraction
RCC	Right Circular Cylinder
RCS	Radar Cross Section
RS	Ray Spacing
SAR	Synthetic Aperture Radar
SB	Single Bounce
SPH	Sphere Primitive
SRIM	Simulated Radar Image Code
TB	Triple Bounce
UTD	Uniform Theory of Diffraction
V&V	Verification and Validation
VV	Vertical Polarization for Transmit and Receive

### ***Abstract***

→ The objectives of this study were to verify and validate the Simulated Radar Image (SRIM) Code Version 4.0 monostatic radar cross section (RCS) predictions. SRIM, developed by the Environmental Research Institute of Michigan, uses the theory of Physical Optics (PO) to predict backscatter for a user specified aspect angle. Target obscuration and multiple reflections are taken into account by sampling the target with ray tracing.

The software verification and validation technique followed in this study entailed comparing the code predictions to closed form PO equations, other RCS prediction software packages, and measured data. The targets analyzed were a sphere, rectangular flat plate, circular flat plate, solid right circular cylinder, dihedral and trihedral corner reflectors, top hat, cone, prolate spheroid, and generic missile. SRIM RCS predictions are shown for each of the targets as a function of frequency, aspect angle, and ray density. Also presented is an automation technique that enables the user to run SRIM sequentially over a range of azimuth angles. The FORTRAN code written by the author for the PO equations is also provided. ↙

The results show that the accuracy of the SRIM predictions improve as the ray density increases. Except for the top hat, there is close agreement between the SRIM predictions and PO equations when the targets are sampled with fine ray spacing. There is also good agreement between the SRIM predictions and measured data near the specular region.

# VERIFICATION AND VALIDATION OF THE SIMULATED RADAR IMAGE (SRIM) CODE RADAR CROSS SECTION PREDICTIONS

## *I. Introduction*

### *Background*

An electromagnetic wave that impinges upon a metallic object, hereinafter referred to as the target, sets up surface currents that reradiate electromagnetic waves. This reradiation is called the scattered field. The amount of power (normalized with respect to the incident field and spherical spreading factor) received from the scattered field by a receiving antenna is called the radar cross section (RCS) of the target (17:47-48). As such, RCS is a parameter that must be considered in the design and modification of every weapon system in which mission application makes electromagnetic concealment, or stealth, of primary importance. Mathematically, the RCS is represented as

$$\sigma = 4\pi \lim_{R \rightarrow \infty} R^2 \frac{|\vec{E}^s|^2}{|\vec{E}^i|^2} = \lim_{R \rightarrow \infty} R^2 \frac{|\vec{H}^s|^2}{|\vec{H}^i|^2} \quad (1.1)$$

where  $\sigma$  is the RCS in square meters,  $R$  is the distance between the target and radar,  $\vec{E}^s$  and  $\vec{H}^s$  are the scattered electric and magnetic fields, and  $\vec{E}^i$  and  $\vec{H}^i$  are the incident electric and magnetic fields, respectively.

At high frequencies, a target can be considered as a collection of individual scattering centers. The first step in reducing the RCS of an object is to identify each of these scattering centers. One way to accomplish this is through high resolution Synthetic Aperture Radar (SAR) imaging (21:Tape 1). However, it would be cost prohibitive to construct a prototype model of a proposed weapon system for each design iteration for the sole purpose of determining its RCS by imaging. A better approach is to simulate the SAR image with software. A further justification for image modeling arises from the need to depict the SAR image as a function of aspect angle to enhance training for image analysts. This prompted the Environmental Research Institute of Michigan (ERIM) to develop the Simulated Radar Image (SRIM) software package. SRIM provides target RCS predictions and the data required to generate simulated synthetic aperture radar images.

SRIM uses a set of 25 primitives and Constructive Solid Geometry (CSG) to model targets. In CSG, the boolean operations of union, difference, and intersection are used to combine primitives to construct the target desired. Primitives are solid geometry shapes such as boxes, cylinders, and spheres. The RCS predictions are based on the high frequency technique of Physical Optics.

As in every software program, SRIM cannot be used with confidence until it is thoroughly verified and validated (V&V). To date, a thoroughly documented SRIM V&V has not been completed for either the RCS predictions or image generation files.



### ***Problem Statement***

Within the realm of Physical Optics applicability, the SRIM radar cross section predictions and image generation files must be verified and validated for each of the primitives used in target model construction.

### ***Research Objective***

The objective of this thesis was to verify and validate the SRIM Version 4.0 RCS predictions for primitives that have a closed form Physical Optics solution.

### ***Research Questions***

In the process of verifying and validating the SRIM RCS predictions, the following questions were specifically addressed:

1. Do the SRIM RCS predictions agree with independently generated Physical Optics RCS predictions?
2. What effect does combining primitives have on the SRIM RCS predictions, and do the results agree with Physical Optics theory?
3. Do the SRIM predictions agree with measured RCS test data?
4. What additional steps must be taken to completely validate SRIM?

### ***Definitions***

Often, the terms verification and validation are used interchangeably. However, the terms can have far different meanings. In this thesis the terms are defined in the same manner as in a software verification and validation article by Sargent (26). He defines model verification as "ensuring that the computer program of the computerized model and its implementation is correct" (26:33). On the other hand, he quotes Schlesinger's definition of model validation as "substantiation that a computerized model within its domain of applicability

possesses a satisfactory range of accuracy consistent with the intended application of the model" (26:33). In other words, verification is ensuring the software has been encoded correctly, while validation is ensuring the code gives results to within an acceptable accuracy over its region of applicability.

### ***Scope***

Only the SRIM RCS predictions were evaluated, not the image generation files. Image validation requires comparison to real SAR imagery, thus necessitating the use of classified data and the construction of geometric models for real targets. Extensive target modeling and the use of classified data was considered beyond the scope of this project.

The main thrust of this effort focused on perfectly conducting primitives for which a closed form solution exists. A closed form Physical Optics solution does not exist for the full set of 25 primitives used by SRIM. As such, the primitives evaluated were the sphere, rectangular flat plate (thin box), flat disk (thin cylinder), and solid right circular cylinder. To address the second question under *Research Questions*, a dihedral and trihedral were modeled and evaluated using flat plates, and a top hat was modeled and evaluated using a disk and cylinder. In addition, SRIM predictions were compared with measured data for a rectangular plate, circular disk, solid cylinder, top hat, cone, prolate spheroid, and generic missile.

The high frequency electromagnetic scattering prediction technique used for this study was limited to Physical Optics theory. Other high frequency techniques are more accurate when away from the specular region of the target but they are more computer intensive.

Although SRIM has the capability of providing fully polarimetric RCS predictions, this study was limited to co-polarized scattering. That is, the transmit and receive signals were both either horizontally or vertically polarized.

### ***Approach/Methodology***

Following the techniques proposed by Sargent (26), the approach used to verify and validate SRIM consisted of internal validity tests, fixed value tests, comparison to other models, and historical data validation.

Internal validity tests verify the SRIM RCS predictions give consistent results from run to run. To verify repeatability, multiple SRIM runs were completed using identical targets and radar parameters. As a further test of repeatability, additional runs were completed in which the user-provided input parameters that should not affect RCS predictions were varied.

Fixed value tests verify that SRIM gives the same RCS predictions that would be obtained if the predictions were generated by "hand calculation." To establish a baseline that served as the hand calculated values, equations that predict RCS using the theory of Physical Optics were encoded in FORTRAN. SRIM predictions were compared to this baseline.

The final technique used for SRIM verification was comparison to other RCS prediction software programs. The programs selected for this comparison were the Denmark Backscatter Signatures Method (DBSM) code and SarTool. DBSM is a prediction scattering code developed by Denmark Inc., Carson City NV. It is based on the high frequency asymptotic techniques of Physical Optics and Physical Theory of Diffraction. In this study, edge currents were not used so the DBSM predictions were based solely on Physical Optics. SarTool was developed by The

Analytic Sciences Corporation and is also based on Physical Optics. SarTool and SRIM RCS predictions were compared because a thorough analysis of SarTool's RCS predictions for the primitives of interest was completed in the work done by Young (28).

The technique of historical data validation was used to validate SRIM. It is a comparison of SRIM predictions with measured RCS data. In this comparison, the code's region of applicability (limitations due to assumptions in Physical Optics theory) was taken into consideration.

### ***Materials and Equipment***

This work was sponsored by Wright Laboratories, Model Based Vision Lab, Wright-Patterson Air Force Base, Ohio. They provided the computer resources necessary to complete the project. This included a copy of SRIM, Version 4.0, and a Sun Sparc Station. Measured RCS data was acquired in the AFIT anechoic test chamber.

### ***Overview***

The Literature Review and Theory, Chapter II, discusses software verification and validation methodology, electromagnetic scattering theory, closed form Physical Optic equations, and electromagnetic software validation efforts similar to this thesis. A description of SRIM detailing the general theory and input file requirements are provided in Chapter III. An automation technique that enables the user to run SRIM sequentially over a range of azimuth angles is also presented in Chapter III. Chapter IV presents and discusses the data used for verification and validation. Results and conclusions, and suggestions for future

work are presented in Chapter V. The Appendix includes: 1) a description of the input files required to run SRIM; 2) a listing of the geometry files used in this study; 3) a description of the user provided responses to SRIM prompts; and 4) the FORTRAN source code written during this effort for the automation technique, closed-form Physical Optic equations, and AFIT data acquisition.

## ***II. Literature Search and Theory***

### ***Introduction***

The Simulated Radar Image (SRIM) software package, developed by the Environmental Research Institute of Michigan (ERIM), generates simulated synthetic aperture radar images, radar cross section predictions, and shaded images of a target (4:Sec I,1). As explained in the previous chapter, the purpose of this thesis is to verify and validate (V&V) the radar cross section predictions SRIM provides. V&V is required to assess the accuracy of the code because, as so aptly stated by Miller (22:1283), "Accuracy is put foremost since results of insufficient or unknown accuracy have uncertain value and may even be harmful." Thus, V&V of SRIM is essential, especially when considering it is being used in the analysis of polarimetric Synthetic Aperture Radar imagery (5:9).

The Electromagnetic Code Consortium (EMCC), a committee consisting of a government steering group and members from the industry/academic community, has undertaken efforts to address the issue of code validation. Two of their goals are to establish validation/measurement procedures which will be imposed on all future code developments and to benchmark industry/university-developed computational codes (14:20). Currently, a baseline for RCS software validation does not exist. Unfortunately, a common sense or 'seat-of-the-pants' approach is the accepted methodology for electromagnetic code software validation (23). The RCS community realizes this is a serious shortfall and is taking steps to alleviate the problem. In addition to the work being instigated by the EMCC, RCS

validation workshops are scheduled to be held in Chicago at the Antennas and Propagation Symposium in July 92.

Due to the lack of a baseline, this literature search initially concentrated on acceptable software methodologies that apply to software in general. It was apparent from that effort that code V&V can be broken down into a four step process as explained in the next section. Therefore, additional literature addressing each of the four steps was reviewed. This led into a review of electromagnetic scattering theory. Finally, a search was made of thesis efforts similar to this one.

### ***Software Verification and Validation Methodology***

There is considerable literature available on the subject of verification and validation of computer simulations. Sargent (26:33-38) provides an excellent overview governing the V&V of simulation models by relating V&V to the overall modeling process. According to Sargent, the steps comprising the modeling process are: (1) conceptual model validity; (2) computerized model verification; (3) operational validity; and (4) data validity (26:33).

Conceptual model validation entails verifying that any assumptions used in developing the model are correct. Computerized model verification involves ensuring the code was correctly encoded. Both of these steps correspond to the definition of verification given in Chapter 1. Sargent (26:33) defines operational validity as determining that the code gives sufficiently accurate results over its domain of applicability. This corresponds to the definition of validation given in Chapter 1. The domain of applicability is a point that needs to be stressed because, as pointed out in the next section, Physical Optics solutions give excellent

results (that is, they agree with measured data) at and near the specular region. But, as the aspect angle approaches grazing incidence, agreement with measured data seriously degrades. The final step in the modeling process is data validity which Sargent defines as "ensuring that the data necessary for model building, model evaluation and testing, and conducting the model experiments to solve the problem are adequate and correct" (26:33).

In addition to the four step modeling process, Sargent (26:33-34) lists several "validation" techniques that can be used for verification and validation. The techniques that apply to this study are: internal validity tests, fixed value tests, comparison to other models, and historical data validation. Internal validity tests ensure that the results are consistent from run to run. Fixed value tests involve comparing the output of the simulation against hand calculated values. Comparison to other models is self-explanatory. Historical data validation is comparing simulation results to measured test data. The internal validity tests usually apply to a stochastic model, or in other words, to a model in which one or more input parameters vary in a random fashion. SRIM does not use stochastic modeling to generate RCS predictions. However, one of the input parameters required to run SRIM, ray spacing, is somewhat arbitrarily-selected by the user. This introduces a variability that should be evaluated by what Sargent calls internal validity tests.

Since the objective of this thesis was to verify and validate SRIM, the thrust of the effort was focused on computerized model verification and operational validity. However, for completeness, the following sub-sections address each of the four steps. The electromagnetic theory that applies to each step is also presented.



**Conceptual Model Validation.** The Physical Optics theory used by SRIM is only one of several techniques that are available for predicting electromagnetic scattering. As is common in all the prediction scattering techniques, Physical Optics is based on certain approximations. It is beyond the scope of this thesis to verify if the approximations are accurate. But, the approximations must be taken into consideration because they constrain the region of code applicability. This sub-section reviews the most commonly used scattering prediction techniques and identifies their limitations and/or approximations.

The analytic techniques used to predict scattering are categorized as either low frequency, resonance, or high frequency methods. The category names are somewhat misleading because the techniques do not depend solely on frequency, but rather on the electrical size of the target. Low frequency methods are employed for targets where the linear dimension of the target,  $l$ , is less than the wavelength,  $\lambda$ , of the radar signal. Whereas, resonance region methods are used when  $l$  and  $\lambda$  are approximately the same; high frequency methods are used when  $l$  is greater than  $\lambda$ . An article sponsored by the Air Force (5:22-23) provides a listing of the various techniques and their general region of applicability. However, caution must be used in selecting a technique for a particular application because the applicability regions are not clearly defined (19:53-57).

High frequency methods can be employed for many applications because  $l/\lambda$  usually exceeds 10 for typical target sizes and radar frequencies. In this regime, a target can be treated as a collection of localized scattering centers. The total RCS is then determined by coherently adding the contribution from each scatterer. The difficulty lies in predicting the return from each scatterer. By employing approximations, asymptotic mathematical techniques can be used to predict this

return. Common techniques in use are: Geometrical Optics (GO), Geometric Theory of Diffraction (GTD), Uniform Theory of Diffraction (UTD), Physical Optics (PO), and the Physical Theory of Diffraction (PTD) (5:23). There are many texts and papers published on each of these methods. As an example, Knott (18:252-264) provides an extensive historical survey and theoretical development for each of these methods. Rather than go into a detailed explanation of each method, the reader is referred to the previous reference or the text authored by Knott (17:113-154). The following paragraphs briefly summarize the methods.

GO, GTD, and UTD are considered ray tracers because they assume energy propagates along slender tubes or rays. One short coming of ray tracers is the prediction of infinite fields at caustics. A caustic is defined as a point, line, or surface where an infinite number of rays intersect.

The simplest and oldest ray tracing method is GO. It is easily implemented if the principle radii of curvature at the surface are known at the point of reflection. Thus, it works well for doubly curved surfaces such as a sphere, but fails for singly curved or flat surfaces. Furthermore, it fails to predict scattering from tips and edges.

GTD, developed by Keller, extends the concepts of GO by including diffraction coefficients that take into account the reflections caused by edges and tips. However, the coefficients predict infinite fields at the reflection and shadow boundaries. These boundaries mark the discontinuity between the regions where the incident field does and does not exist.

The Uniform Theory of Diffraction (UTD), developed by Kouyoumjian and Pathak, corrects for the singularities in the GTD diffraction coefficients by

multiplying them by the appropriate Fresnel integrals (27:11). This correction factor is often called a transition function.

PO and PTD are non-ray tracing methods, hence there are no caustics. Using a tangent plane approximation, the theory of Physical Optics integrates the induced surface current over the illuminated portion of the target to determine the scattered field. It assumes zero surface current in the shadow region, thereby introducing a current discontinuity at the shadow/lit region boundary. PO does not accurately predict the currents near the discontinuity (i.e. edge currents). The accuracy degrades even further as the radar illumination angle is moved away from the target normal. Thus, PO works well only near the specular region of the target. SRIM uses ray tracing only to determine the target geometry, it uses PO to predict the scattered field. The Physical Theory of Diffraction (PTD), developed by Ufimtsev, extends the theory of PO by replacing the erroneous edge effects by the proper edge diffraction coefficients.

Many papers are devoted to comparing the superiority of one method over another. For example, Ando (2:1282-1285) uses a large circular disk to show that, when away from the mainbeam or specular region, PTD gives better predictions than PO. Marsland (20:1442) concluded that UTD and a creeping wave model for the circulating current on the edge of a disk give accurate RCS predictions even for grazing incidence. Choi (12:427-434) investigates the RCS of a cone-sphere by using a modified version of UTD that takes into consideration the creeping wave and compares the results with a Method of Moments (MOM) solution. In an analysis of flat plates, Ross (24:335) has shown that GTD offers significant improvement over PO when the viewing angle is away from the specular region. And as Lee (19:408-411) points out, there are actually 12 different versions of PO

that can be used to determine the bistatic cross section of a target. The versions give different answers because the fields calculated by PO do not generally satisfy reciprocity.

From the preceding paragraph, it is quite obvious that each method does have inherent advantages and disadvantages, but if used within the constraints of their limitations, give reasonably accurate results. Even though there are techniques that give better results than PO, the simplicity of incorporating PO in computer simulations makes it an attractive choice if the area of interest is at or near the specular region. Anderson (1:1158) demonstrates this by showing remarkable agreement between PO predictions and experimental data for a dihedral corner reflector with an internal angle of 85°.

**Computerized Model Verification.** To assess the accuracy to which SRIM implements PO theory, it is necessary to either independently encode the same equations used in SRIM or to encode PO equations available in the literature for the primitives being evaluated. Pursuing the latter option, this section lists the PO equations that were used during this effort.

**Sphere.** Knott (17:129) gives the PO solution for the sphere depicted in Fig 2.1 as

$$\sqrt{\sigma} = \sqrt{\pi a} \left[ \left( 1 + \frac{1}{j2ka} \right) \exp(-j2ka) - \frac{1}{j2ka} \right] \quad (2.1)$$

where  $\sigma$  is the complex RCS in units of square meters,  $a$  is the radius of the sphere, and  $k$  is the wavenumber ( $2\pi/\lambda$ ). In Eq (2.1), the phase has been referenced to the center of the sphere.

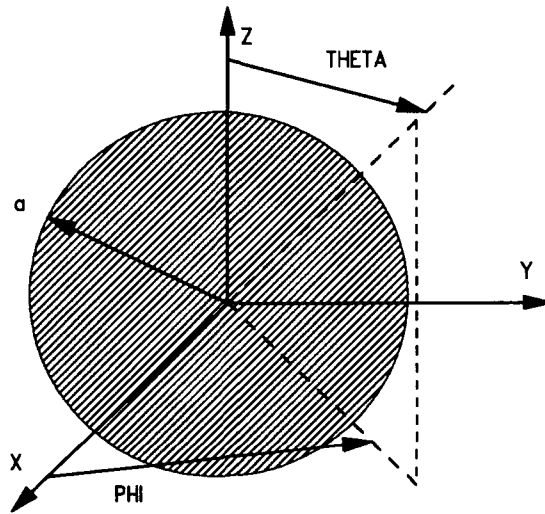


Figure 2.1. Sphere with Origin at Center

**Flat Rectangular Plate.** The complex RCS of a square flat plate oriented in the XZ plane as shown in Figure 2.2 is given in Blejer (10:5). Young (28:3-8) generalized the equation so it applies to a rectangular flat plate. The equation as given by Young, using the standard spherical coordinate system, is

$$\sqrt{\sigma_{XZ}} = -\frac{jkab}{\sqrt{\pi}} \sin\theta \sin\phi \frac{\sin(kb\cos\theta)}{kb\cos\theta} \cdot \frac{\sin(ka\sin\theta\cos\phi)}{ka\sin\theta\cos\phi} \cdot \exp(jk\sqrt{a^2+b^2}(\sin\theta\cos\phi+\cos\theta)) \quad (2.2)$$

where the subscript XZ indicates the plate lies in the XZ plane,  $k$  is the wavenumber,  $a$  is the dimension along the X axis, and  $b$  is the dimension along the Z axis.

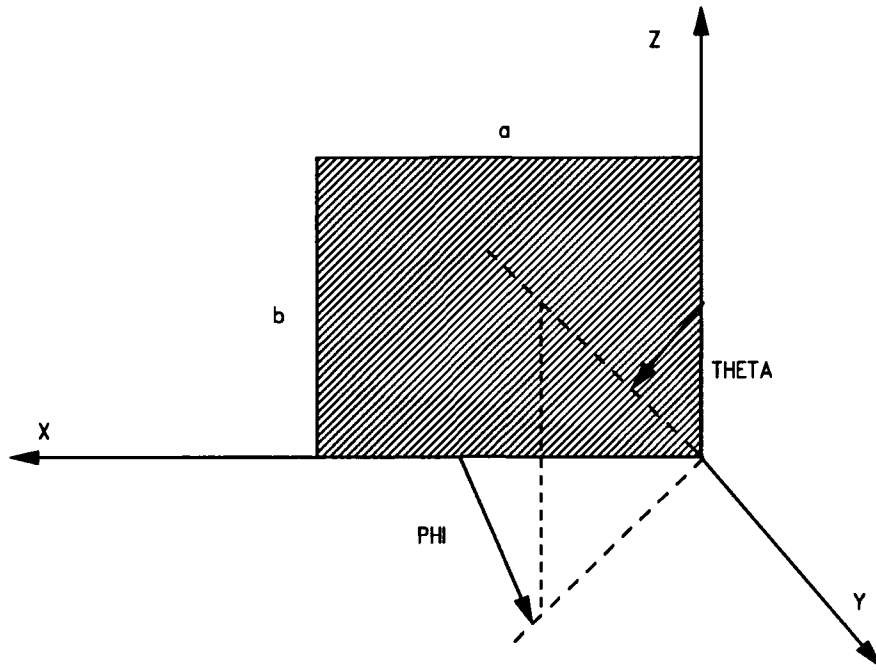


Figure 2.2. Flat Plate in XZ Plane

Likewise, the equations given by Blejer (10:5) and generalized by Young (28:3-8,3-4,5) for a rectangular flat plate situated in the YZ and XY planes, respectively, are given by

$$\sqrt{\sigma_{YZ}} = -\frac{jkab}{\sqrt{\pi}} \sin\theta \cos\phi \frac{\sin(kb \cos\theta)}{kb \cos\theta} \cdot \frac{\sin(ka \sin\theta \sin\phi)}{ka \sin\theta \sin\phi} \cdot \exp(jk\sqrt{a^2+b^2}(\sin\theta \sin\phi + \cos\theta)) \quad (2.3)$$

where  $a$  is the dimension along the Y axis and  $b$  is the dimension along the Z axis, and

$$\sqrt{\sigma_{XY}} = -\frac{jkab}{\sqrt{\pi}} \cos\theta \frac{\sin(ka \sin\theta \cos\phi)}{ka \sin\theta \cos\phi} \cdot \frac{\sin(kb \sin\theta \sin\phi)}{kb \sin\theta \sin\phi} \cdot \exp(jk\sqrt{a^2+b^2} \sin\theta (\sin\phi + \cos\phi)) \quad (2.4)$$

where  $a$  is the dimension along the X axis and  $b$  is the dimension along the Y axis.

**Flat Circular Plate or Disk.** The PO equation for a disk oriented in the XY plane is given in Ruck (25:510,513) as

$$\sigma = \frac{4\pi(\pi a^2)^2 \cos^2\theta}{\lambda^2} \left[ \frac{J_1(2ka \sin\theta)}{ka \sin\theta} \right]^2 \quad (2.5)$$

where  $\sigma$  is the RCS in square meters,  $a$  is the disk radius,  $J_1$  is the Bessel function of the first kind of order 1,  $k$  is the wavenumber, and  $\theta$  is the angle between the surface normal and the radar line of sight. For this orientation, the outward surface normal corresponds to the positive Z axis.

If the plate is oriented in the XZ plane, as shown in Figure 2.3 (standard spherical coordinate system), the outward surface normal is along the positive Y axis. In this case, Eq (2.5) can be used to find the RCS by replacing theta with the angle between the Y axis and the radar line of sight, psi, where

$$\cos\psi = \sin\theta \sin\phi \quad (2.6)$$

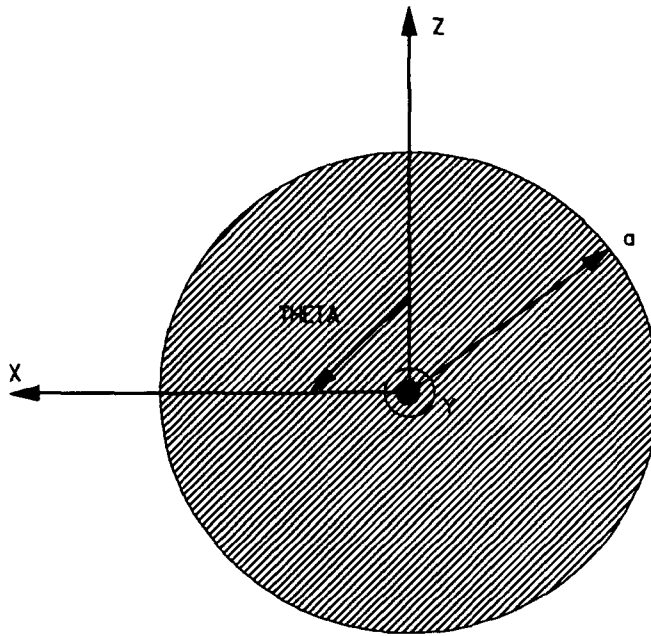


Figure 2.3. Disk in XZ Plane

which leads to

$$\sigma = 4\pi(\pi a^2)^2 \frac{\sin^2\theta \sin^2\phi}{\lambda^2} \cdot \left[ \frac{J_1(2ka \sin(\cos^{-1}(\sin\theta \sin\phi)))}{ka \sin(\cos^{-1}(\sin\theta \sin\phi))} \right]^2 \quad (2.7)$$

The angular relationship of Eq (2.6) is easily derived from trigonometry.

Similarly, if the plate is oriented in the YZ plane the outward surface normal is along the positive X axis. Then using the trigonometric relationship

$$\cos\psi = \sin\theta \cos\phi \quad (2.8)$$

where psi is now the angle between the X axis and the radar line of sight, Eq (2.5)

becomes



$$\sigma = 4\pi (\pi a^2)^2 \frac{\sin^2 \theta \cos^2 \phi}{\lambda^2} \cdot \left[ \frac{J_1(2ka \sin(\cos^{-1}(\sin \theta \cos \phi)))}{ka \sin(\cos^{-1}(\sin \theta \cos \phi))} \right]^2 \quad (2.9)$$

**Right Circular Cylinder.** The RCS for a right circular cylinder (RCC) can be derived by coherently summing the contribution from the circular end caps (disks) and cylindrical body. The difference in phase between the end cap contributions must be considered. Crispin (13:105-107) gives such an expression that was derived from a rigorous application of optics. However, the formula includes both Bessel and Struve functions.

A simpler method, albeit an approximation, is to 'blend' the RCS solutions of a disk and cylinder. That is, the RCS of a solid cylinder can be approximated by only considering the major scattering source. In other words, when the disk gives a larger return than the cylinder, use the disk RCS and vice versa. As an example, for the cylinder of Figure 2.4, the major return for phi angles between 0 and 45 is due to the cylindrical portion of the target. On the other hand, at phi angles between 45 and 90 most of the return is due to the end cap. Thus, the RCS can be approximated by using the cylinder return for  $0^\circ \leq \phi \leq 45^\circ$  and the disk return for  $46^\circ \leq \phi \leq 90^\circ$ . In this thesis, the 'blended' method was used to predict the total RCS of a solid RCC.

Using PO and the method of stationary phase, Knott (17:126) gives the equation for the RCS of the cylindrical body oriented as shown in Figure 2.4 as

$$\sigma = kaL^2 \left| \cos \phi \frac{\sin(kL \sin \phi)}{kL \sin \phi} \right|^2 \quad (2.10)$$

where  $\sigma$  is the RCS in square meters,  $a$  and  $L$  are the radius and length of the cylinder, respectively,  $k$  is the wavenumber, and  $\phi$  is the angle from broadside incidence.

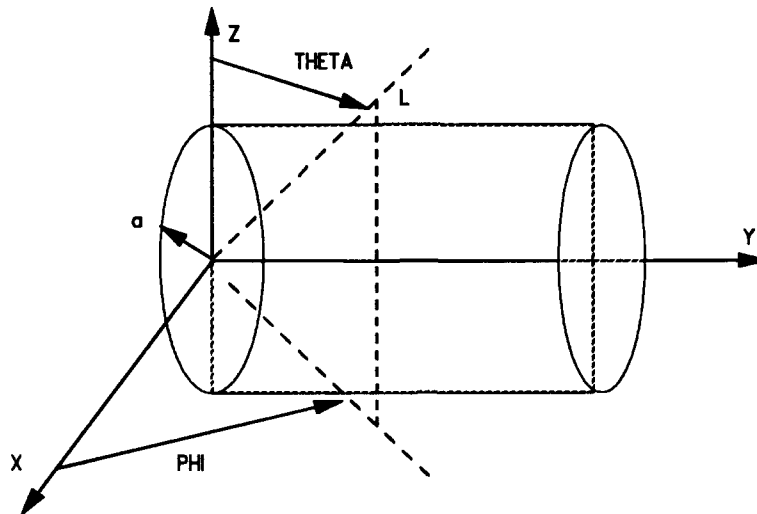


Figure 2.4. Cylinder Oriented Along Y Axis

**Dihedral.** A dihedral is not a primitive in the true sense of the word. Rather, it is a target comprised of two flat plates as shown in Figure 2.5. The plates join each other along a single common edge, and if perpendicular, make a right dihedral corner. Dihedrals are of interest in RCS studies because they occur whenever two flat surfaces meet at right angles and give large radar returns.

The specular returns from a dihedral consist of Single Bounce (SB) and Double Bounce (DB) returns. The SB return is due to the direct reflection off either face. The DB return is the return that undergoes a reflection from one surface to the other and then back to the radar or observer. The SB contribution can be calculated using Eqs (2.2), (2.3), and (2.4). Knott (16:408) solves for the DB

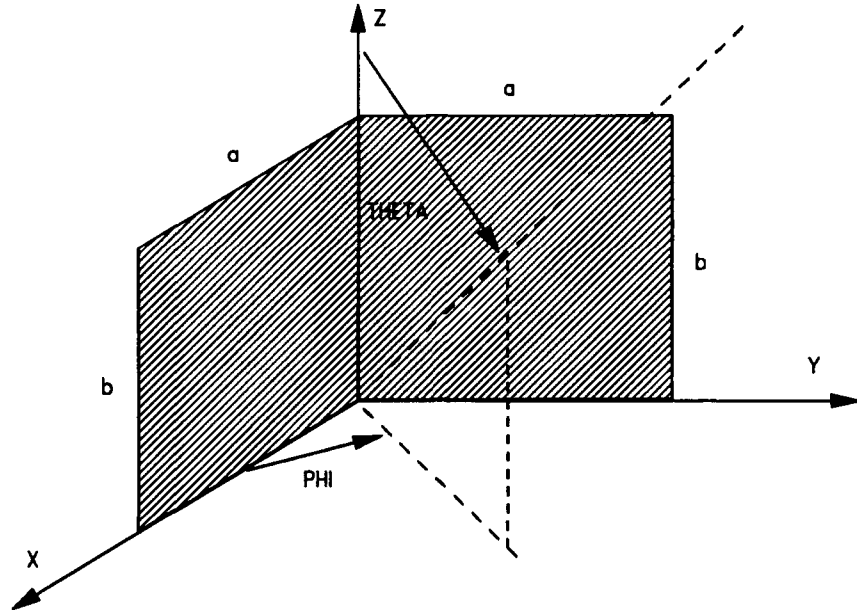


Figure 2.5. Right Dihedral

contribution by using a combined GO and PO approach. He uses GO (ray tracing) to determine the extent of surface illumination on the second face due to the reflections off the first face. Then using the effective width of illumination on the second plate, he uses PO to approximate the DB return for vertical and horizontal polarizations. In his development the transmitter and receiver were assumed to have the same polarization.

The DB return is polarization sensitive because the electric field of a vertically polarized wave reverses direction ( $180^\circ$  phase shift) at each reflection. The convention used in this thesis for vertical and horizontal polarization follows Balanis (6:189), in which vertical polarization implies the electric field is parallel to the plane of incidence. Thus, the terms parallel and vertical polarization are used interchangeably, as are the terms perpendicular and horizontal polarization. The plane of incidence is further defined as "the plane formed by a unit vector normal

to the reflecting interface and the vector in the direction of incidence" (6:185).

As reported by Blejer (8:1-2), P. Corona uses Knott's GO-PO formulation to give the backscattered field from the dihedral due to an incident plane wave for parallel and perpendicular polarizations. Blejer (8:3) extends this work and gives the complex RCS for horizontal polarization and vertical polarization for the dihedral geometry shown in Figure 2.5 as

$$\begin{aligned}
 \sqrt{\sigma_{HH}} = ab \frac{jk}{\sqrt{\pi}} & \left\{ \sin\theta \frac{\sin(kb\cos\theta)}{kb\cos\theta} \cdot \left[ \frac{\sin(kac\cos\phi\sin\theta)}{kac\cos\phi\sin\theta} \sin\phi \exp(jkac\cos\phi\sin\theta) \right. \right. \\
 & \left. \left. + \cos\phi \frac{\sin(kasin\phi\sin\theta)}{kasin\phi\sin\theta} \exp(jkasin\phi\sin\theta) + 2\sin\phi \right] \right. \\
 & \left. \left. + \sin\theta\sin\phi \frac{\exp(-jkb\cos\theta)}{jkb\cos\theta} \left[ 1 - \exp(-jkd\cos\theta) \frac{\sin(kd\cos\theta)}{kd\cos\theta} \right] \right\} \quad (2.11)
 \end{aligned}$$

and

$$\begin{aligned}
 \sqrt{\sigma_{VV}} = ab \frac{jk}{\sqrt{\pi}} & \left\{ \sin\theta \frac{\sin(kb\cos\theta)}{kb\cos\theta} \cdot \left[ \frac{\sin(kac\cos\phi\sin\theta)}{kac\cos\phi\sin\theta} \sin\phi \exp(jkac\cos\phi\sin\theta) \right. \right. \\
 & \left. \left. + \cos\phi \frac{\sin(kasin\phi\sin\theta)}{kasin\phi\sin\theta} \exp(jkasin\phi\sin\theta) - 2\sin\phi \right] \right. \\
 & \left. \left. - \sin\theta\sin\phi \frac{\exp(-jkb\cos\theta)}{jkb\cos\theta} \left[ 1 - \exp(-jkd\cos\theta) \frac{\sin(kd\cos\theta)}{kd\cos\theta} \right] \right\} \quad (2.12)
 \end{aligned}$$

where the double subscripts HH and VV imply horizontal and vertical polarization, respectively, for both transmit and receive antennas. The above equations are valid in the regions  $0^\circ \leq \theta \leq 90^\circ$  and  $0^\circ \leq \phi \leq 45^\circ$ . Blejer defines  $d$  as

$$d = \frac{a \cos \theta}{\cos \phi \sin \theta} \quad (2.13)$$

Alternatively, Blejer (8:4) gives the complex RCS double bounce response for the horizontal and vertical polarizations as

$$\begin{aligned} \sqrt{\sigma_{HH}^{DB}} = ab \frac{jk}{\sqrt{\pi}} \sin \theta \sin \phi \left\{ \frac{2 \sin(kb \cos \theta)}{kb \cos \theta} + \frac{\exp(-jkb \cos \theta)}{jkb \cos \theta} \right. \\ \left. \cdot \left[ 1 - \exp(-jkd \cos \theta) \frac{\sin(kd \cos \theta)}{kd \cos \theta} \right] \right\} \end{aligned} \quad (2.14)$$

and

$$\begin{aligned} \sqrt{\sigma_{VV}^{DB}} = -ab \frac{jk}{\sqrt{\pi}} \sin \theta \sin \phi \left\{ \frac{2 \sin(kb \cos \theta)}{kb \cos \theta} + \frac{\exp(-jkb \cos \theta)}{jkb \cos \theta} \right. \\ \left. \cdot \left[ 1 - \exp(-jkd \cos \theta) \frac{\sin(kd \cos \theta)}{kd \cos \theta} \right] \right\} \end{aligned} \quad (2.15)$$

**Trihedral.** Another common target used for RCS studies is the trihedral corner reflector. As in the dihedral, it frequently occurs in complex

targets and gives large radar returns. A trihedral is formed where three flat surfaces meet at a point, as shown in Figure 2.6.

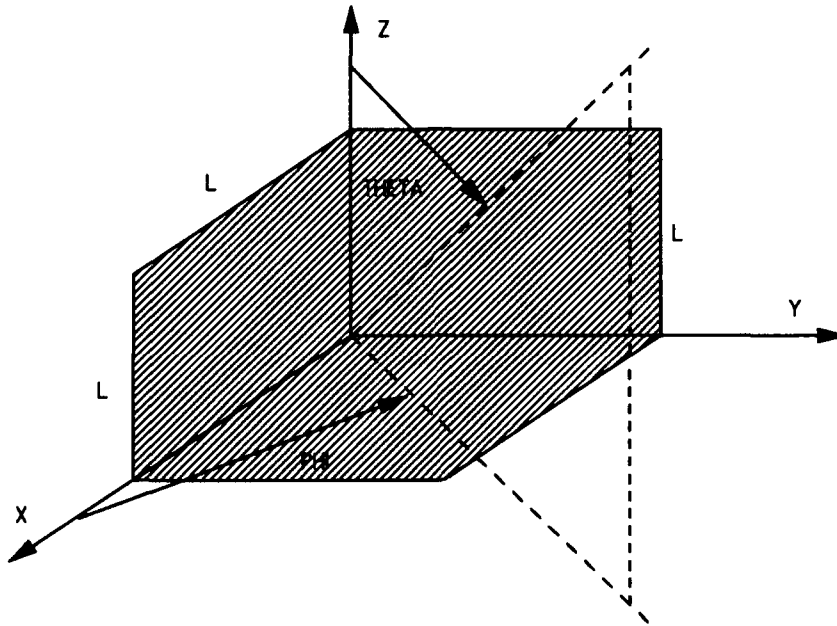


Figure 2.6. Square Trihedral

In addition to the SB and DB responses inherent in the dihedral, a trihedral also has a Triple Bounce (TB) response. To find the total RCS, all contributions must be added coherently. The coherent sum of the responses can be obtained by summing the Polarization Scattering Matrix (PSM) of each contribution. The PSM is a widely accepted method of representing an electromagnetic wave that takes into account both horizontal and vertical polarizations (28:2-3). It is given as

$$PSM = \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix} \quad (2.16)$$

where the subscripts indicate horizontal or vertical polarizations for the transmitter and receiver, respectively.

The following are the PSMs for the SB, DB<sub>x</sub>, DB<sub>y</sub>, DB<sub>z</sub>, and TB responses as given by Blejer (10). There are three DB responses because there are three dihedrals inherent in the trihedral. The subscript corresponds to the axis the dihedral crease is aligned along.

The Single Bounce PSM is (10:5)

$$SB(\theta, \phi) = \begin{pmatrix} \sqrt{\sigma_{xy}} + \sqrt{\sigma_{xz}} + \sqrt{\sigma_{yz}} & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \quad (2.17)$$

where the complex RCS for each of the plates is given in Eqs (2.2), (2.3), and (2.4).

The Double Bounce PSMs are (10:5-6)

$$DB_x = \frac{\sqrt{\sigma_{HB}^{DB}(\theta'', \phi'')}}{1 - \sin^2 \theta \cos^2 \phi} \cdot \begin{bmatrix} \cos^2 \theta \cos^2 \phi - \sin^2 \phi & 2 \cos \theta \sin \phi \cos \phi \\ 2 \cos \theta \sin \phi \cos \phi & \sin^2 \phi - \cos^2 \theta \cos^2 \phi \end{bmatrix} \exp(jkL \sin \theta \cos \phi) \quad (2.18)$$

where (9:9)

$$\begin{aligned} \theta'' &= \sin^{-1} \left( \sqrt{1 - \sin^2 \theta \cos^2 \phi} \right) \\ \phi'' &= \sin^{-1} \left( \frac{\cos \theta}{\sqrt{1 - \sin^2 \theta \cos^2 \phi}} \right) \end{aligned} \quad (2.19)$$

and

$$DB_Y = \frac{\sqrt{\sigma_{HH}^{DB}(\theta', \phi')}}{1 - \sin^2 \theta \sin^2 \phi} \cdot \begin{bmatrix} \cos^2 \phi - \cos^2 \theta \sin^2 \phi & 2 \cos \theta \sin \phi \cos \phi \\ 2 \cos \theta \sin \phi \cos \phi & \cos^2 \theta \sin^2 \phi - \cos^2 \phi \end{bmatrix} \exp(jkL \sin \theta \sin \phi) \quad (2.20)$$

where (9:8)

$$\begin{aligned} \theta' &= \sin^{-1}(\sqrt{1 - \sin^2 \theta \sin^2 \phi}) \\ \phi' &= \sin^{-1}\left(\frac{\cos \theta}{\sqrt{1 - \sin^2 \theta \sin^2 \phi}}\right) \end{aligned} \quad (2.21)$$

and

$$DB_Z = \sqrt{\sigma_{HH}^{DB}(\theta, \phi)} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \exp(jkL \cos \theta) \quad (2.22)$$

where the radicand of Eqs (2.18), (2.20), and (2.22) are given by Eq (2.14).

The Triple Bounce PSM is (10:7)

$$TB = \frac{j\mathbf{kA}}{\sqrt{\pi}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.23)$$



where (10:3)

$$\begin{aligned} A &= 4lmL^2/n & (m \leq n/2) \\ A &= l\left(4 - \frac{n}{m}\right)L^2 & (m \geq n/2) \end{aligned} \quad (2.24)$$

$$\begin{aligned} l &= \min[\cos\psi_x, \cos\psi_y, \cos\psi_z] \\ m &= \text{mid}[\cos\psi_x, \cos\psi_y, \cos\psi_z] \\ n &= \max[\cos\psi_x, \cos\psi_y, \cos\psi_z] \end{aligned} \quad (2.25)$$

and (10:4)

$$\begin{aligned} \cos\psi_x &= \sin\theta \cos\phi \\ \cos\psi_y &= \sin\theta \sin\phi \\ \cos\psi_z &= \cos\theta \end{aligned} \quad (2.26)$$

Finally, the complex RCS of the trihedral shown in Figure 2.6 is given by

$$\sqrt{\sigma_{\text{trihedral}}} = SB + DB_X + DB_Y + DB_Z + TB \quad (2.27)$$

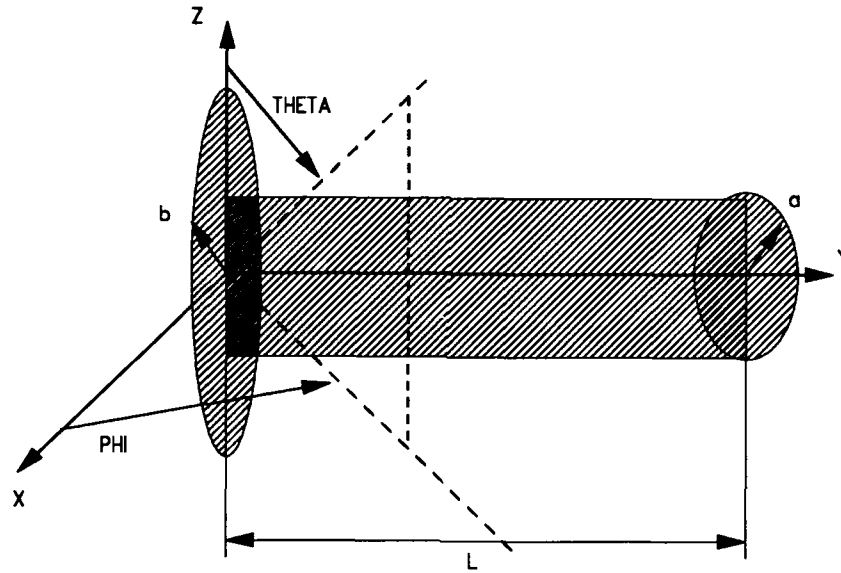


Figure 2.7. Top Hat with Disk in XZ Plane

**Top Hat.** A top hat is shown in Figure 2.7. It consists of a right circular cylinder with radius  $a$  and disk with radius  $b$  ( $a < b$ ). The disk is in the XZ plane and the axis of the cylinder is perpendicular to the plane of the disk.

Blejer (11:858) uses PO theory to derive the PSM of the top hat. He expresses it as

$$\begin{aligned}
 S_{VV}^{HH} = & \sqrt{k a \sin \theta} \exp(2k a \sin \theta + j\pi/4) \\
 & \cdot \left[ \pm \min(2L, 2(b-a)\cot\theta) - \frac{L \sin(kL \cos \theta) \exp(jkL \cos \theta)}{kL \cos \theta} \right] \\
 & -jk\sqrt{\pi} \cos \theta \left[ \frac{2b^2 J_1(2kb \sin \theta)}{2kb \sin \theta} - \frac{2a^2 J_1(2ka \sin \theta)}{2ka \sin \theta} \right] \\
 & -jk\sqrt{\pi} \cos \theta \left[ \frac{2a^2 J_1(2ka \sin \theta)}{2ka \sin \theta} \right] \exp(jkL \cos \theta)
 \end{aligned} \tag{2.28}$$

**Operational Validity.** Operational validity requires the code give 'sufficiently accurate' results only in the region of applicability. Because the code operates within the constraints of PO theory, the major area of comparison with real data should be confined to the proximity of the specular region. Furthermore, the effect of traveling and creeping waves is not modeled with PO. Therefore, differences between measured data and SRIM are expected. As a final note, 'sufficiently accurate' is not well defined. However, most RCS analysts consider agreement to within  $\pm 1$  decibel per square meter reasonable.

**Data Validity.** Numerous articles compare measured RCS test data with analytical derived predictions for flat plates, dihedrals, and cones. Often, only the test results are given with no discussion of data collection error. This is especially prevalent in the more recent articles, whereas, in contrast, an article submitted in 1965 by Bechtel (7:879) caveats all test results with an error band. It is beyond the scope of this thesis to determine the accuracy band of the test data used for comparison.

### ***Similar Electromagnetic Software Validation Efforts***

The Analytic Sciences Corporation developed the Synthetic Aperture Radar prediction Tool, appropriately entitled SarTool, to generate complex (magnitude and phase) radar signature data. Young (28) compared the results of SarTool for flat plates, dihedrals, and top hats with PO equations. His analysis clearly shows good agreement between SarTool predictions and PO solutions for the magnitude and phase of the circular flat plate. However, there are phase disparities between SarTool and the equations for the other primitives tested. Young concluded that the validation is not complete.

Further work in this area includes an Air Force Institute of Technology thesis that is validating the Cross-Section and Glint Evaluation Simulation (CAGES) software package developed by General Dynamics. CAGES generates radar cross section predictions using PO and PTD equations. A comparison between CAGES and SRIM could not be completed during this effort because both codes are being validated concurrently. Any follow on efforts should run a comparison between the two because both codes are modeled on the basis of primitives.

### ***III. SRIM Overview***

#### ***Introduction***

SRIM generates digital simulated synthetic aperture radar images and RCS predictions. Most of the software is coded in FORTRAN, although C code is used for reading and writing some of the input/output files. SRIM consists of six programs, five of which can be run in any sequence. To generate the RCS of a target, only two of the programs, GIFT and RADSIM, must be run. Keeping within the scope of this thesis, this chapter gives the general operating principles of SRIM as related to its RCS prediction capability. Also provided are descriptions of the GIFT and RADSIM input file requirements and code automation technique used during this study. The information that follows was gleaned from references (3), (4), and (15).

#### ***General Operating Principles***

SRIM is based on the theory of Physical Optics. As such, it does not use ray tracing to calculate the scattered field. However, it does use ray tracing to determine the target geometry, normal and curvature information at each reflection point, and the cartesian coordinates of these points. This method of target sampling takes into account target shadowing and multiple reflections. The immediate area surrounding the intersection of the ray and target is called a surface scattering patch. The integrated reflectivity density is calculated for each surface patch. The reflectivity density is used to calculate the RCS of the target by coherently summing the contribution from each surface patch. The following

explains how a target is modeled, the ray tracing technique used, and how the RCS is calculated.

**Target Modeling.** SRIM uses Constructive Solid Geometry (CSG) to model each target. In CSG, targets are modeled by combining solid primitives such as spheres, boxes, cylinders, and cones with the boolean operations of union, intersection, and difference. Figure 3.1 shows a two dimensional example of these operations.

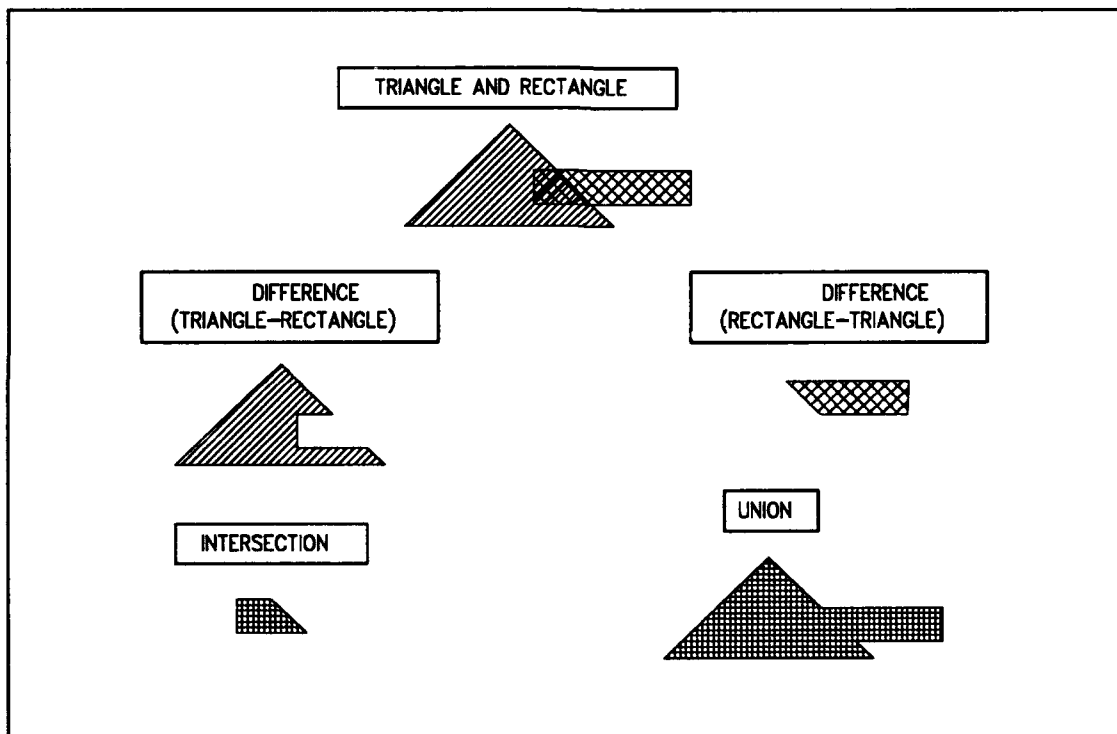


Figure 3.1. Boolean Operations

**Ray Tracing.** SRIM samples the target geometry by firing rays from an emanation plane that is located in space perpendicular to the radar line of sight (LOS). The emanation plane contains a view plane rectangle that is large enough to completely cover the target. SRIM calculates the size of the view plane rectangle by placing a bounding parallelepiped that just encompasses the entire

target. Thus, the size of the rectangle depends upon the radar LOS. The LOS is specified by the azimuth and elevation angles. Azimuth is measured in the XY plane from X towards Y and elevation is measured from the XY plane towards the Z axis. Figure 3.2 shows the geometry.

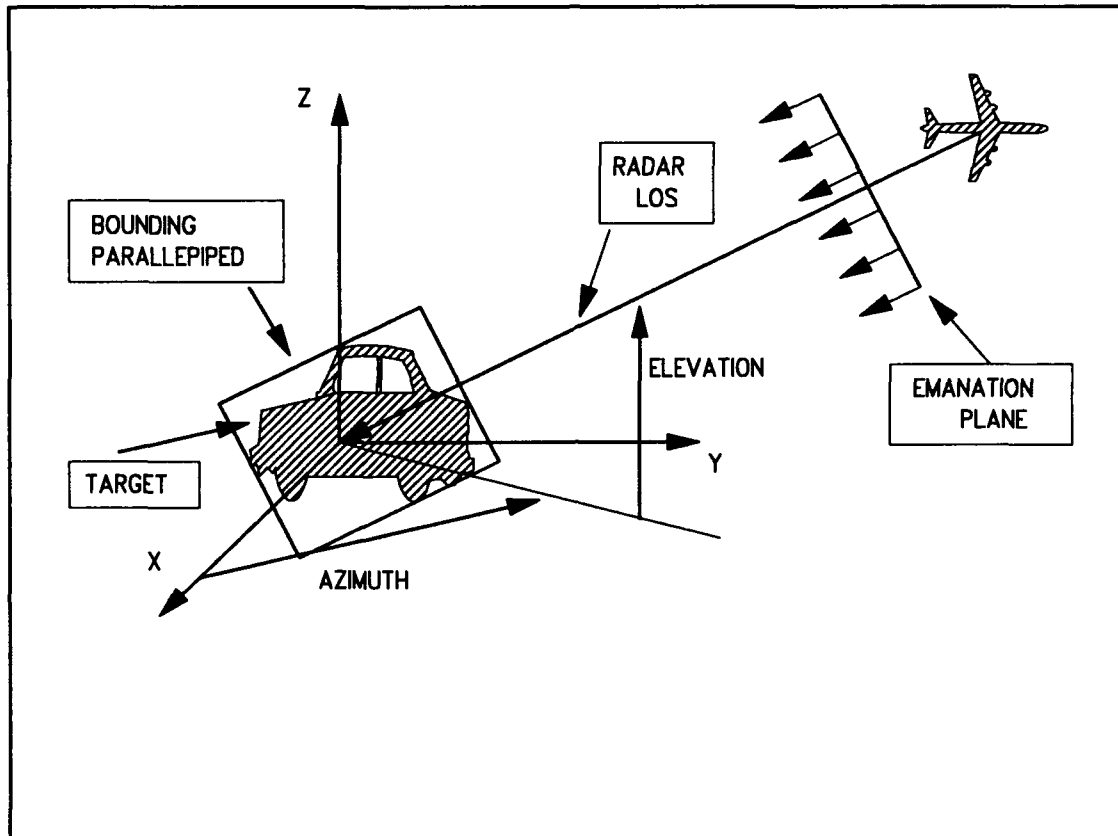


Figure 3.2. Sideview of Target and Emanation Plane

The user either specifies the number of rays or the distance between rays that are fired from the view plane rectangle. Therefore, the ray density the target is sampled with depends upon the size of the view plane rectangle, or conversely, the radar LOS.

To sample the target, SRIM sequentially casts each ray from the view plane rectangle. When a ray strikes the target, the surface normal and curvature are calculated. Each ray is traced through its specular reflections. This continues

until the ray leaves the target scene (i.e. no more reflections) or until a user specified maximum number of reflections is reached. At each subsequent reflection, a ray is also fired towards the emanation plane to determine if the reflection is within the radar LOS (thus accounting for shadowing or obscuration). This process is repeated for every ray fired from the view plane rectangle. The history of each ray emanating from the emanation plane is stored in the ray history file for later use.

**Radar Cross Section Calculations.** SRIM was initially developed to simulate synthetic aperture radar images. One of the parameters it calculates in the process is the reflection density,  $\gamma$ . The reflection density can also be used to calculate the target RCS by

$$\sigma = 4\pi r^2 \left| \frac{\hat{h}_R \cdot \vec{H}^s}{\vec{H}^i} \right|^2 = \frac{4\pi}{\lambda^2} \left| \int_S \gamma(\vec{r}') \exp(ik\hat{r}' \cdot \vec{r}') ds' \right|^2 \quad (3.1)$$

where  $\hat{h}_R$  is the magnetic polarization unit vector of the receiving antenna.

RADSIM performs the integration of Eq (3.1) over each surface patch. The contribution from each surface patch is then added coherently to give the total RCS. If the surface is flat, the integration is performed exactly. If the surface is curved, the method of stationary phase is used by approximating a singly curved surface with a parabolic cylinder and a doubly curved surface with an elliptic paraboloid.

**Derivation of Equation 3.1.** This equation was derived from the definition of RCS, basic electromagnetic theory, and Physical Optics. Consider the geometry shown in Figure 3.3, where the scattering source (target) surface is



depicted as  $S$ ,  $\vec{r}$  is the position vector from source origin to observer, and  $\vec{r}'$  is a position vector from source origin to a location on the source, and  $\hat{n}$  is the source outward surface normal.

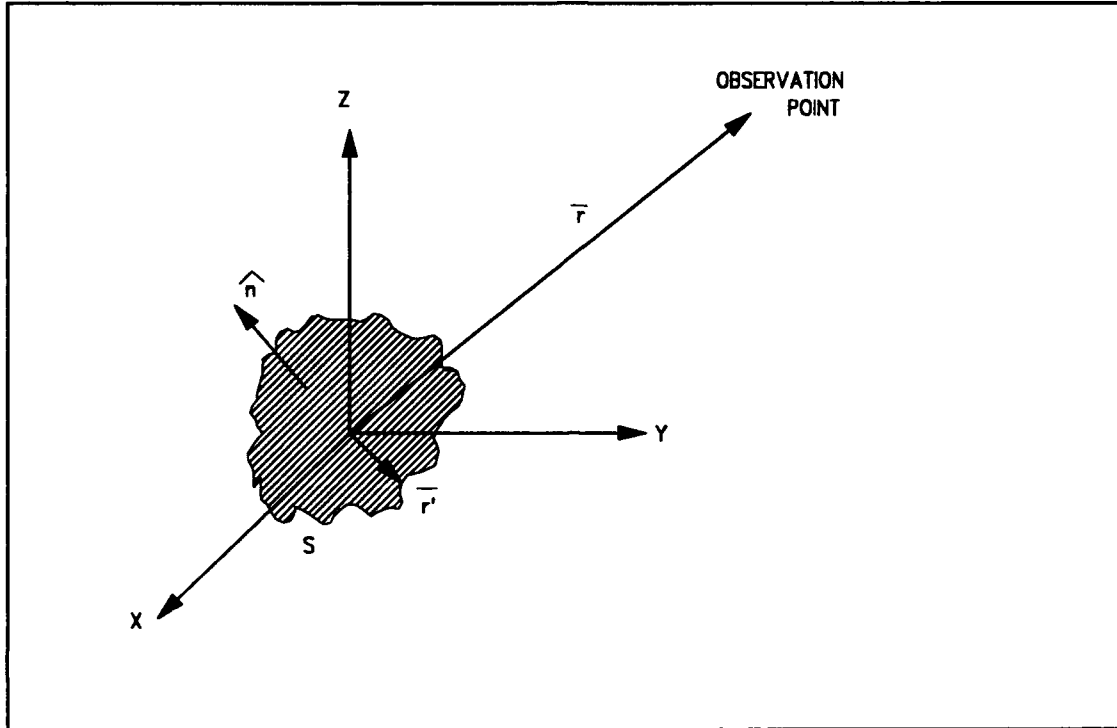


Figure 3.3. Orientation of Source and Observer

For plane wave illumination (a good approximation because the target is in the far zone of the radar) the incident field is given by

$$\vec{H}'(\vec{r}') = \hat{h}_T \exp(-ikr) \exp(ikt\vec{r}') \quad (3.2)$$

where  $\hat{h}_T$  is the magnetic polarization unit vector of the transmitting antenna and  $k$  is the wavenumber.

If the target is a perfect conductor, the far zone scattered field is given by

$$\vec{H}^s(\vec{r}) = \frac{ik}{4\pi r} \exp(-ikr) \int_S \hat{n} \times \vec{H} \exp(ikt\vec{r}') ds' \quad (3.3)$$

where  $\vec{H}$  is the total magnetic field.

Now using the boundary condition for a perfect conductor

$$\vec{J}_s = \hat{n} \times \vec{H} \quad (3.4)$$

where  $\vec{J}_s$  is the surface current distribution, and the Physical Optics approximation

$$\vec{J}_s \approx 2\hat{n} \times \vec{H}^i \quad (3.5)$$

leads to

$$2\hat{n} \times \vec{H}^i = \hat{n} \times \vec{H} \quad (3.6)$$

Substituting Eqs (3.6) and (3.2) into (3.3) gives

$$\vec{H}^s(\vec{r}) = \frac{ik}{2\pi r} \exp(-2ikr) \hat{r} \times \int_s \hat{n} \times \hat{h}_T \exp(2ik\hat{r} \cdot \vec{r}') ds' \quad (3.7)$$

This can be further simplified by using the definition for the SAR vector reflectivity density (15:11),

$$\vec{\gamma}(\vec{r}', \hat{r}) = -\hat{r} \times (\hat{n} \times \hat{h}_T) \quad (3.8)$$

which is related to the SAR scalar reflectivity by (15:11)

$$\gamma(\vec{r}', \hat{r}) = \hat{h}_R \cdot \vec{\gamma}(\vec{r}', \hat{r}) \quad (3.9)$$

Substituting Eq (3.8) into (3.7) gives

$$\vec{H}'(\vec{r}) = -\frac{ik}{2\pi r} \exp(-2ikr) \int_S \vec{\gamma} \exp(2ik\hat{r}\cdot\vec{r}') ds' \quad (3.10)$$

Then taking the dot product of  $\hat{h}_R$  with both sides of Eq (3.10), substituting Eq (3.9) into the results, and dividing by Eq (3.2) gives the RCS as given by Eq (3.1).

### ***Input File Requirements***

The SRIM programs, GIFT and RADSIM, generate the RCS of a target. GIFT does the ray casting, samples the target geometry, and creates the ray history file (\*.ray) and binary representation (\*.4) of the input geometry file. RADSIM then uses the ray history file to calculate the target RCS (written to \*.rcs) and to produce the image file (\*.ci). The image file was not used during this study.

The only input file required to run GIFT is a geometry file. On the other hand, RADSIM requires three externally created input files as shown in Figure 3.4. The following sub-sections give cursory descriptions of the four user supplied input files. The details necessary to construct the files are provided in Appendix A. The actual files used during this thesis are listed in Appendix B.

***Geometry File.*** As mentioned earlier, the targets are modeled with elemental solids called primitives. Boolean operations are used to combine the primitives to make a "region". The regions are then implicitly unioned in GIFT to construct the target. The geometry file gives a geometric description of each primitive, and lists the boolean operations required to make each region. The file name must have a .cg extension.

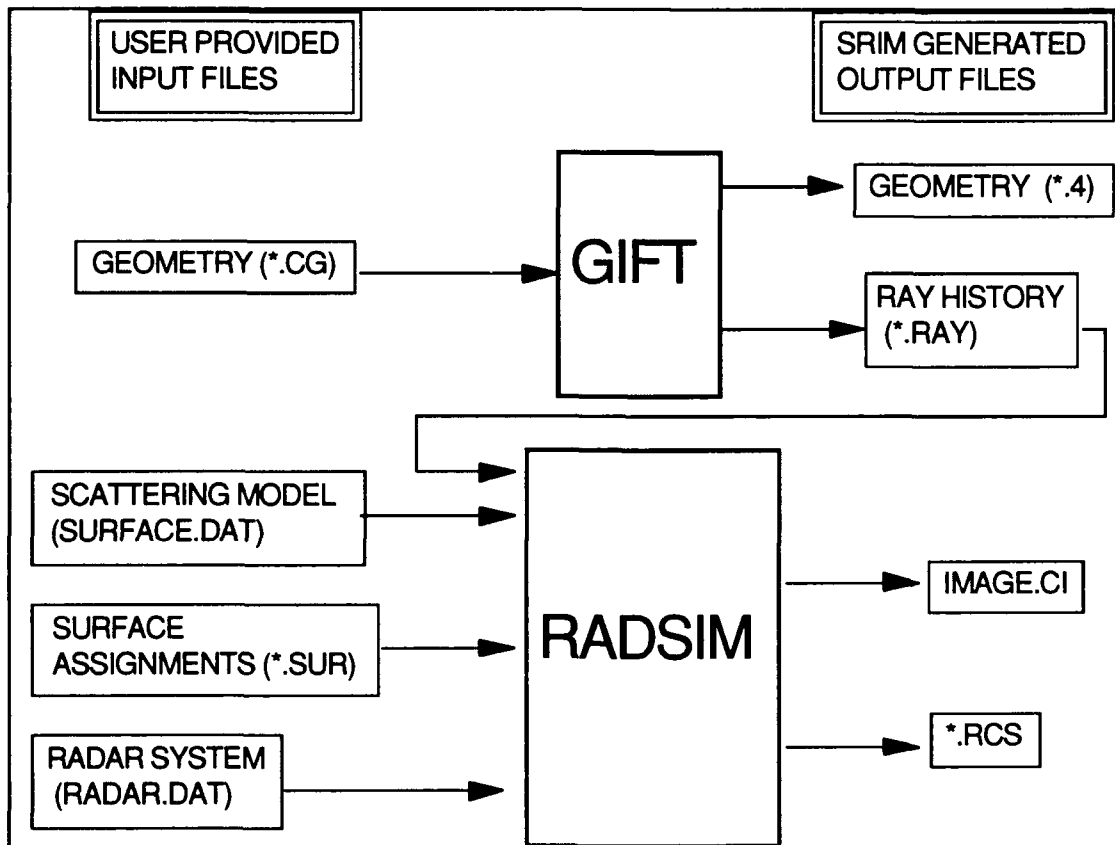


Figure 3.4. GIFT and RADSIM Input/Output Files

To reduce run time, the regions should be as simple as possible. Ideally, a region should consist of only one primitive. However, this is not possible if the boolean differencing operation must be implemented because regions are always implicitly unioned. For example, to construct a hollow cylinder, two right circular cylinders of different radii must be put in one region and differenced.

**Scattering Model File.** This file associates a model number with the type of scattering surface and the percentage of energy absorbed by that surface. The model numbers are then referenced in the surface assignment file which is discussed in the next sub-section. SRIM recognizes three types of scattering surfaces--smooth reflective, smooth finite impedance, and terrain models. In this study only smooth reflective surfaces were used. The file name is surface.dat.

***Surface Assignments File.*** This file is used to assign the scattering models defined in the scattering model file to the surfaces of the primitives in the geometry file. If a surface assignments file is not used, SRIM considers all primitive surfaces perfect reflectors. The file name must have a .sur extension. Other than the extensions, the geometry file and surface assignments file should have the same name.

***Radar System Data File.*** This is a 14 line file containing the parameters that define the radar system. The RCS is independent of most of these parameters, so values for a typical radar system were used during this thesis. The parameters in this file that affect the RCS are the wavelength and receiver and transmitter polarizations. This file is named radar.dat.

### ***Code Automation Technique***

SRIM RCS predictions are generated by sequentially running GIFT and RADSIM. For each run through these programs, the target RCS is calculated for a specific radar LOS. Hence, generating a plot of RCS versus the radar LOS requires running the programs X times, where X corresponds to the number of data points desired. In other words, an azimuth RCS scan from 0 to 90 degrees, in 1° increments, requires running SRIM 91 times. Furthermore, each run through GIFT and RADSIM requires the user to answer 37 SRIM prompts. Thus, collecting the data for a single azimuth scan is a non-trivial task.

However, it was noted that the only input response to SRIM that changed from run to run was the azimuth angle. Therefore, an automation technique was devised that alleviated the problem of extensive user intervention. It uses three FORTRAN programs that run under the control of a UNIX C-shell script file. This

section gives a flow description of the automation technique. In the following, source code names are capitalized and file names are italicized.

Figure 3.5 shows the flow of events as controlled by the script file named RCS.SCRIPT. Not shown are the input and output files that were discussed in conjunction with Figure 3.4. Prior to starting the run, the files required to run SRIM (geometry, surface assignments, and scattering model) and the compiled FORTRAN programs (AZIMUTH.F, AZIMVARY.F, and RCSPLOT.F) must be located in the same directory as RCS.SCRIPT. The radar data file (discussed in the previous section) does not have to be in the directory because it is created by AZIMUTH.F.

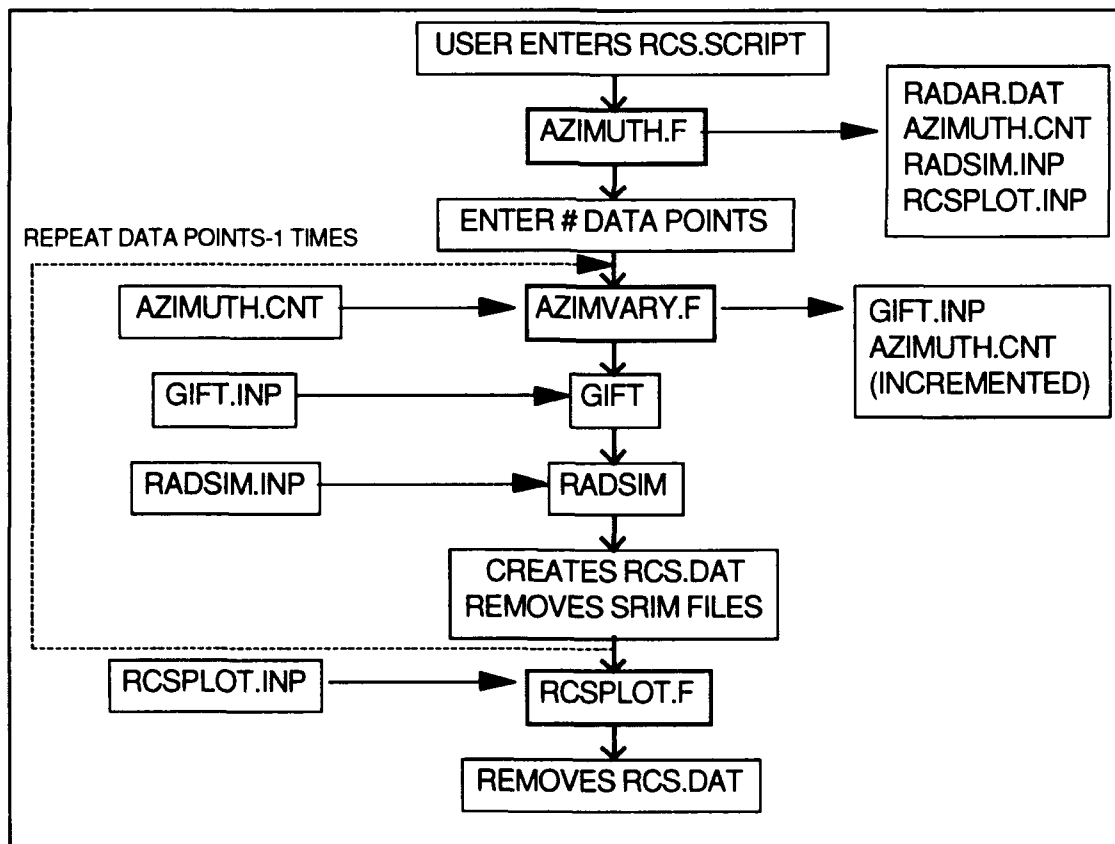


Figure 3.5. RCS.SCRIPT Flow Diagram

To start an azimuth RCS scan, the user initiates RCS.SCRIPT by typing `rcs.script`. This, in turn, starts AZIMUTH.F which prompts the user for the geometry file name, starting and stopping azimuth points, increment between points, elevation angle, radar frequency, ray spacing information, and an output file name for the RCS data. It then creates the files required to automate the azimuth scan. In addition, it calculates and outputs to the screen the total number of data points in the azimuth scan. The script file then prompts the user to enter this integer value. This value is used to set a loop counter in the RCS.SCRIPT file.

Next in the process is the running of AZIMVARY.F. This program creates the prompt responses required to run GIFT. Since the azimuth angle is one of the inputs to GIFT, AZIMVARY.F must be utilized for every run through GIFT. Then, GIFT and RADSIM run in sequence. The prompt responses and radar data file required to run RADSIM were created earlier by AZIMUTH.F.

Continuing, the script file then puts the RCS values created by RADSIM (*\*.rcs*) in a file named *rcs.dat*. In subsequent runs, the RADSIM RCS values are appended to *rcs.dat*. The files created by GIFT and RADSIM during the run (*\*.4*, *\*.ray*, *\*.ci*, and *\*.rcs*) are then removed. The process then loops back up to AZIMVARY and repeats N-1 times, where N is the number of data points.

After N runs, the azimuth scan is completed and the RCS data is in the *rcs.dat* file. However, there is information in the file that is not required for this study. Furthermore, this extraneous data (which includes the date and time the run was completed, ground RCS values, ground/target interaction RCS values, polarization information, etc.) is included for every azimuth position. Therefore, the *rcs.dat* file is read into RCSPLOT.F where only the essential data is retained.

RCS PLOT.F also converts the RADSIM RCS values from units of square meters to dBsm using the relation

$$\sigma_{dBsm} = 10 \log_{10}(\sigma_m) \quad (3.11)$$

and writes the data to a file which was named by the user while running AZIMUTH.F. The *rcs.dat* file is then removed in preparation for the next azimuth scan.

A listing and explanation of each of the prompts required to run GIFT and RADSIM are in Appendix C. Documented source code for the automation technique described above is listed in Appendix D.



## ***IV. Results***

### ***Introduction***

This chapter presents the data used to verify and validate SRIM. As discussed in the methodology section of Chapter 1, the techniques used to verify SRIM consisted of internal validity tests, fixed value tests, and comparison to other models. The technique used to validate SRIM was the comparison of SRIM RCS predictions and measured RCS data. Following this format, this chapter is subdivided into verification and validation sections. Each of these sections is further subdivided into sections that address each technique.

### ***SRIM Verification***

***Internal Validity Tests.*** This phase of the testing was done to ensure the SRIM RCS predictions were repeatable. It served a three-fold purpose. First, it was essential to establish that when identical targets and radar parameters are used, SRIM gives identical results. Second, the effect on the RCS predictions caused by varying radar parameters (the radar.dat file discussed in Chapter 3) that should not affect RCS had to be determined. This step was necessary because SRIM uses several radar parameters that are required to generate a SAR image, but should not affect the RCS of a target. Third, because SRIM samples the geometry with rays, the effect on RCS predictions due to the distance between rays (i.e. ray spacing) or ray density (i.e. number of rays per target) had to be established.

**Results.** SRIM repeatability for identical target and radar parameters was established by analyzing a 10 x 15 cm rectangular flat plate. Three runs were completed using .5 cm ray spacing (RS), 0° elevation (ELEV) angle, and 10 GHz frequency (F). For each run, the plate was oriented in the XZ plane (the long dimension was in the Z direction) but was defined differently with respect to the origin. As seen in Figure 4.1, the results are identical. The traces XZ cen, XZ neg, and XZ pos, are for the cases when the plate is centered at the origin, in the negative XZ plane, and in the positive XZ plane, respectively.

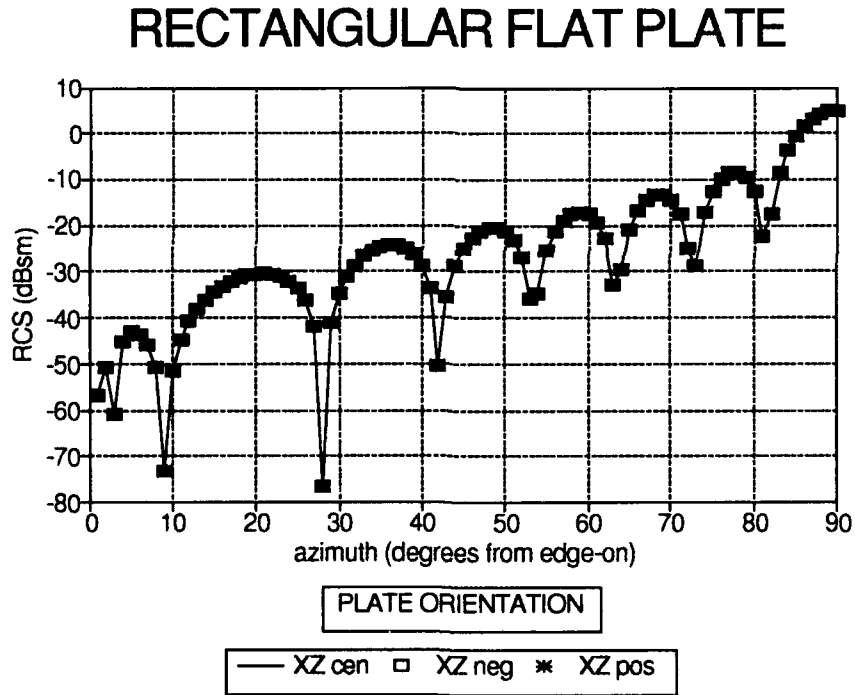


Figure 4.1. Plate in XZ Plane - SRIM RCS Prediction Repeatability, X=10 cm, Z=15 cm, F=10 GHz, Elev=0°, RS=.5 cm

As mentioned in Chapter 3, one of the input files required to run SRIM is the radar system data file (radar.dat). The only parameters of interest in the file from a RCS perspective are the wavelength and polarization. To address the

second purpose of the internal validity tests, the parameters that should not affect RCS were varied. Varying these parameters (listed in Appendix A.4, Lines 6-14) had no effect on the SRIM RCS predictions.

The parameter that has the greatest effect on the SRIM RCS predictions is ray spacing. As an example, Figure 4.2 shows the variation in SRIM RCS predictions for a 5 cm radius sphere as a function of ray spacing. As noted in the figure, the RCS fluctuations are reduced as the distance between rays decreases. In this study, ray spacing was specified as a percentage of a distance  $L$ .  $L$  is the shortest length of the target that is perpendicular to the radar line of sight when the target is normal to the radar. This method was selected to provide for consistency between targets of different sizes. That is, targets of different sizes that have the same ray spacing percentage are sampled with the same number of rays. In the initial phases of this effort, ray spacing was specified as the number of rays per wavelength. However, this method was discarded because it did not take into account the target size. Due to the dependency on ray spacing, several SRIM runs were completed for each target.

At this point, the question may be "Why not use the smallest ray spacing supported by SRIM?" The obvious answer is that denser ray spacing increases the time required to complete a run. For example, Table 4.1 shows the system time required to generate the RCS for four simple primitives at their specular point of return for 5%, 1%, and .5% ray spacing. The runs were completed on a Sun Sparc Station. The time includes both the GIFT and RADSIM system run time as given by the output of the respective modules. See Chapter 3 for more information regarding GIFT and RADSIM. Comparing the run times shows that efficiency requires using the maximum ray spacing that gives acceptable accuracy.

# SPHERE

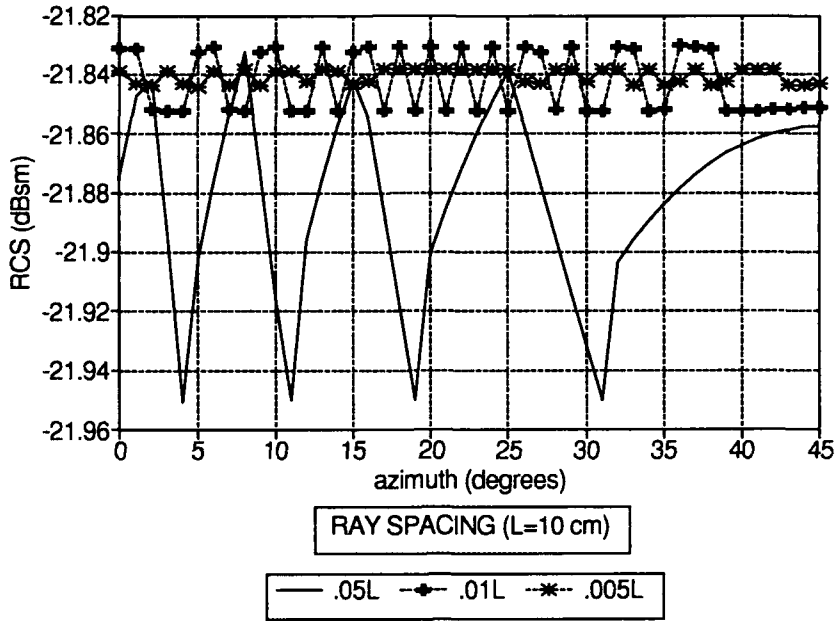


Figure 4.2. Sphere - SRIM RCS Prediction as a Function of Ray Spacing, Radius=5 cm, F=10 GHz, Elev=0°

Table 4.1. Run Times as a Function of Ray Spacing

TARGET	TIME (sec)		
	Ray Spacing (%)		
	5	1	.5
Sphere	1.3	14.5	55.1
Rectangular Plate	1.8	23.1	85.1
Circular Plate	1.2	12.0	46.0
Right Circular Cylinder	1.6	18.7	71.6

**Fixed Value Tests.** In this phase of the study, SRIM monostatic RCS predictions were compared to RCS predictions generated with established Physical

Optic equations. The FORTRAN source code for each of the equations is listed in Appendix E. For each target evaluated, SRIM predictions were generated for several ray spacing distances. The comparisons amply show the effect ray spacing has on the SRIM predictions.

**Sphere Results.** Figure 4.3 shows the comparison between SRIM predictions and the PO equation (17:129) for a 5 cm radius sphere centered at the origin. As already noted in the discussion pertaining to Figure 4.2, ray spacing plays a major role as demonstrated by the fluctuations prevalent in the .05L trace. As the ray density increases, the fluctuations decrease. Aside from the oscillations, the PO and SRIM predictions agree quite well with less than .6 dBsm difference.

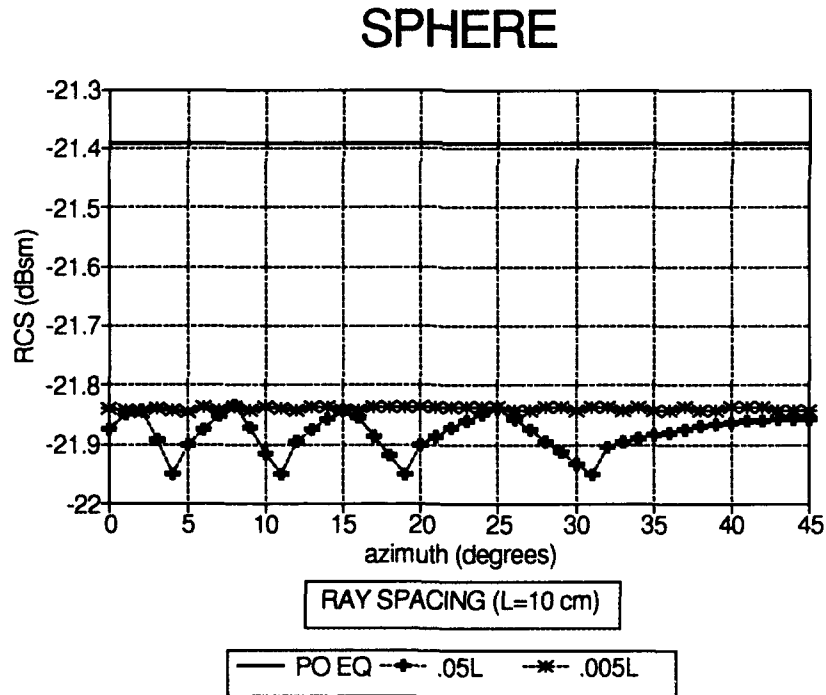


Figure 4.3. Sphere - SRIM & PO RCS Predictions, Radius=5 cm, F=10 GHz, Elev=0°

**Rectangular Flat Plate Results.** As mentioned earlier, SRIM uses Constructive Solid Geometry (CSG) modeling. As such, each primitive is modeled

as a solid with a depth, height, and width. Therefore, a plate must be modeled as a thin box. Consequently, when viewing the plate at grazing incidence, SRIM sees the edge as a narrow plate. This brings up a problem when comparing SRIM and PO predictions because the PO equations assume the plate is infinitesimally thin and do not take into account the thickness of the plate. To overcome this apparent discrepancy, the edge of the box in the SRIM geometry file can be made a perfect absorber by using a \*.sur file discussed in Chapter 3. This is demonstrated in Figure 4.4 which shows the SRIM predictions for a plate with and without absorbing edges and the PO equation (28:3-8) predictions. As seen in the figure, PO and SRIM match well, even at grazing incidence, when the plate edges are perfectly absorbing. Therefore, in the remainder of this study, all SRIM and PO comparisons are made with the edges being perfect absorbers.

Figure 4.5 shows the effect of ray spacing on SRIM predictions for a 10 x 15 cm plate in the XZ plane viewed at 0° elevation (measured from XY plane). As seen in the figure, the match between SRIM and the PO equation (28:3-8) improves as the ray spacing decreases. There are significant differences in the depths of the nulls, but the deep nulls are an artifact of PO and thus, are inconsequential.

Figure 4.6 overlays the PO prediction (left ordinate scale) and the RCS difference (right ordinate scale) between PO and SRIM using 1% ray spacing. As shown, there is an exact match at the positive peaks of each lobe.

Figures 4.7 and 4.8 show the same type of data as the previous two figures but with the viewing angle at 35° from the XY plane. Notice in these figures that the distance between rays was decreased to .005L (.05 cm). The increased ray density was required to maintain a high level of agreement in the lobing structure near grazing incidence. The sampling had to be increased because the projected

area of the target becomes smaller (i.e. projected area = area  $\times$  cos(elevation angle)) as the elevation angle increases. Again, there is an exact match between SRIM and PO at the positive peaks of each lobe.

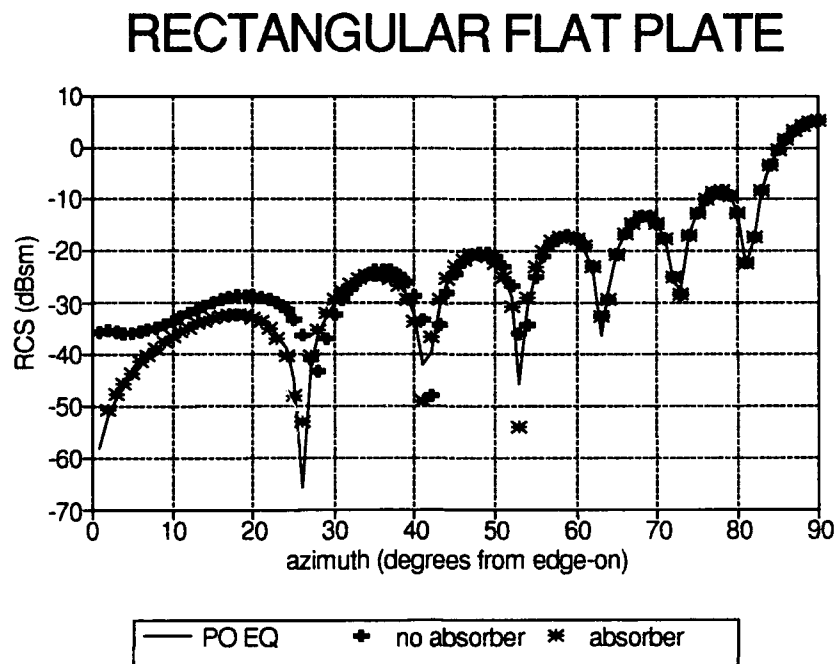


Figure 4.4. Plate in XZ Plane Showing Effect of Absorbing Edges, X=10 cm, Y=1 mm, Z=15 cm, F=10 GHz, Elev=0°, RS=1 mm

## RECTANGULAR FLAT PLATE

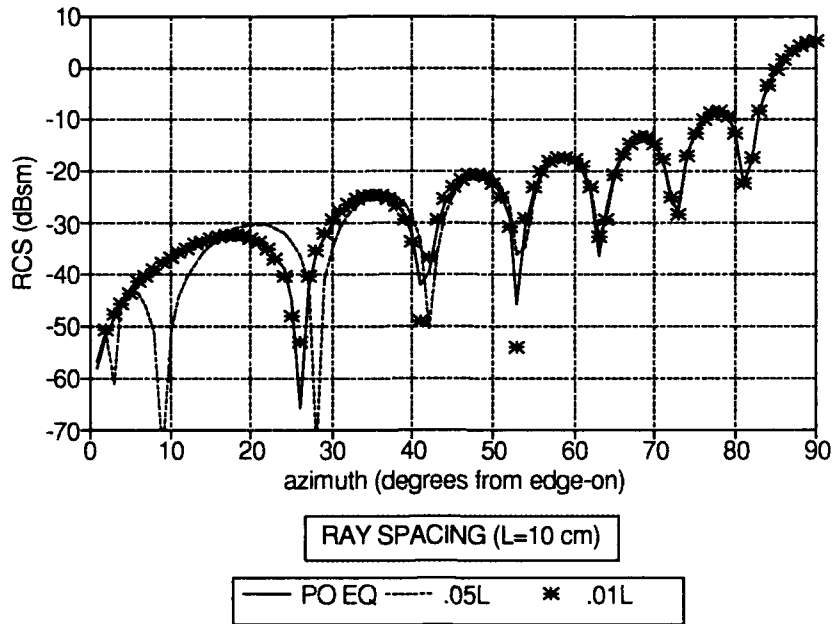


Figure 4.5. Plate in XZ Plane - SRIM & PO RCS Predictions, X=10 cm, Z=15 cm, F=10 GHz, Elev=0°

## RECTANGULAR FLAT PLATE

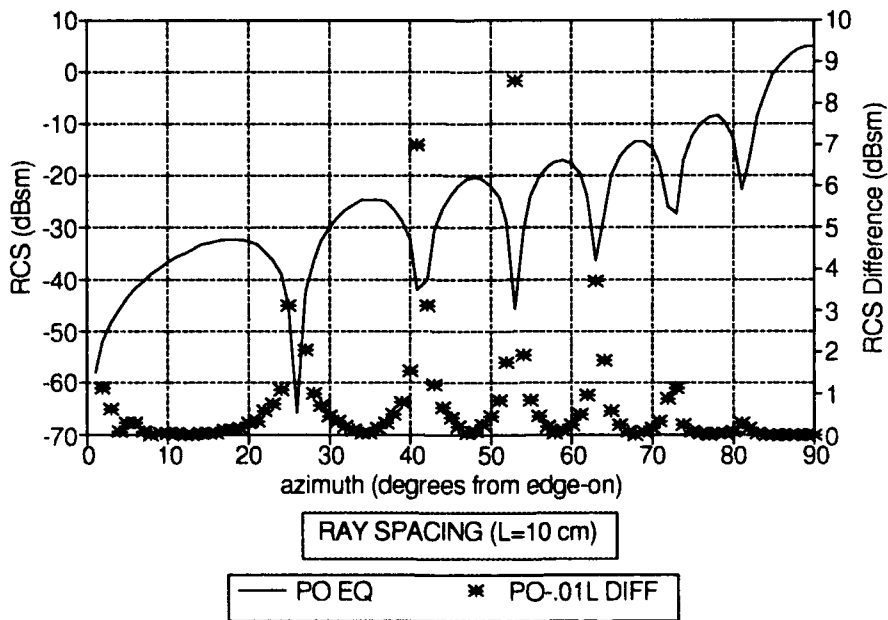


Figure 4.6. Plate in XZ Plane - SRIM & PO RCS Prediction Differences, X=10 cm, Z=15 cm, F=10 GHz, Elev=0°



## RECTANGULAR FLAT PLATE

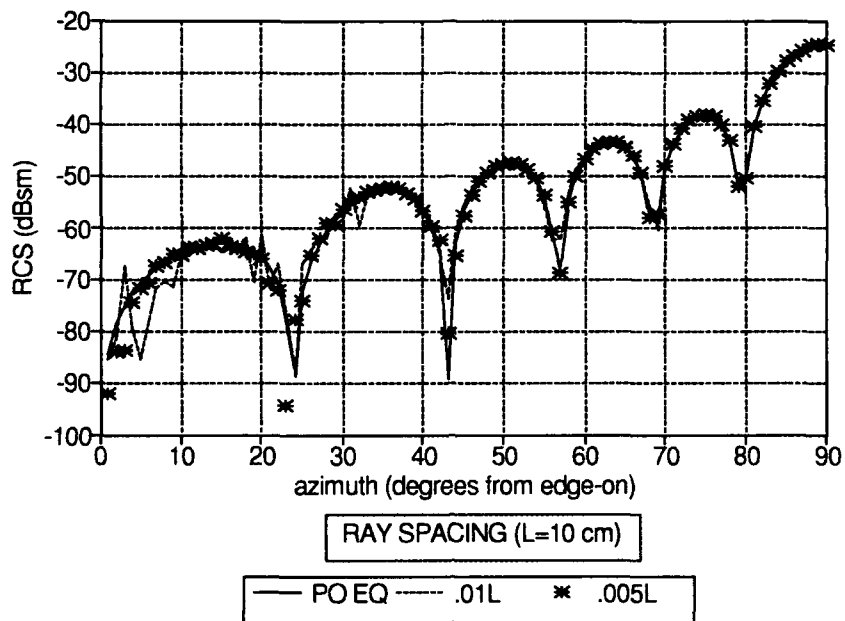


Figure 4.7. Plate in XZ Plane - SRIM & PO RCS Predictions, X=10 cm, Z=15 cm, F=10 GHz, Elev=35°

## RECTANGULAR FLAT PLATE



Figure 4.8. Plate in XZ Plane - SRIM & PO RCS Prediction Differences, X=10 cm, Z=15 cm, F=10 GHz, Elev=35°

***Circular Disk Results.*** Figure 4.9 shows the comparison between SRIM and PO equation (25:513) predictions for a 5 cm radius circular flat plate oriented in the XZ plane at 0° elevation. Again, it is quite evident that the agreement is better as the distance between rays decreases. Figure 4.10 quantifies the disagreements. The right ordinate shows that the differences between SRIM and PO are minor except at the null points and grazing incidence.

Figures 4.11 and 4.12 shows the comparison between PO and SRIM predictions for the plate when viewed from 35° elevation. Again, as in the case of the rectangular plate, as the viewing angle increases, the distance between rays has to be decreased to maintain agreement near grazing incidence.

# CIRCULAR PLATE

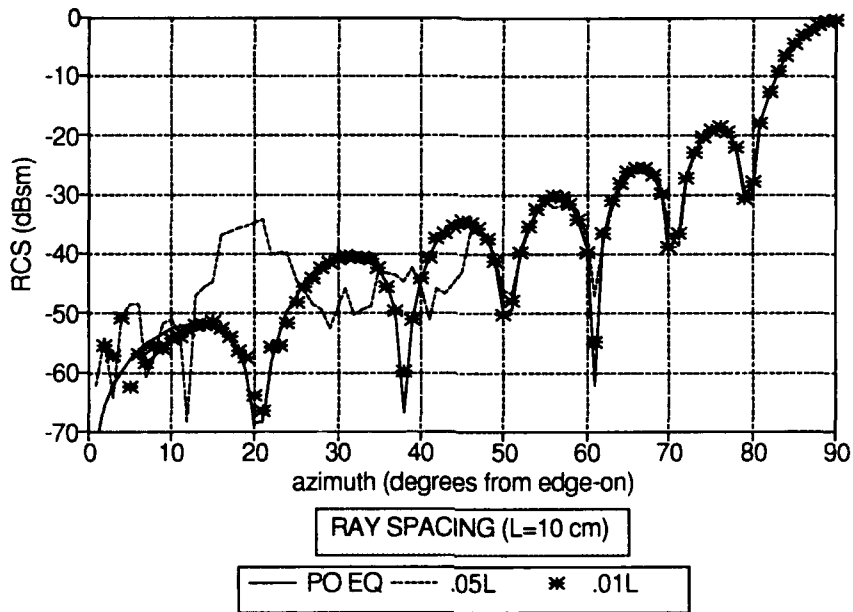


Figure 4.9. Disk in XZ Plane - SRIM & PO RCS Predictions, Radius=5 cm, F=10 GHz, Elev=0°

# CIRCULAR PLATE

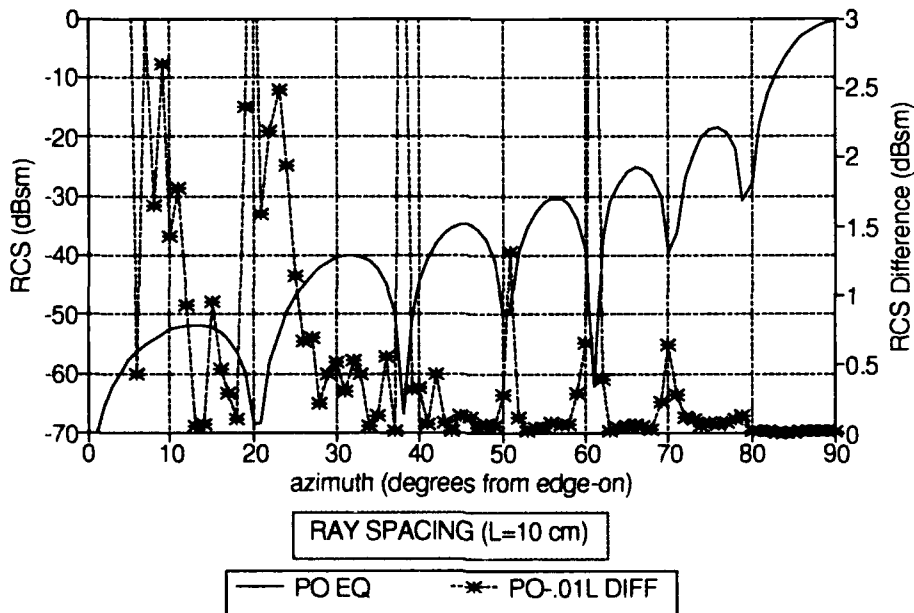


Figure 4.10. Disk in XZ Plane - SRIM & PO RCS Prediction Differences, Radius=5 cm, F=10 GHz, Elev=0°

# CIRCULAR PLATE

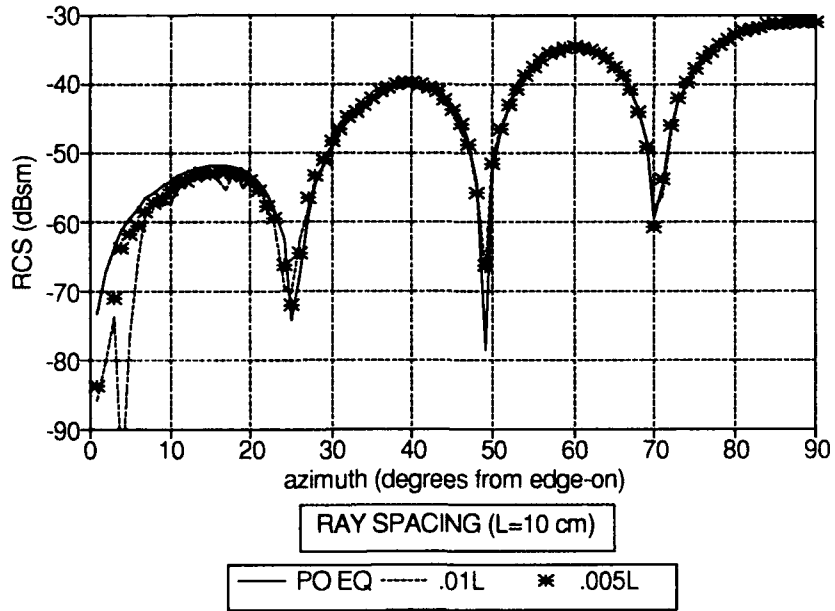


Figure 4.11. Disk in XZ Plane - SRIM & PO RCS Predictions, Radius=5 cm, F=10 GHz, Elev=35°

# CIRCULAR PLATE

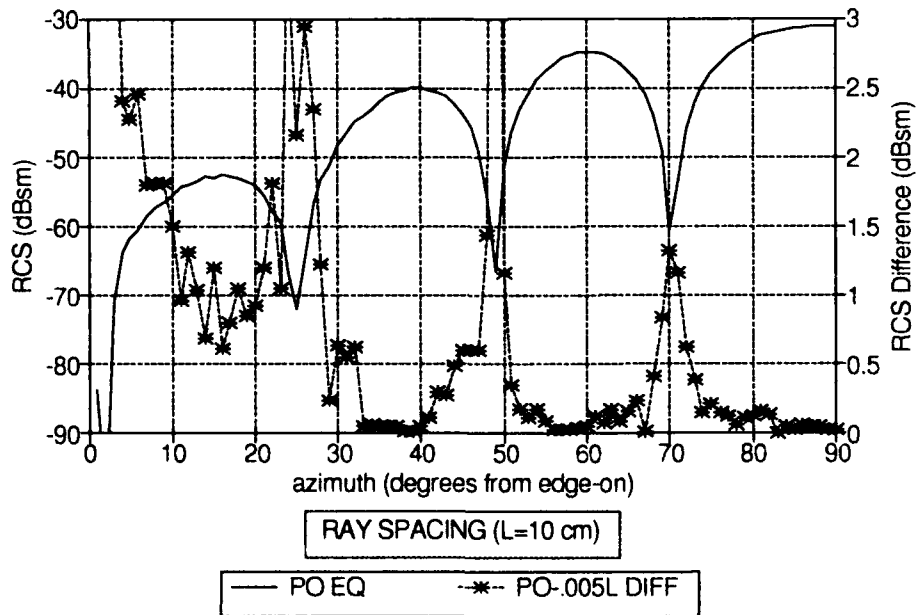


Figure 4.12. Disk in XZ Plane - SRIM & PO RCS Prediction Differences, Radius=5 cm, F=10 GHz, Elev=35°

***Solid Right Circular Cylinder.*** Figure 4.13 shows the comparison between SRIM and the PO equation (17:126; 25:513) predictions for a solid cylinder axially positioned along the Y axis at 0° elevation. The cylinder radius and length are 10.16 cm and 20.32 cm, respectively. The predictions are identical when broadside to either the end cap or cylinder. The differences in the 30° to 60° region were expected because the PO equations were blended. That is, the contribution from the cylinder and disk were not coherently summed (difference in phase was not taken into account). Rather, the PO equation for a cylinder was used for angles between 0° and 45° and the PO equation for a disk was used for angles between 46° and 90°. SRIM does not use this approach. As explained in Chapter 3, SRIM calculates the RCS by coherently summing the contribution from each surface patch. Coherent summation is the more accurate of the two approaches. Figure 4.14 quantifies the differences between SRIM and PO with the scale on the right ordinate. Ignoring the 30° to 60° region, agreement is within 2.5 dBsm.

Figures 4.15 and 4.16 show the same type of comparisons, the only difference being the elevation angle of 35°. Notice that even with the blended approach, the differences between SRIM and PO predictions are less than 2 dBsm at most of the positive peaks.

## RIGHT CIRCULAR CYLINDER

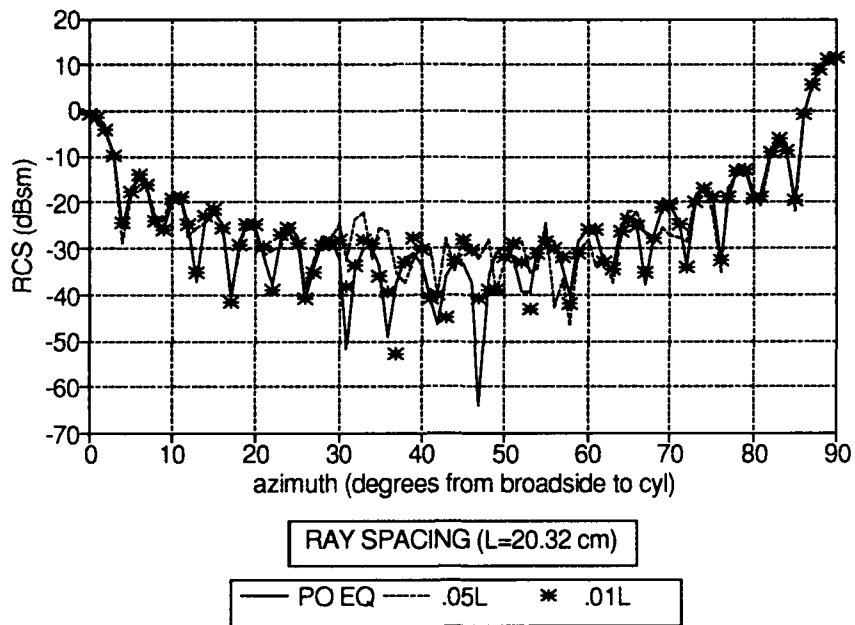


Figure 4.13. RCC Along Y Axis - SRIM & PO RCS Predictions, Radius=10.16 cm, Length=20.32 cm, F=10 GHz, Elev=0°

## RIGHT CIRCULAR CYLINDER

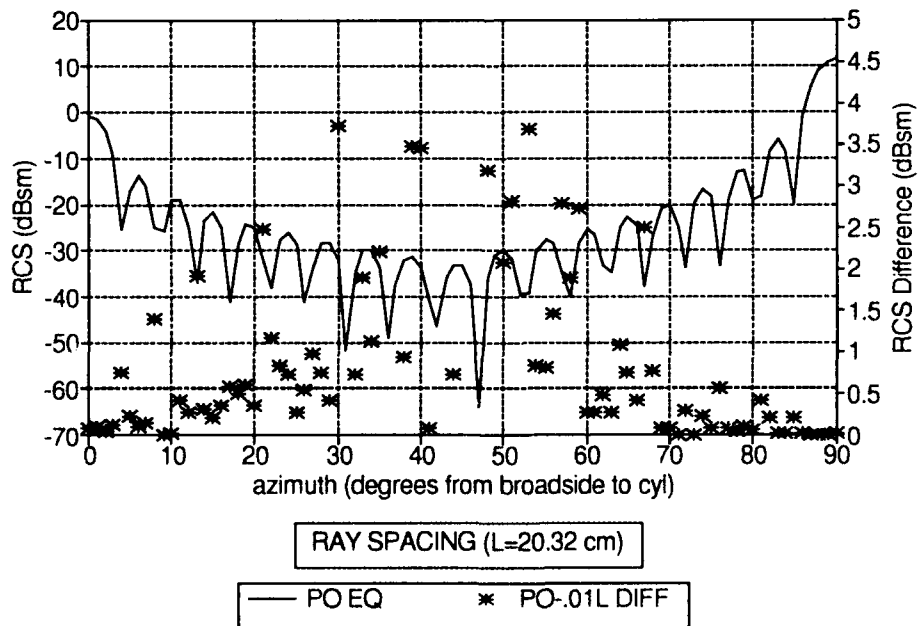


Figure 4.14. RCC Along Y Axis - SRIM & PO RCS Difference, Radius=10.16 cm, Length=20.32 cm, F=10 GHz, Elev=0°

## RIGHT CIRCULAR CYLINDER

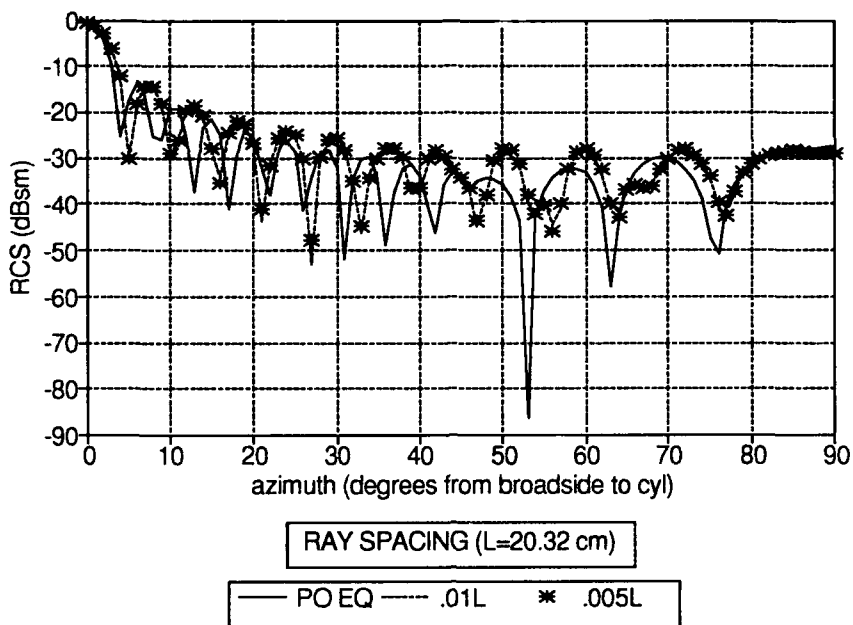


Figure 4.15. RCC Along Y Axis - SRIM & PO RCS Predictions, Radius=10.16 cm, Length=20.32 cm, F=10 GHz, Elev=35°

## RIGHT CIRCULAR CYLINDER

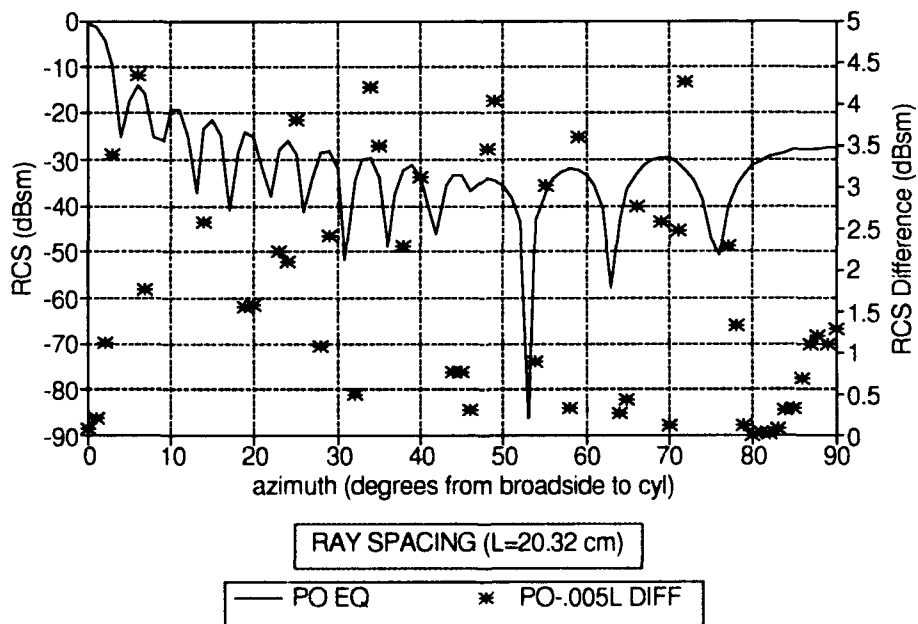


Figure 4.16. RCC Along Y Axis - SRIM & PO RCS Differences, Radius=10.16 cm, Length=20.32 cm, F=10 GHz, Elev=35°

***Right Dihedral Results.*** PO predictions are generally polarization insensitive, that is, they do not take into account the wave polarization. However, this is not true when the signal undergoes an even number of reflections before returning to the observer because the electric field of a vertically polarized wave undergoes a  $180^\circ$  phase shift at each perfectly conducting boundary. Thus, the dihedral response is polarization sensitive because the return consists not only of the single bounce response from each plate, but also a double bounce response due to the interaction between the plates. Therefore, this section makes the stipulation between polarizations. The nomenclature HH and VV imply the transmitter and receiver are both either horizontally or vertically polarized, respectively.

Figures 4.17 and 4.18 compare SRIM and PO equation (8:3) RCS predictions for a 10 x 15 cm right dihedral at  $0^\circ$  elevation for HH and VV polarization, respectively. The dihedral crease (15 cm) is aligned along the Z axis. The vertical plates are in the XZ and YZ planes. The right ordinate quantifies the differences between predictions. As shown, there is excellent agreement for both polarizations. When using ray spacing of .01L (1 mm) the differences are less than .25 dBsm even at the null points. The reason for the excellent agreement is because the PO equation double bounce contribution was derived using GO, which is essentially the same manner used in SRIM to account for multiple reflections. To reiterate, SRIM uses Physical Optics to calculate the RCS, but uses Geometrical Optics (ray tracing) to define the field illumination for multiple reflections.

Figures 4.19 and 4.20 shows the same type of comparisons as Figures 4.17 and 4.18, but from an elevation angle of  $35^\circ$ . Even with finer ray spacing, the disagreements are greater than in the  $0^\circ$  elevation cases. For HH polarization the



disagreements are less than 1 dBsm near the region of maximum return. The disparities are greater for VV polarization, but still less than 3 dBsm.

***Square Trihedral Results.*** The trihedral has single, double, and triple bounce responses. Thus, its response is polarization sensitive (due to the three double bounces). Figures 4.21 through 4.24 show the comparisons between SRIM and PO equation (9; 10) RCS predictions for HH and VV polarizations for a 10 cm right trihedral at 0° and 35° elevation. The vertical crease is aligned along the Z axis. The plates are oriented in the XZ, YZ, and XY planes. There are large disagreements in the null depths. But generally, the differences are less than 1 dBsm in the region of interest.

## RIGHT DIHEDRAL

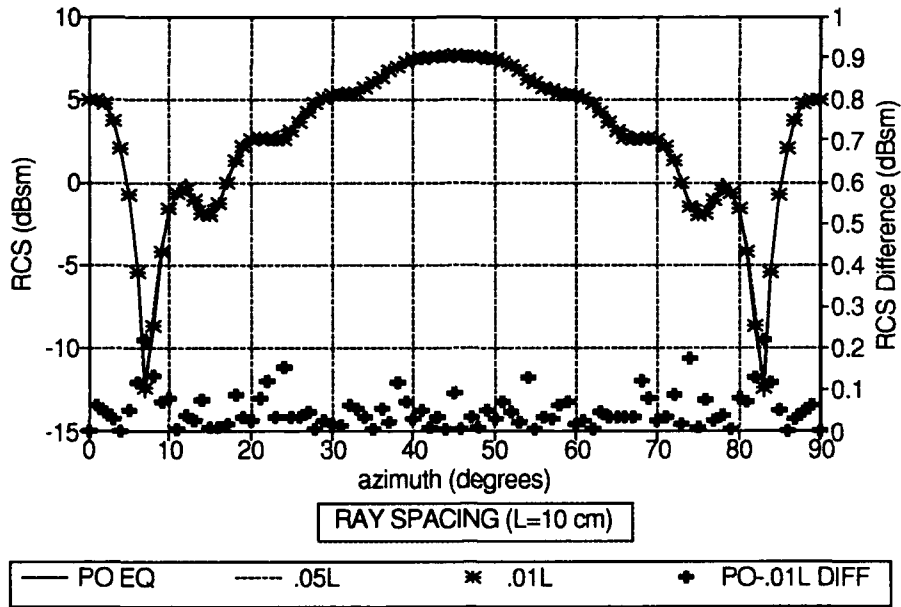


Figure 4.17. Right Dihedral - SRIM & PO RCS Predictions, X=Y=10 cm, Z=15 cm, F=10 GHz, P=HH, Elev=0°

## RIGHT DIHEDRAL

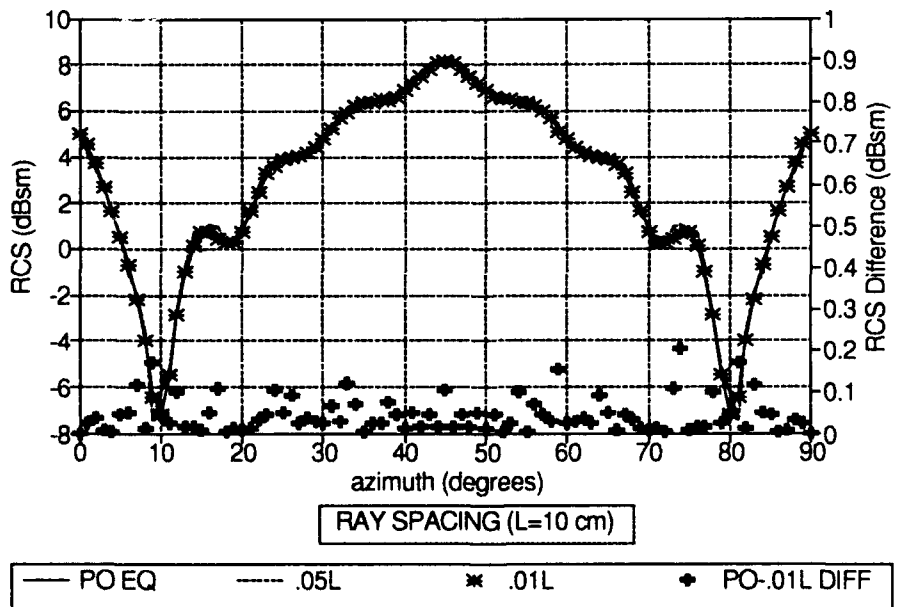


Figure 4.18. Right Dihedral - SRIM & PO RCS Predictions, X=Y=10 cm, Z=15 cm, F=10 GHz, P=VV, Elev=0°

## RIGHT DIHEDRAL

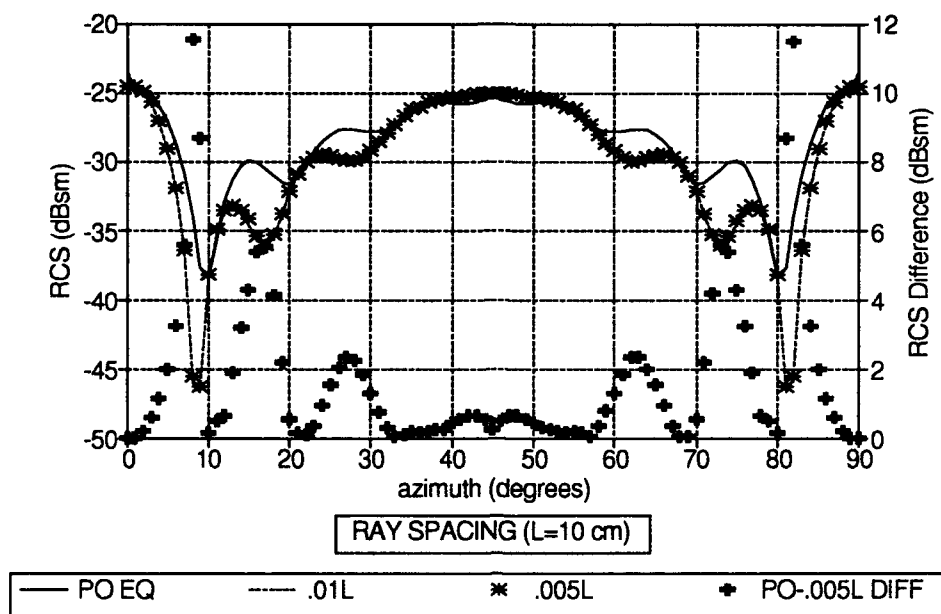


Figure 4.19. Right Dihedral - SRIM & PO RCS Predictions, X=Y=10 cm, Z=15 cm, F=10 GHz, P=HH, Elev=35°

## RIGHT DIHEDRAL

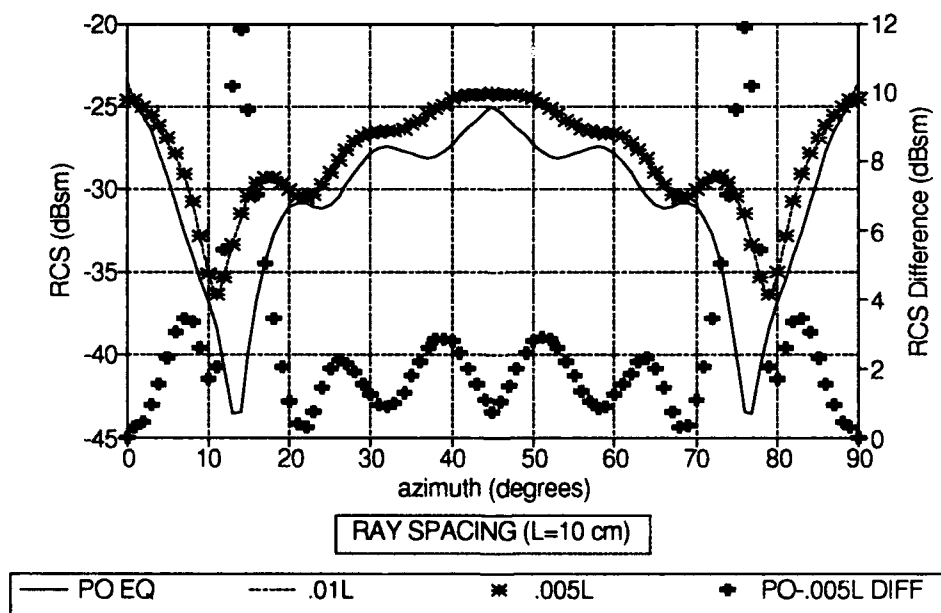


Figure 4.20. Right Dihedral - SRIM & PO RCS Predictions, X=Y=10 cm, Z=15 cm, F=10 GHz, P=VV, Elev=35°

# SQUARE TRIHEDRAL

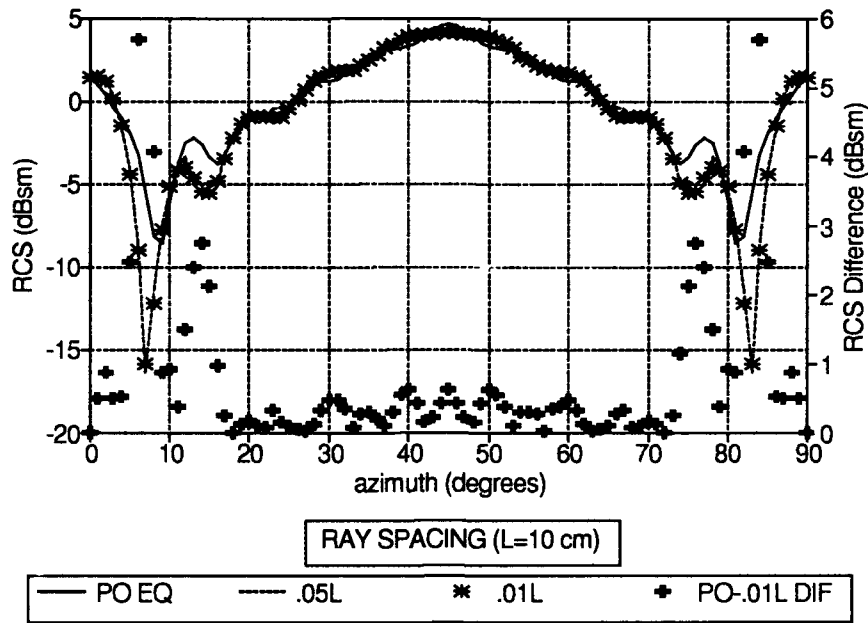


Figure 4.21. Square Trihedral - SRIM & PO RCS Predictions, X=Y=Z=10 cm, F=10 GHz, P=HH, Elev=0°

# SQUARE TRIHEDRAL

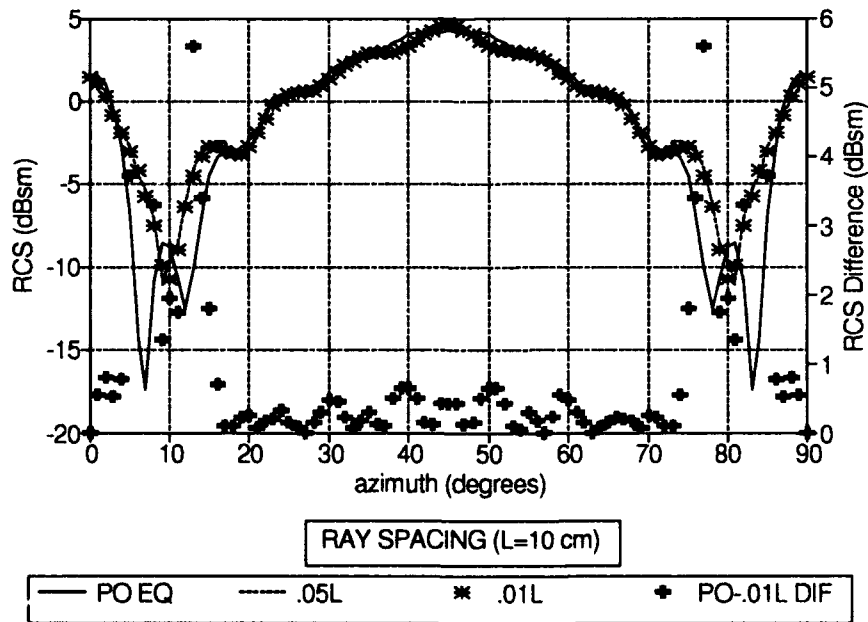


Figure 4.22 Square Trihedral - SRIM & PO RCS Predictions, X=Y=Z=10 cm, F=10 GHz, P=VV, Elev=0°

## SQUARE TRIHEDRAL

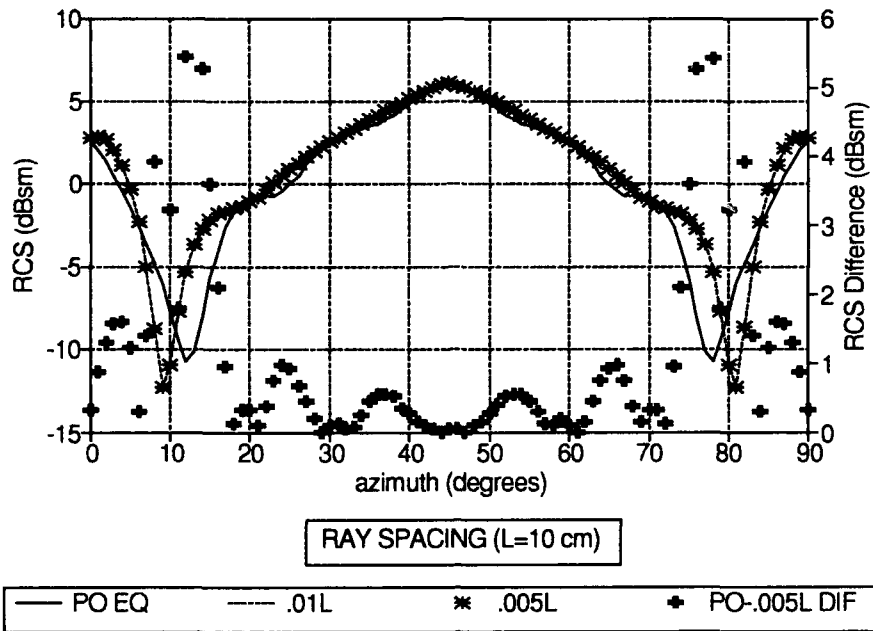


Figure 4.23. Square Trihedral - SRIM & PO RCS Predictions, X=Y=Z=10 cm, F=10 GHz, P=HH, Elev=35°

## SQUARE TRIHEDRAL

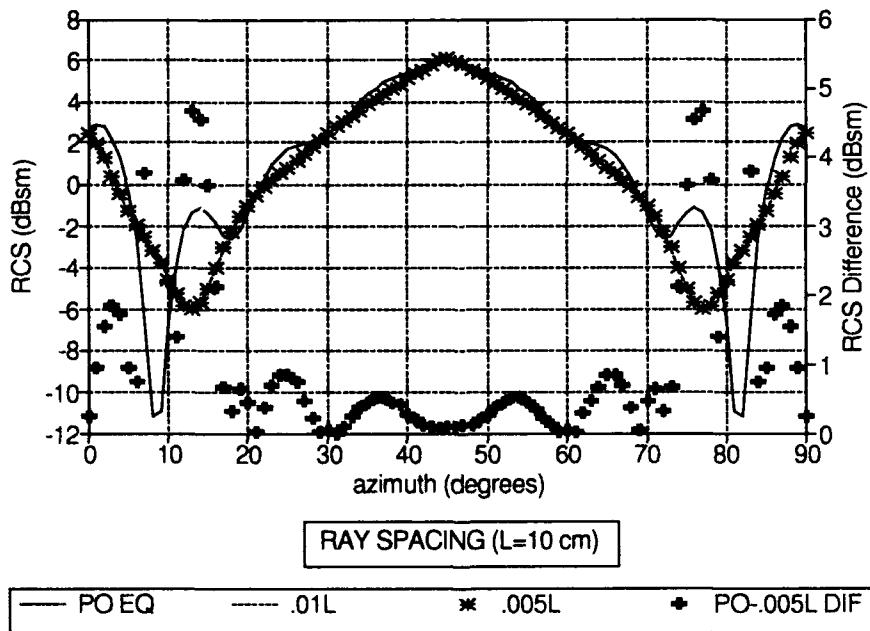


Figure 4.24. Square Trihedral - SRIM & PO RCS Predictions, X=Y=Z=10 cm, F=10 GHz, P=VV, Elev=35°

**Top Hat Results.** Physically, a top hat is constructed of a solid right circular cylinder which is axially positioned perpendicular to a circular disk. In this study, the disk radius was 20.32 cm (8"), the cylinder radius was 10.16 cm (4"), and the cylinder length was 20.32 cm. These measurements were selected because Blejer (11:858) presents Ohio State University experimental results for this configuration. The comparison between SRIM predictions and the Ohio State data is shown in the validation section of this chapter.

A top hat is polarization sensitive because there is a double bounce response between the cylinder and circular base. Therefore, both HH and VV polarizations predictions are presented within this section. Figures 4.25 and 4.26 show the SRIM and PO equation (11:858) RCS predictions for a top hat with the cylinder axially positioned along the X axis at 0° elevation. As shown, the only exact agreement occurs when broadside to the cylinder (90°). Greater disagreements are prevalent in the 70° to 85° region (HH case) and the 7° to 12° region (VV case) because of differences in the lobing structures. Other than the null point differences, the disagreement is approximately 4 dBsm across the entire scan.

## TOP HAT

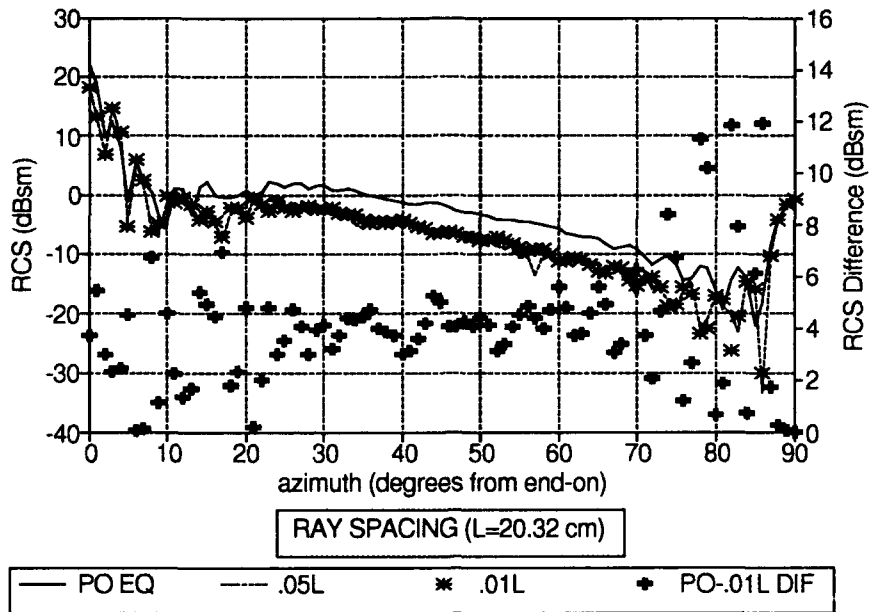


Figure 4.25. Top Hat - SRIM & PO RCS Predictions, Disk Radius=20.32 cm, Cyl Radius=10.16 cm, Cyl Length=20.32 cm, F=10 GHz, P=HH, Elev=0°

## TOP HAT

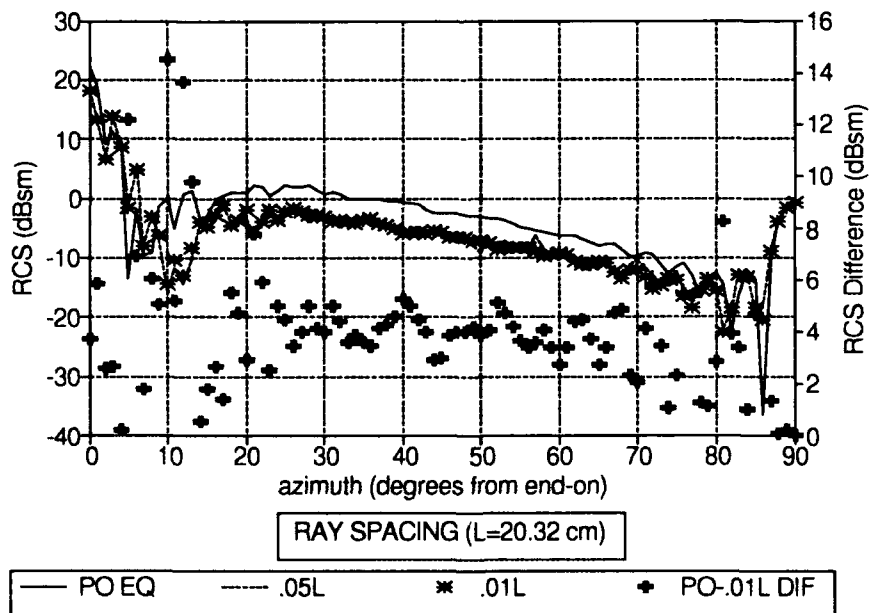


Figure 4.26. Top Hat - SRIM & PO RCS Predictions, Disk Radius=20.32 cm, Cyl Radius=10.16 cm, Cyl Length=20.32 cm, F=10 GHz, P=VV, Elev=0°

**Comparison to Other Models.** In this section, SRIM, Denmar Backscatter Signatures Method (DBSM), and SarTool RCS predictions are compared.

**DBSM and SRIM RCS Prediction Comparisons.** Figures 4.27 through 4.30 show the comparison for rectangular and circular plates. The left scale gives the RCS in units of dBsm. The right scale gives the difference in dBsm between the two RCS predictions. Except for the null depths, there is excellent agreement.

Figures 4.31 and 4.32 show the comparison for a solid circular cylinder at  $0^\circ$  and  $35^\circ$  elevation angles. There is excellent agreement when the angle of incidence is within  $20^\circ$  of the cylinder or end cap normal. However, a slight lobing structure difference between  $20^\circ$  and  $70^\circ$  gives large magnitude differences in this region.

Figure 4.33 shows the comparison for a right dihedral with HH polarization. As is quite evident, there is a large disparity between the predictions. The reason for this is because DBSM does not account for the interactions between the plates (i.e. double bounce response). As such, it cannot be used to compare predictions for the dihedral, trihedral, or top hat.



## RECTANGULAR FLAT PLATE

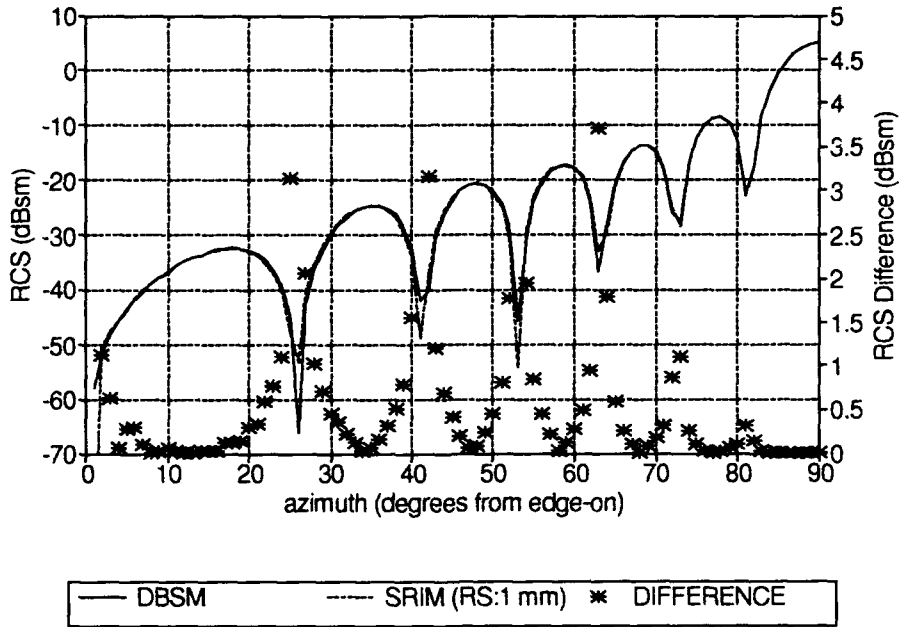


Figure 4.27. Plate in XZ Plane - SRIM & DBSM RCS Predictions, X=10 cm, Z=15 cm, F=10 GHz, Elev=0°

## RECTANGULAR FLAT PLATE

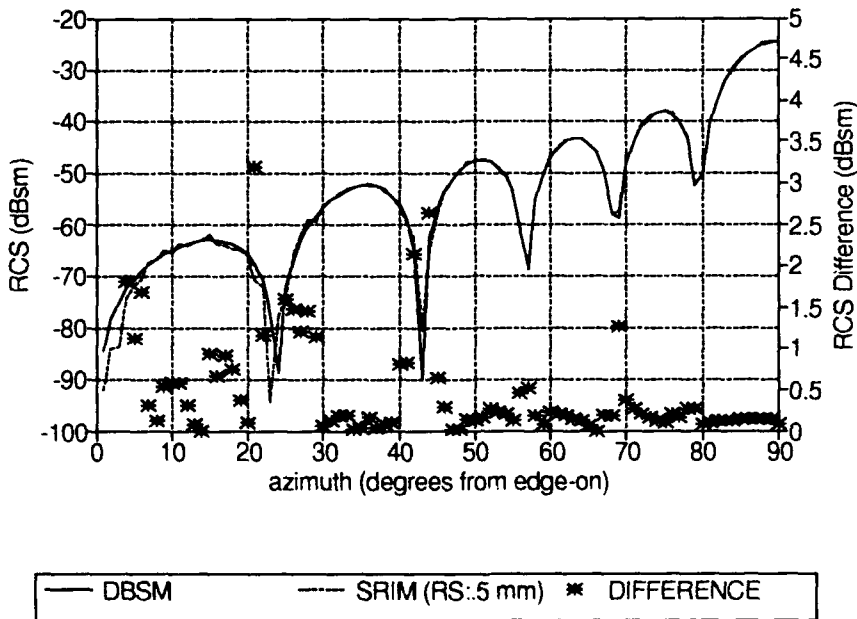


Figure 4.28. Plate in XZ Plane - SRIM & DBSM RCS Predictions, X=10 cm, Z=15 cm, F=10 GHz, Elev=35°

## CIRCULAR PLATE

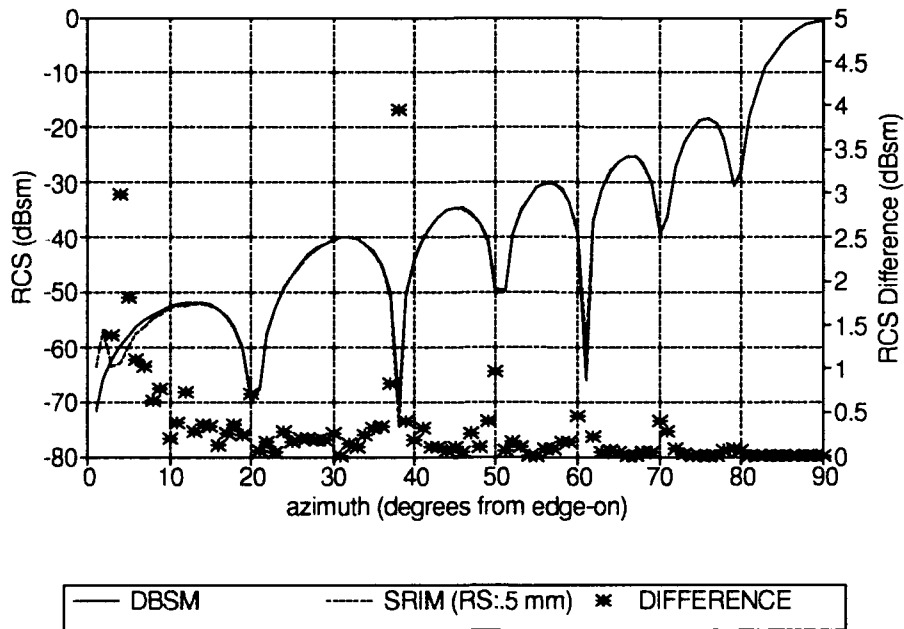


Figure 4.29. Disk in XZ Plane - SRIM & DBSM RCS Predictions, Radius=5 cm, F=10 GHz, Elev=0°

## CIRCULAR PLATE

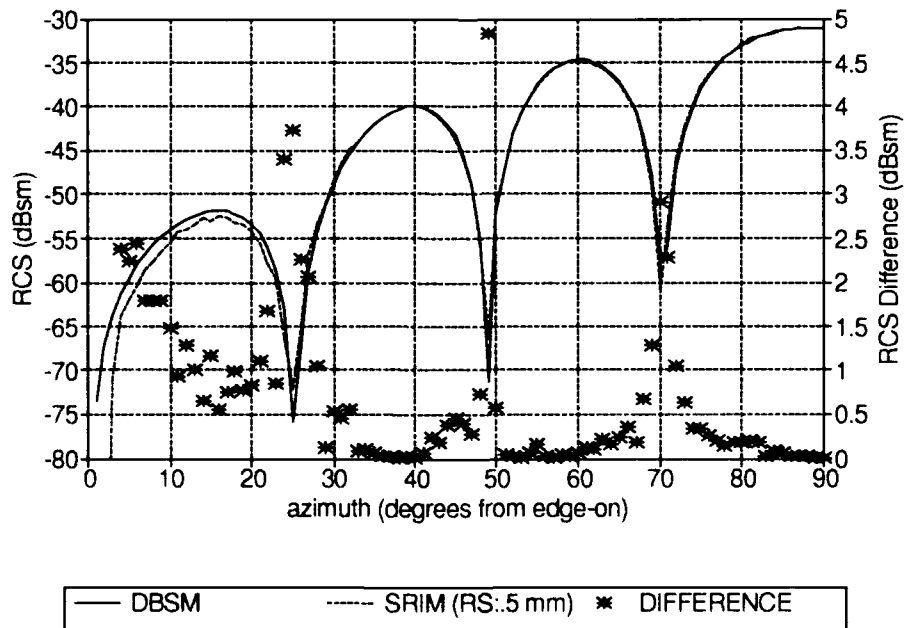


Figure 4.30. Disk in XZ Plane - SRIM & DBSM RCS Predictions, Radius=5 cm, F=10 GHz, Elev=35°

## RIGHT CIRCULAR CYLINDER

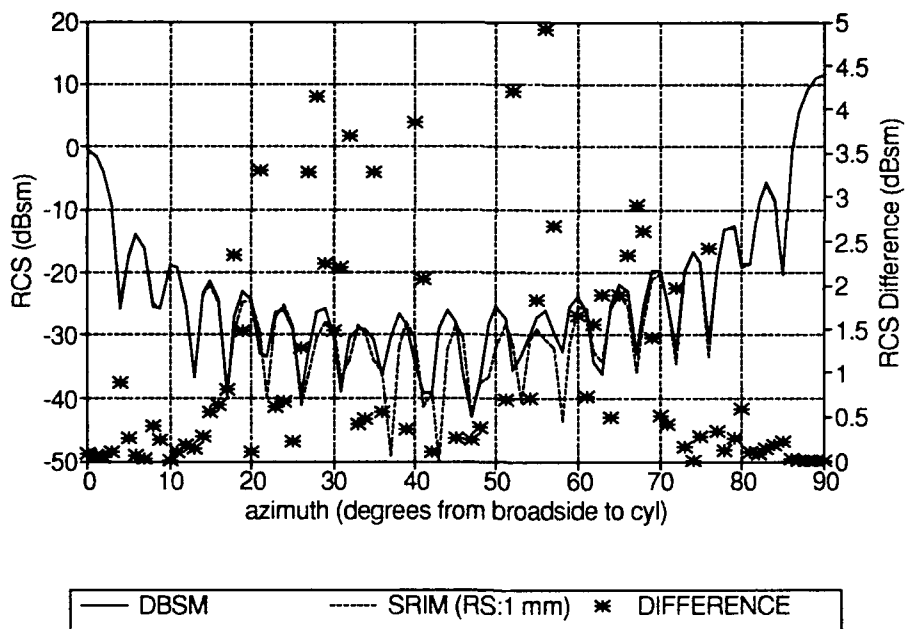


Figure 4.31. Cylinder Along Y Axis - SRIM & DBSM RCS Predictions, Radius=10.16 cm, Length=20.32 cm, F=10 GHz, Elev=0°

## RIGHT CIRCULAR CYLINDER

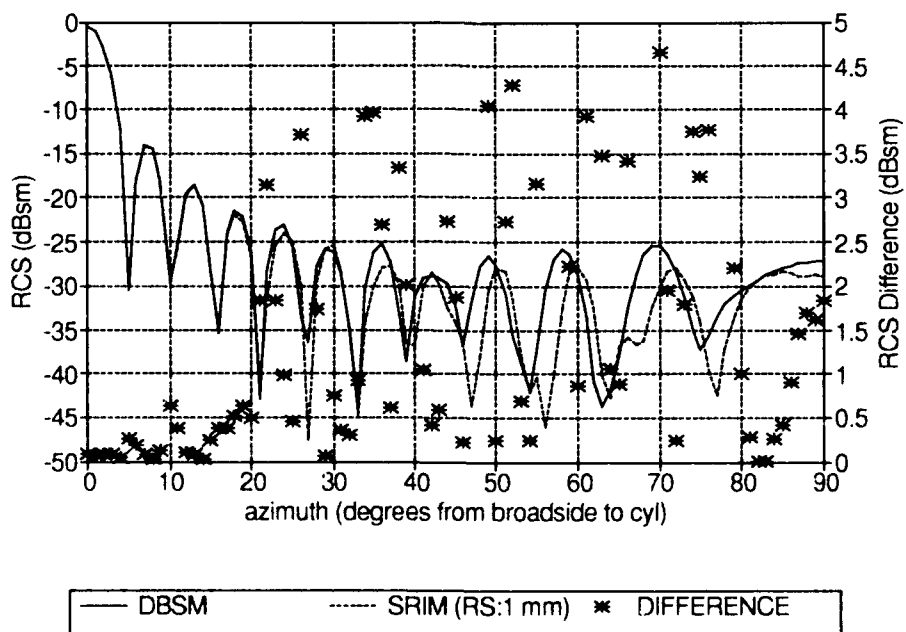


Figure 4.32. Cylinder Along Y Axis - SRIM & DBSM RCS Predictions Radius=10.16 cm, Length=20.32 cm, F=10 GHz, Elev=35°

# RIGHT DIHEDRAL

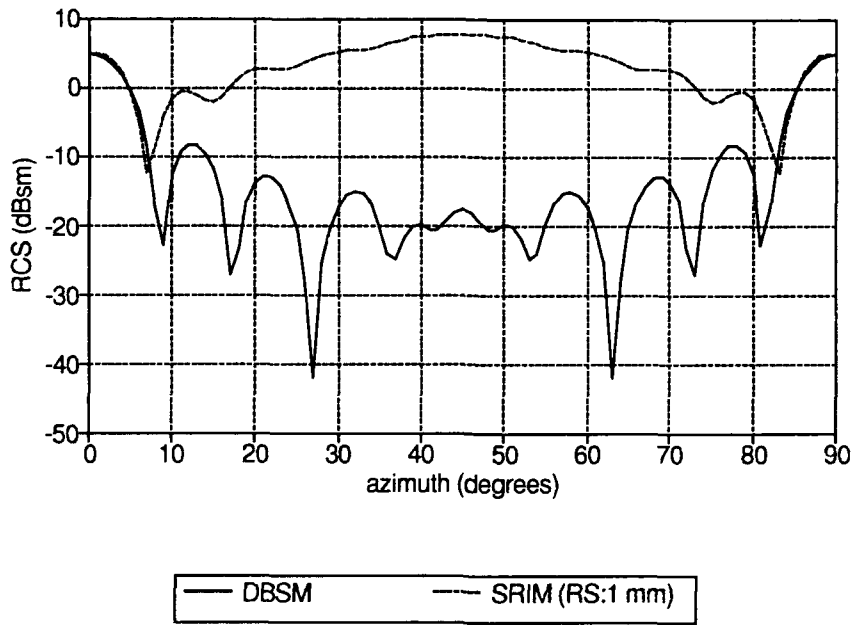


Figure 4.33. Right Dihedral - SRIM & DBSM RCS Predictions, X=Y=10 cm, Z=15 cm, F=10 GHz, P=HH, Elev=0°

***SarTool and SRIM RCS Prediction Comparisons.*** These codes were used to compare RCS predictions for the plates, dihedral, trihedral, and top hat. The SarTool figures were taken from Capt Young's thesis (28). The target geometries and ordinate scales for each of the comparisons are identical. However, there is a slight difference in the  $0^\circ$  elevation plots. SarTool does not have the capability for a  $0^\circ$  look angle so the SarTool predictions were generated with a  $.1^\circ$  elevation angle.

Figures 4.34 through 4.37 show the comparisons for the flat plate at  $0^\circ$  ( $.1^\circ$  for SarTool) and  $35^\circ$  elevation. The only differences noted are the null depths.

Figures 4.38 through 4.41 show similar predictions for the circular plate. Again, the only differences are the null depths.

Figures 4.42 through 4.45 compare the predictions for a right dihedral. There are disagreements of notable interest. For instance, the polarizations seem to be reversed. That is, the SRIM prediction for HH polarization more closely resembles the SarTool VV predictions and vice versa. Also, the null depths are different and the SRIM maximum return for the  $35^\circ$  elevation case is 8 dBsm lower than predicted by SarTool. These results reaffirm Young's (28) conclusion that the SarTool predictions for the dihedral are in error.

Figures 4.46 through 4.49 are the comparisons for a square trihedral. Again, the polarizations are apparently reversed. Other than the polarization reversal, the predictions generally agree.

Figures 4.50 and 4.51 show the top hat predictions. The only location the predictions match is broadside to the cylinder ( $90^\circ$ ). Otherwise, the differences are approximately 5 dBsm across the azimuth range. Young concluded that SarTool does not calculate the RCS due to the base plate disk. This explains the

differences between SRIM and SarTool near  $0^\circ$  (broadside to disks or end-on), but does not explain the 5 dBsm difference between  $20^\circ$  and  $70^\circ$ . Since the PO equation and SRIM RCS prediction comparisons showed similar differences, the problem is evidently within SRIM. It indicates there may be a problem with the way SRIM accounts for the multiple bounce between a curved and flat surface.

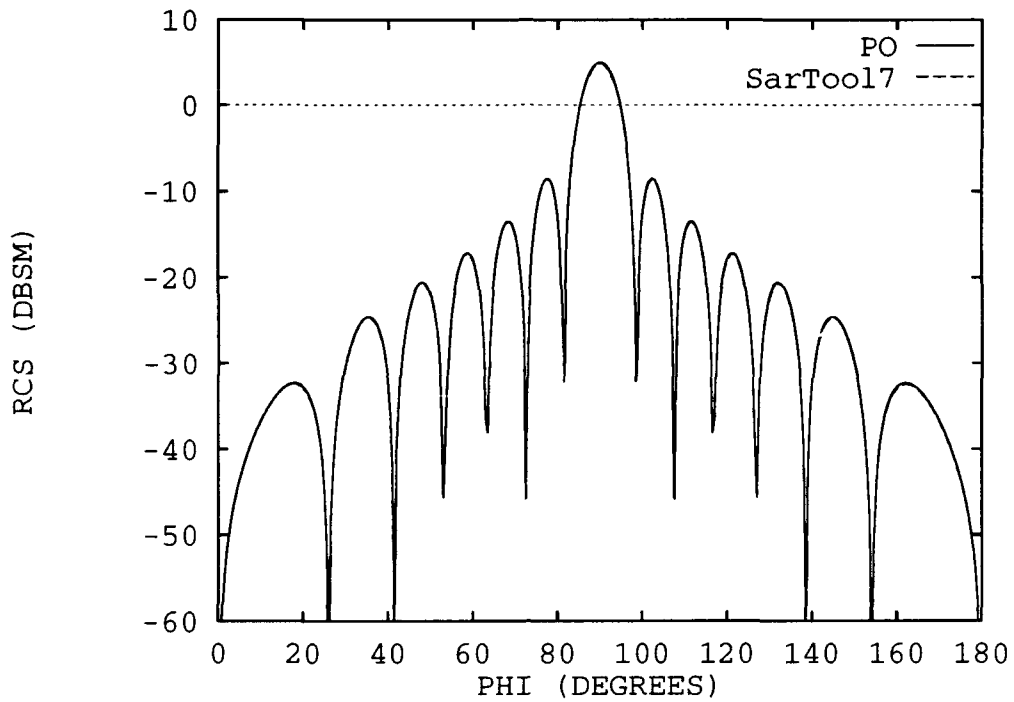


Figure 4.34. Plate in XZ Plane - SarTool RCS Prediction, X=10 cm, Z=15 cm, F=10 GHz, Elev=.1° (28:4-3)

## RECTANGULAR FLAT PLATE

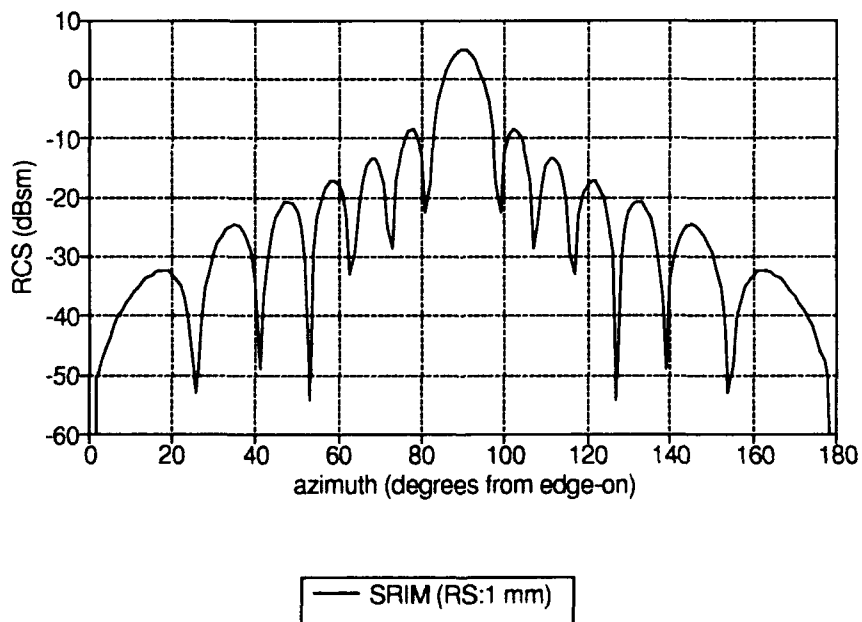


Figure 4.35. Plate in XZ Plane - SRIM RCS Prediction, X=10 cm, Z=15 cm, F=10 GHz, Elev=0°

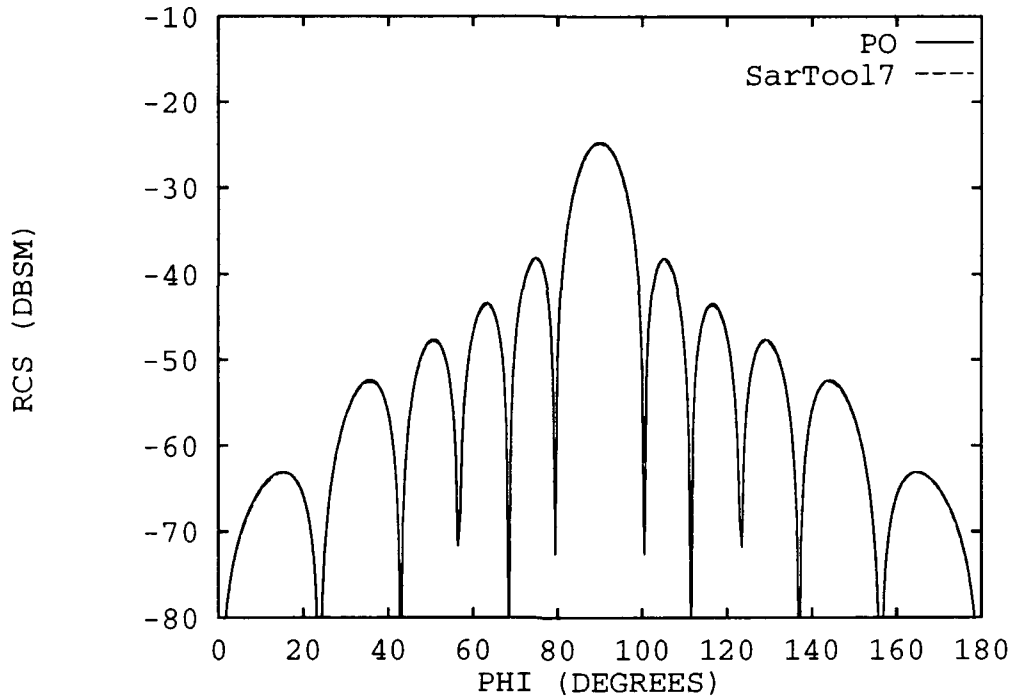


Figure 4.36. Plate in XZ Plane - SarTool RCS Prediction, X=10 cm, Z=15 cm, F=10 GHz, Elev=35° (28:4-4)

### RECTANGULAR FLAT PLATE

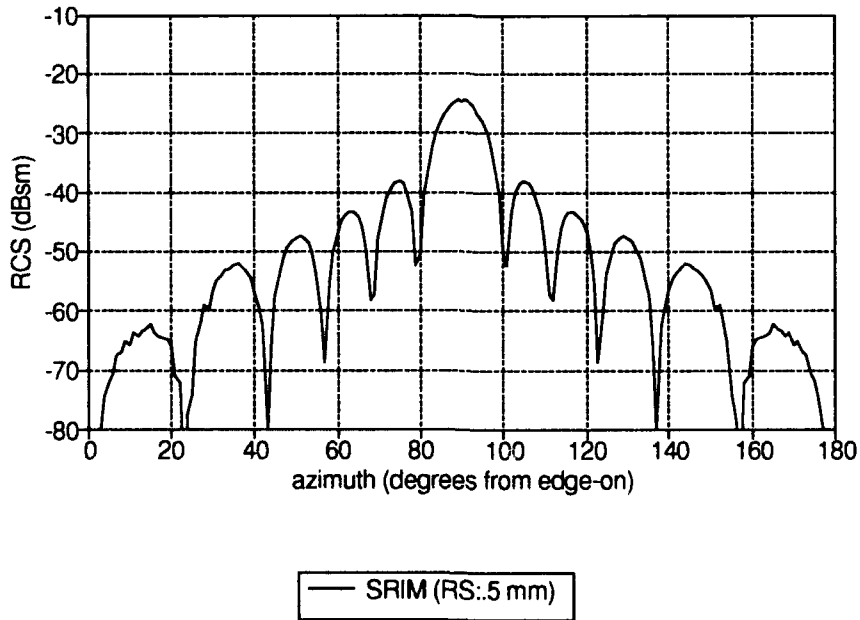


Figure 4.37. Plate in XZ Plane - SRIM RCS Prediction, X=10 cm, Z=15 cm, F=10 GHz, Elev=35°



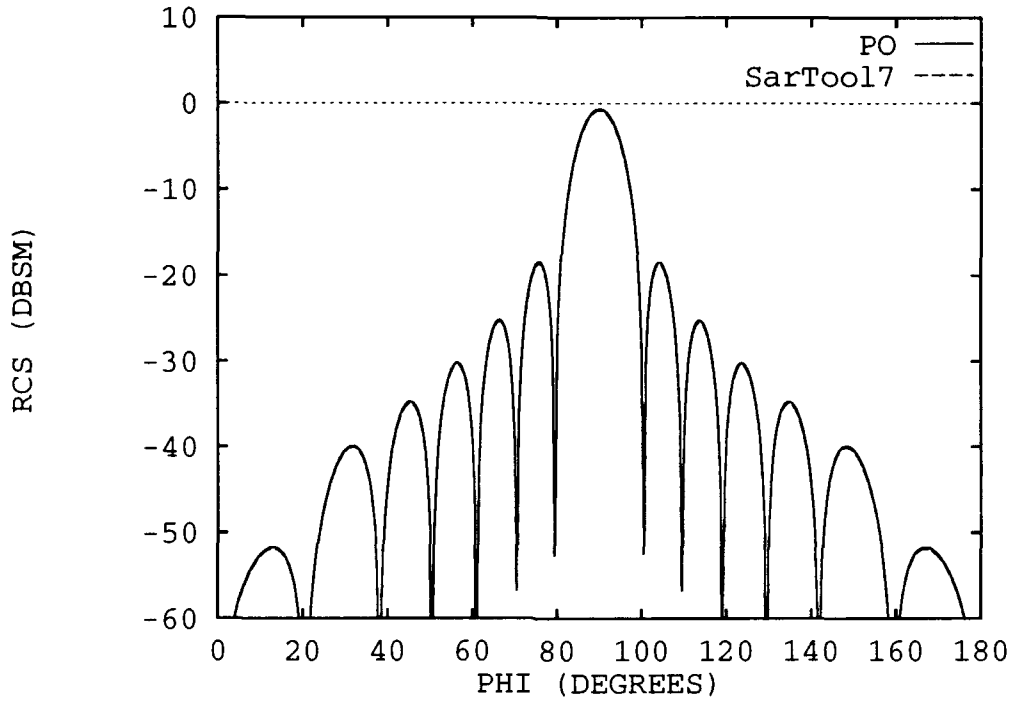


Figure 4.38. Disk in XZ Plane - Sar Tool RCS Prediction, Radius=5 cm, F=10 GHz, Elev=.1° (27:4-7)

### CIRCULAR PLATE

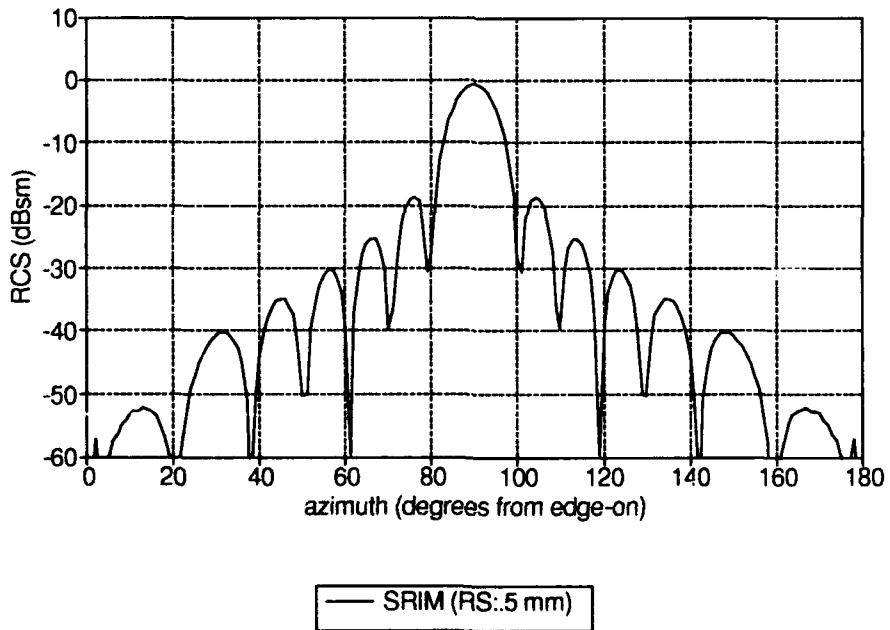


Figure 4.39. Disk in XZ Plane - SRIM RCS Prediction, Radius=5 cm, F=10 GHz, Elev=0°

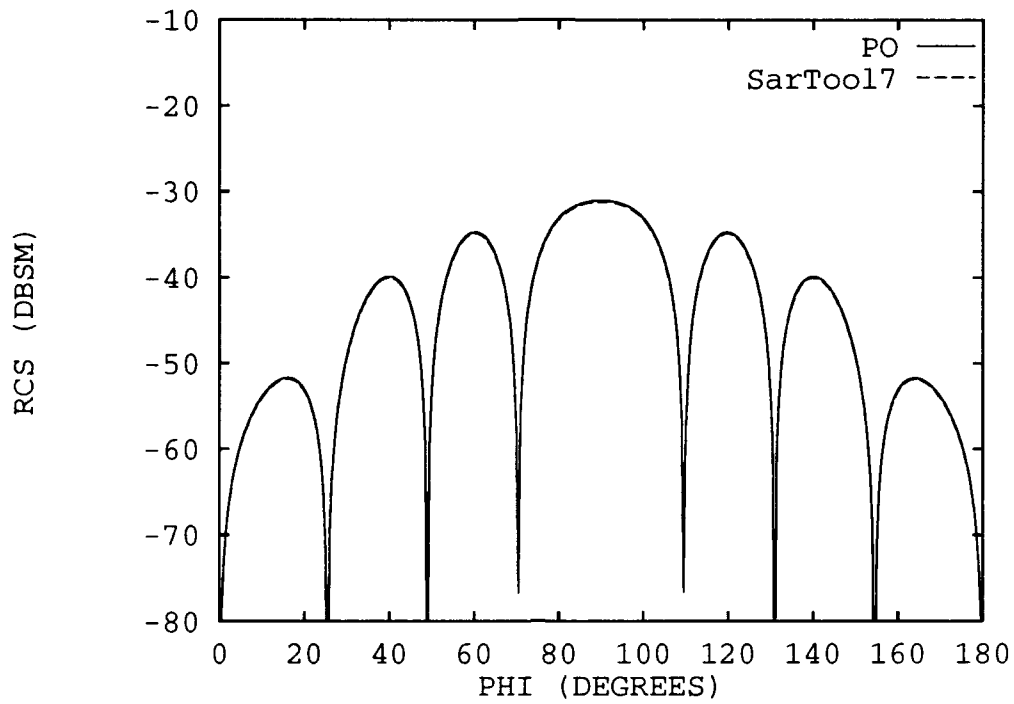


Figure 4.40. Disk in XZ Plane - SarTool RCS Prediction, Radius=5 cm, F=10 GHz, Elev=35° (28:4-8)

### CIRCULAR PLATE

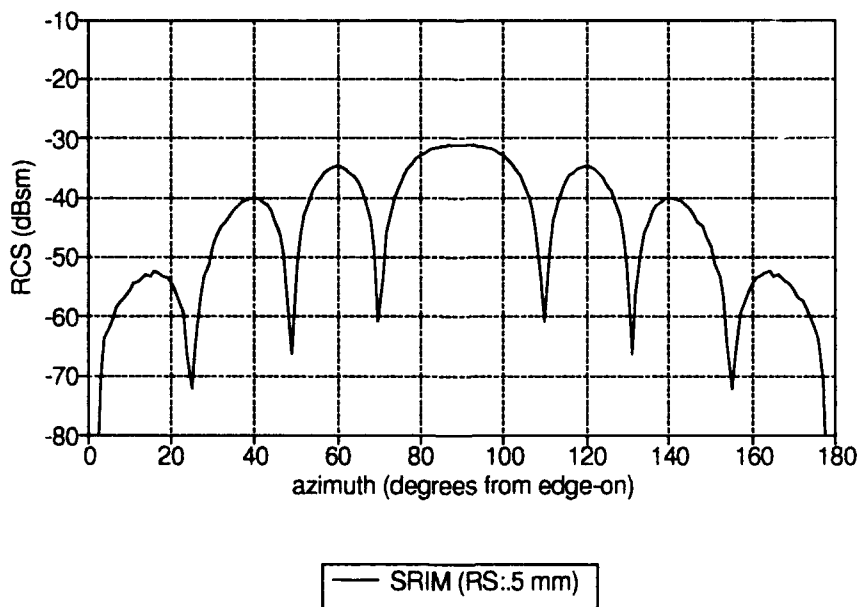


Figure 4.41. Disk in XZ Plane - SRIM RCS Prediction, Radius=5 cm, F=10 GHz, Elev=35°

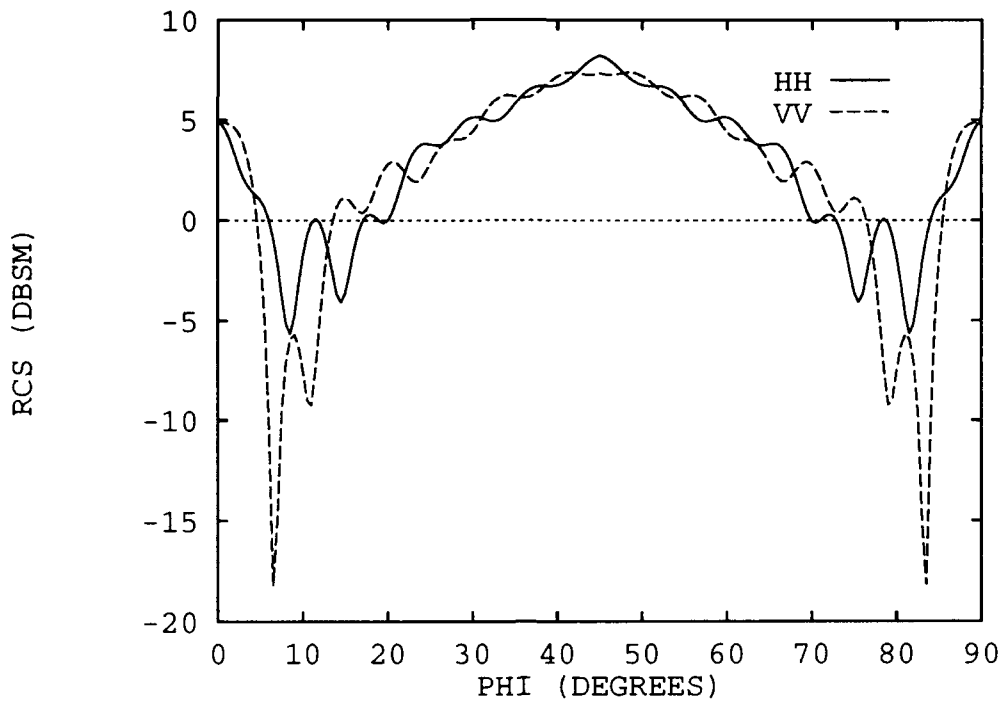


Figure 4.42. Right Dihedral - SarTool RCS Prediction, X=Y=10 cm, Z=15 cm, F=10 GHz, Elev=.1° (28:4-12)

### RIGHT DIHEDRAL

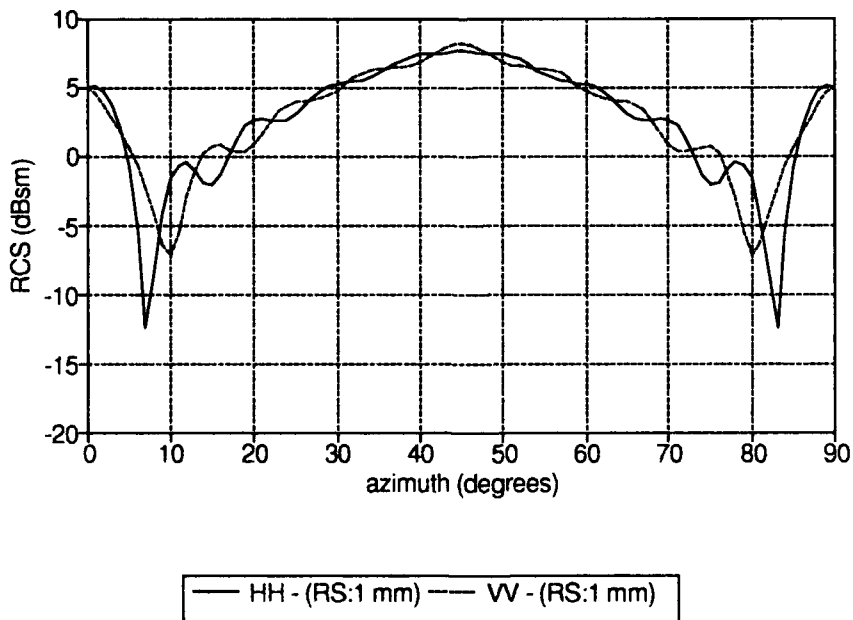


Figure 4.43. Right Dihedral - SRIM RCS Prediction, X=Y=10 cm, Z=15 cm, F=10 GHz, Elev=0°

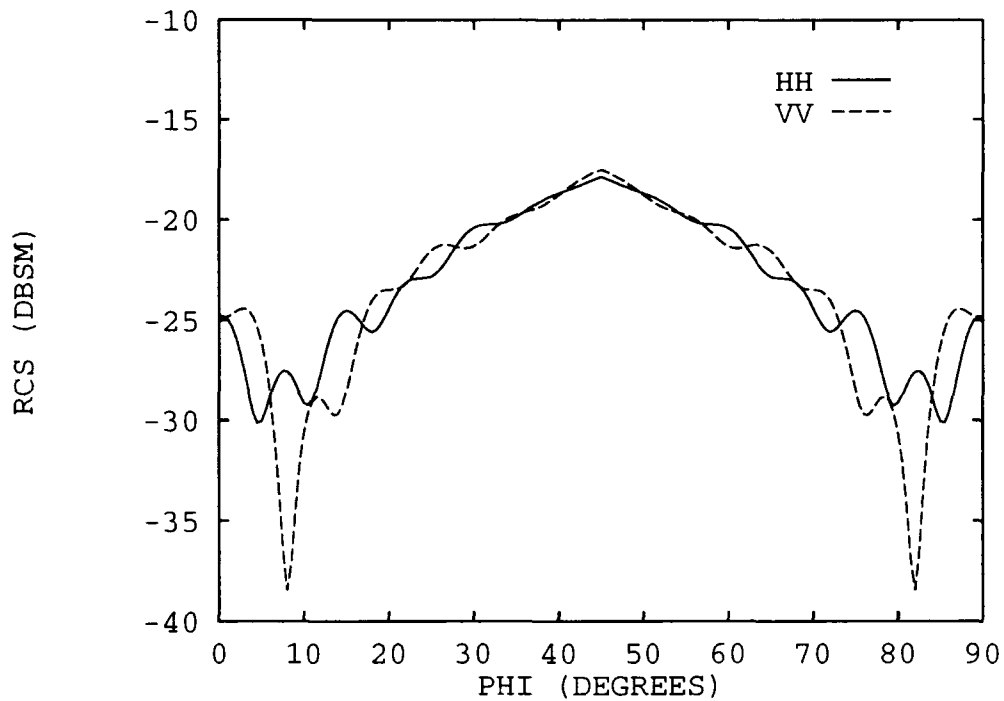


Figure 4.44. Right Dihedral - SarTool RCS Prediction, X=Y=10 cm, Z=15 cm, F=10 GHz, Elev=35° (28:4-15)

### RIGHT DIHEDRAL

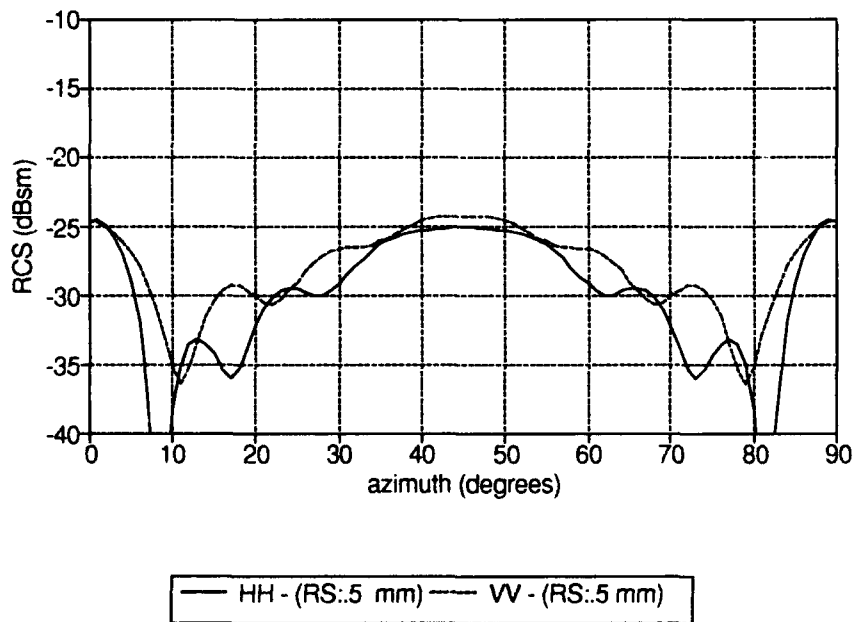


Figure 4.45. Right Dihedral - SRIM RCS Prediction, X=Y=10 cm, Z=15 cm, F=10 GHz, Elev=35°

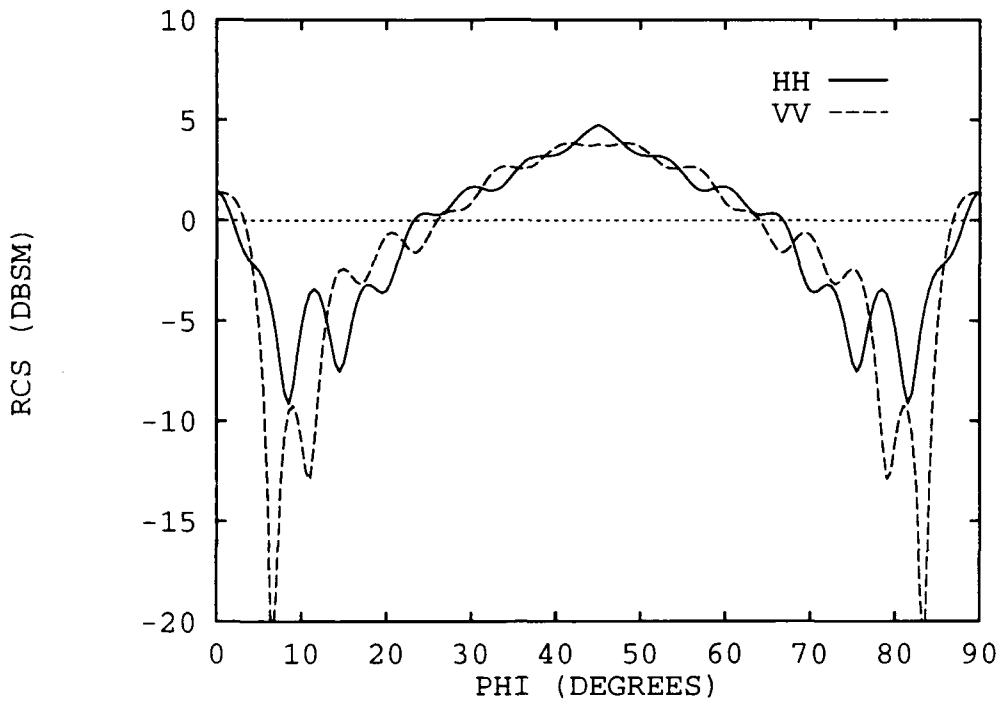


Figure 4.46. Square Trihedral - SarTool RCS Prediction, X=Y=10 cm, F=10 GHz, Elev=.1° (28:4-18)

### SQUARE TRIHEDRAL

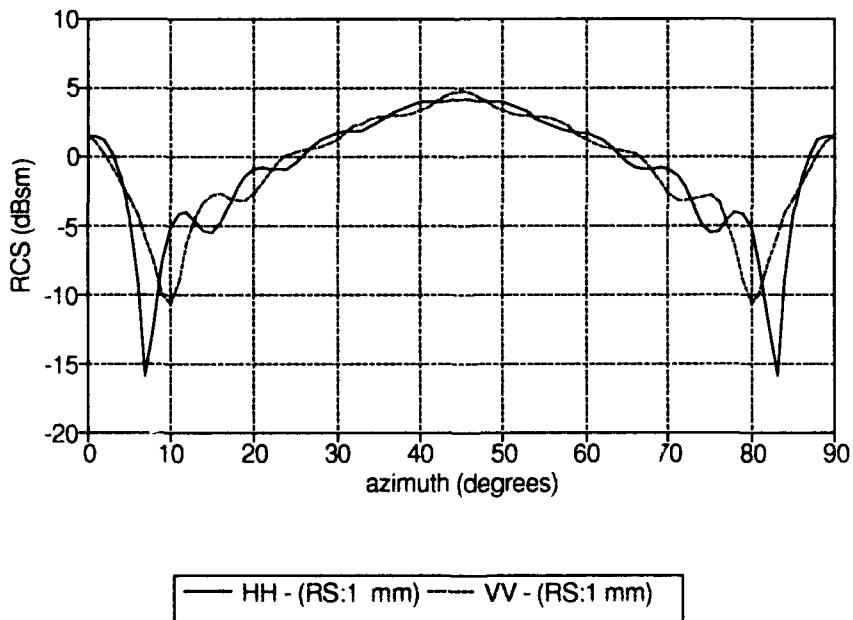


Figure 4.47. Square Trihedral - SRIM RCS Prediction, X=Y=Z=10 cm, F=10 GHz, Elev=0°

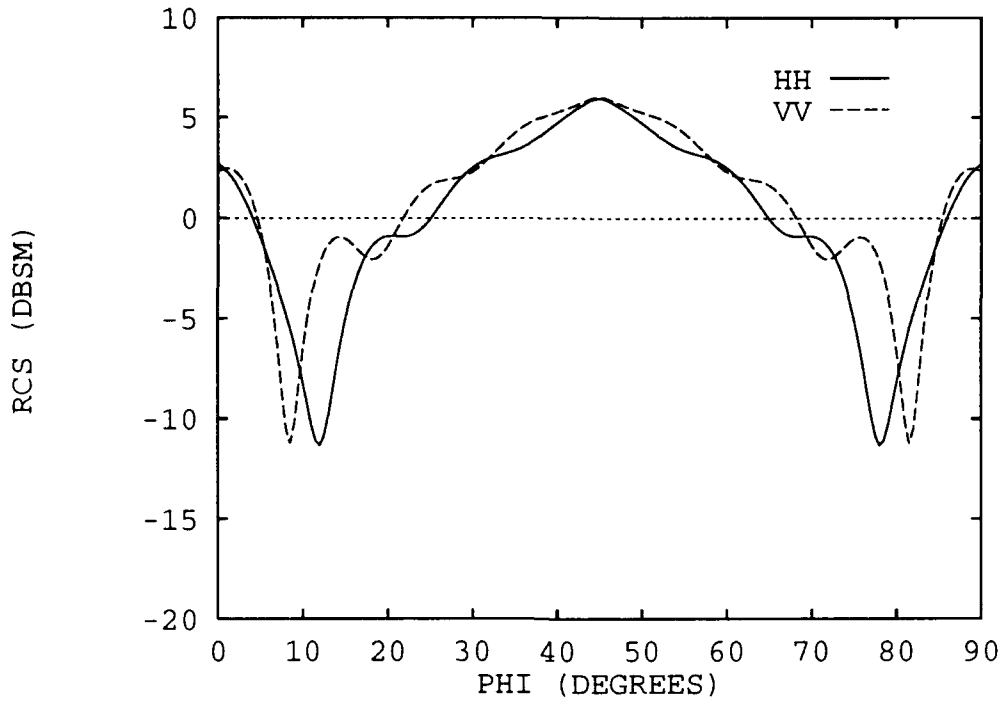


Figure 4.48. Square Trihedral - SarTool RCS Prediction, X=Y=Z=10 cm, F=10 GHz, Elev=35° (28:4-21)

## SQUARE TRIHEDRAL

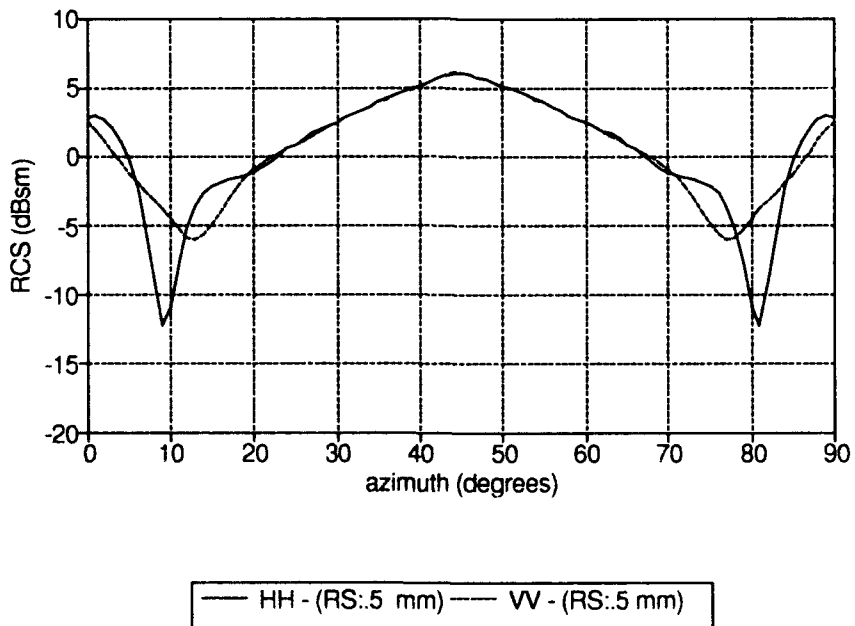


Figure 4.49. Square Trihedral - SRIM RCS Prediction, X=Y=Z=10 cm, F=10 GHz, Elev=35°

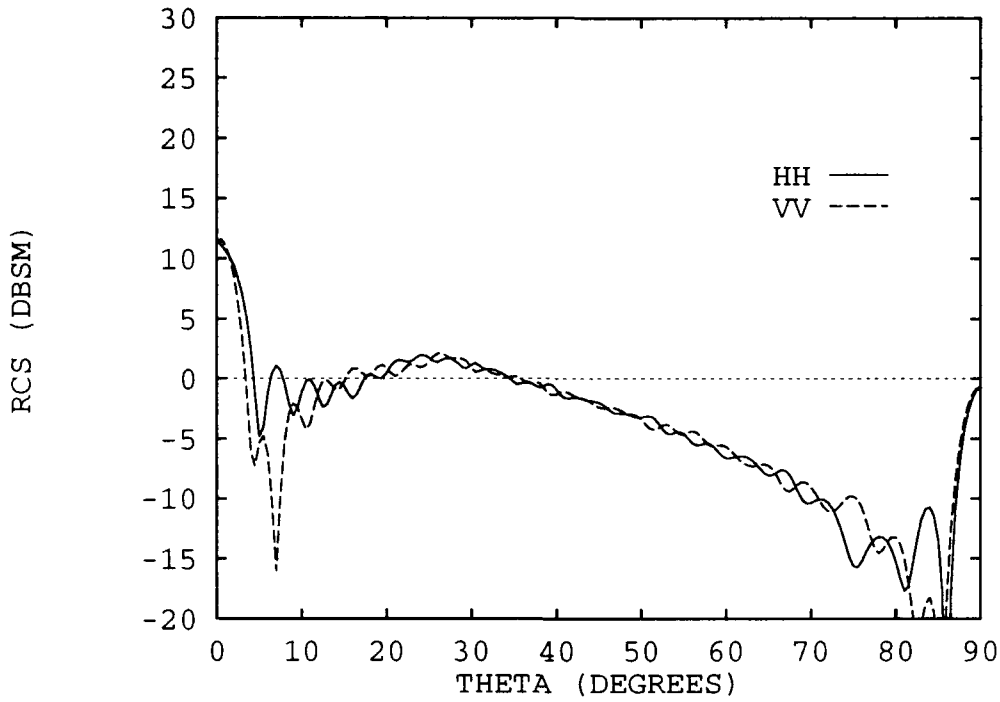


Figure 4.50. Top Hat - SarTool RCS Prediction, Disk Radius=20.32 cm, Cyl Radius=10.16 cm, Cyl Length=20.32 cm, F=10 GHz, Elev=.1° (28:4-26)

## TOP HAT

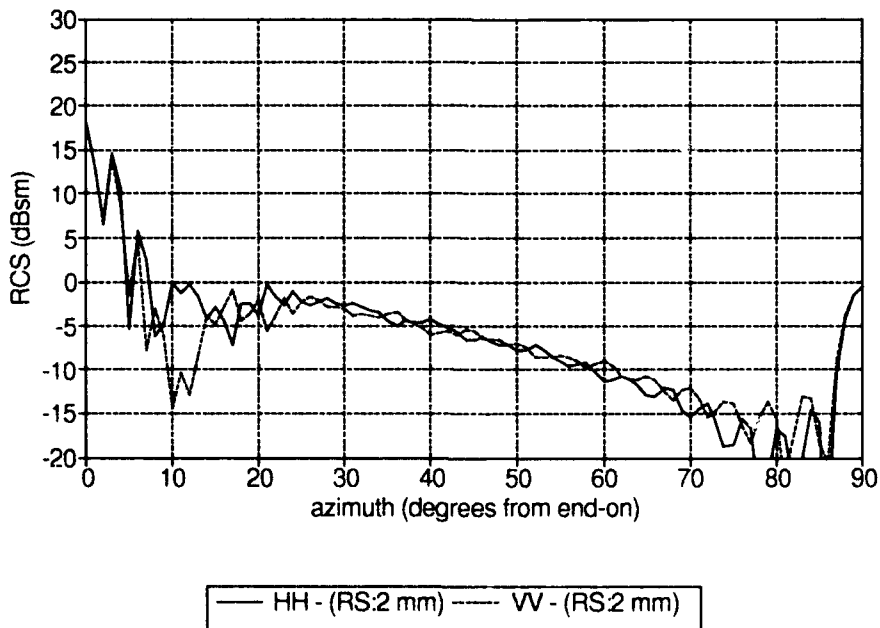


Figure 4.51. Top Hat - SRIM RCS Prediction, Disk Radius=20.32 cm, Cyl Radius=10.16 cm, Cyl Length=20.32 cm, F=10 Ghz, Elev=0°

## ***Validation***

The technique selected to validate SRIM was comparison of its predictions with measured data. Since PO theory does not take into account surface waves (traveling or creeping) and incorrectly accounts for edge diffractions, SRIM is only expected to match predictions in the specular region of the target. This section compares SRIM predictions and measured data for a square plate, circular disk, right circular cylinder, top hat, cone, prolate-spheroid, and generic missile. The geometry and orientation of the targets are depicted in Figure 4.52 and Table 4.2.

***Data Acquisition.*** The top hat data is from Ohio State University (11:858); the other data was collected in the AFIT anechoic test chamber. The AFIT data had to be retrieved from a Hewlett Packard (HP) 9000 Series 236 computer system. For future reference, Appendix F.1 lists the program written to transfer the files from the HP system to an IBM compatible computer. The data also had to be shifted in azimuth angle to match the target orientation used in the SRIM runs. For this purpose, a FORTRAN program, shiftpk.f, was written that searches for the maximum return and correlates it with a user specified azimuth location. The shiftpk.f source code is listed in Appendix F.2.



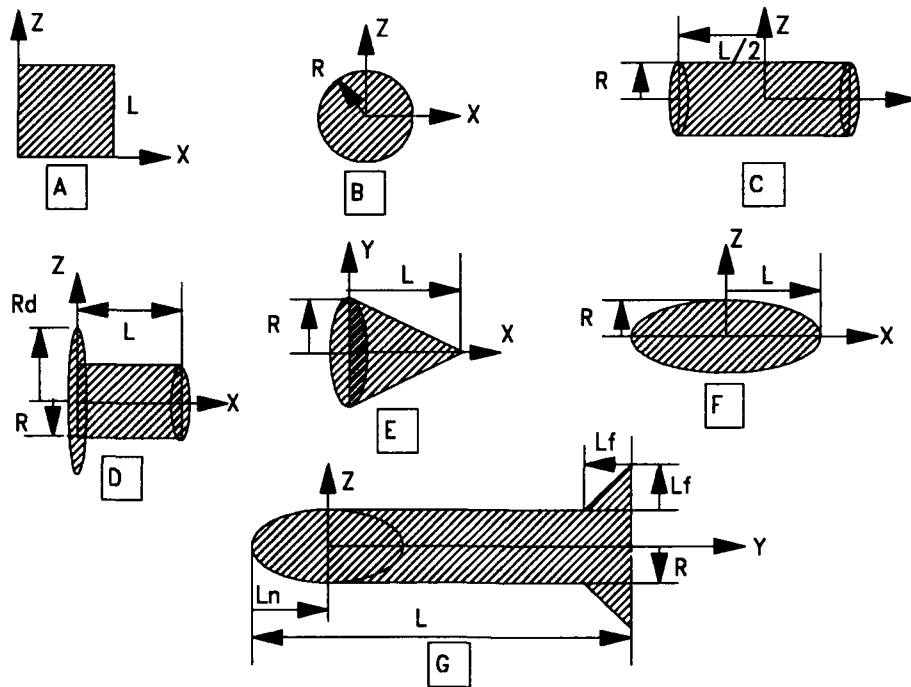


Figure 4.52. Measured Data Target Orientation - A) Square Plate; B) Circular Disk; C) Solid Cylinder; D) Top Hat; E) Cone; F) Prolate Spheroid; G) Generic Missile (2 fins in YZ Plane and 2 fins in XY Plane)

Table 4.2. Measured Data Target Dimensions

TARGET S	DIMENSIONS (cm)				
	R	Rd	L	Ln	Lf
Square Plate			15.24		
Circular Disk	15.24				
Solid Cylinder	3.8		31.27		
Top Hat	10.16	20.32	20.32		
Cone	3.8		8.5		
Prolate Spheroid	3.85		9		
Generic Missile	3.81		54.6	8.9	2.9

**Comparison of Measured Data and SRIM RCS Predictions.** The following comparison data must be caveated with discussions of the frequency regime in which PO is applicable and measured data uncertainties. PO approximations are expected to work well at high frequencies in the specular region of the target. Strictly speaking, the high frequency region starts when  $L/\lambda \gg 1$ . As a rule of thumb,  $L/\lambda > 10$  is usually considered the region in which PO can be applied with confidence. Table 4.3 shows the  $L/\lambda$  ratios of the measured data used in this study. As shown, most of the targets were not in the "high frequency" regime. This fact must be taken into consideration when comparing measured data and SRIM predictions. Target availability and AFIT anechoic chamber quiet zone limitations precluded the collection of higher frequency data.

Table 4.3. Target  $L/\lambda$  Ratios

TARGET	$\lambda$ (cm)	LENGTH (cm)	$L/\lambda$
Rectangular Plate	2.5	15.24	6.1
Circular Disk	2.5	30.48	12.2
Solid Cylinder	3.75	31.27	8.3
Top Hat	3	20.32	6.8
Cone	1.67	9.3	5.6
Prolate Spheroid	3	18.0	6
Generic Missile	3.53	54.6	15.5

The AFIT data was collected by rotating the target in one degree increments. This implies there is a .5° uncertainty as to the azimuth location of the target. Thus, .5° azimuth location differences between measured data and SRIM predictions were expected. Another possible uncertainty which cannot be quantified deals with the orientation of the target with respect to the ground plane. In the SRIM runs, the targets were perfectly aligned so the target normals were boresighted on the antenna. This was also the goal when positioning the targets for measurements, but slight misalignments are possible and likely.

Figure 4.53 shows the results for the square flat plate with thickness .159 cm (1/16"). For realism, the edges of the primitive (BOX) in the SRIM geometry file were not perfectly absorbing as they were when compared to PO equations. SRIM sampled the target with rays 1.52 mm apart (1% of target length). The right ordinate quantifies the differences between the measured data and predictions. The null depths do not match, but the magnitude of the peaks agree remarkably well (less than 2 dBsm difference) out to 45° from the target normal. The largest difference occurs at 22° from edge-on because this is where the effect of the traveling wave is seen in the measured data as predicted by (17:149)

$$\theta = 49.35 \sqrt{\frac{\lambda}{length}} \quad (4.1)$$

Figure 4.54 shows the comparison for the circular disk with thickness .159 cm. Again, the edges of the primitive (RCC) in the SRIM geometry file were not perfectly absorbing. The target was sampled with a ray spacing of 3 mm (1% of diameter). As in the case of the square plate, the depth of the nulls do not match,

## SQUARE FLAT PLATE

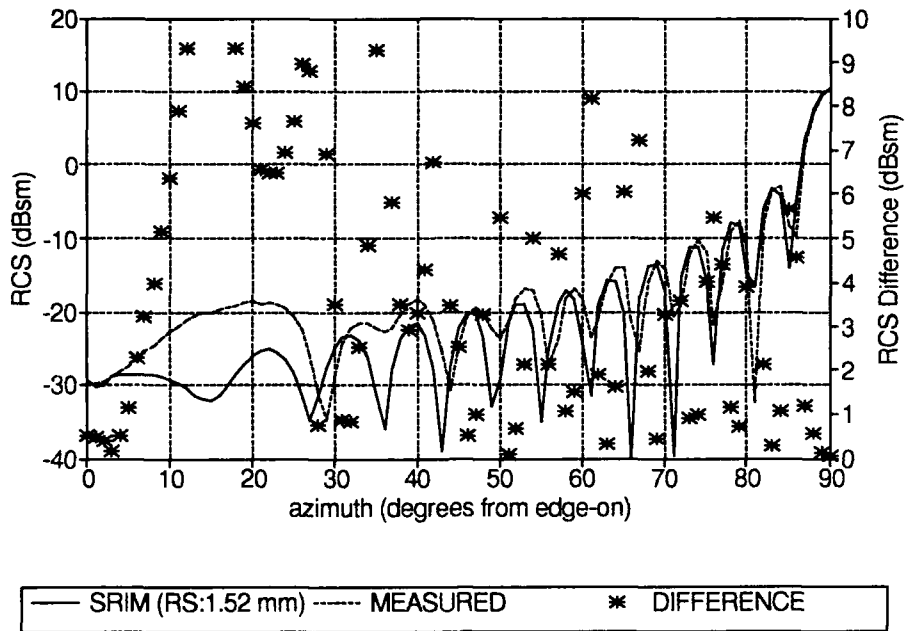


Figure 4.53. Square Plate in XZ Plane - SRIM RCS Prediction & Measured Data, X=Z=15.24 cm, Y=.159 cm, F=12 GHz, P=HH, Elev=0°

## CIRCULAR PLATE

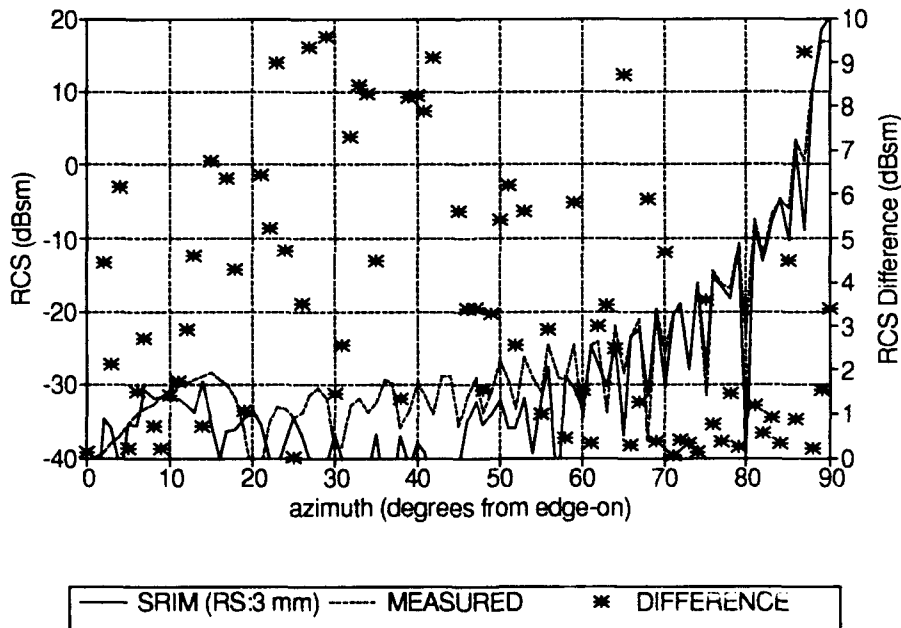


Figure 4.54. Circular Disk in XZ Plane - SRIM RCS Prediction & Measured Data, Radius=15.24 cm, Y=.159 cm, F=12 GHz, P=HH, Elev=0°

but the differences at the peaks are less than 1.5 dBsm out to 45° from the target normal. There is a larger difference at the specular point (90°) than expected. This indicates that the target may have been leaning slightly. Between 0° and 45° SRIM does not accurately predict the lobe positions. In this region, the return is dominated by edge diffractions and traveling waves, neither of which is modeled by Physical Optics. The lobing structure is more complicated for the disk than for the square plate because of the phase variations caused by the disk curvature.

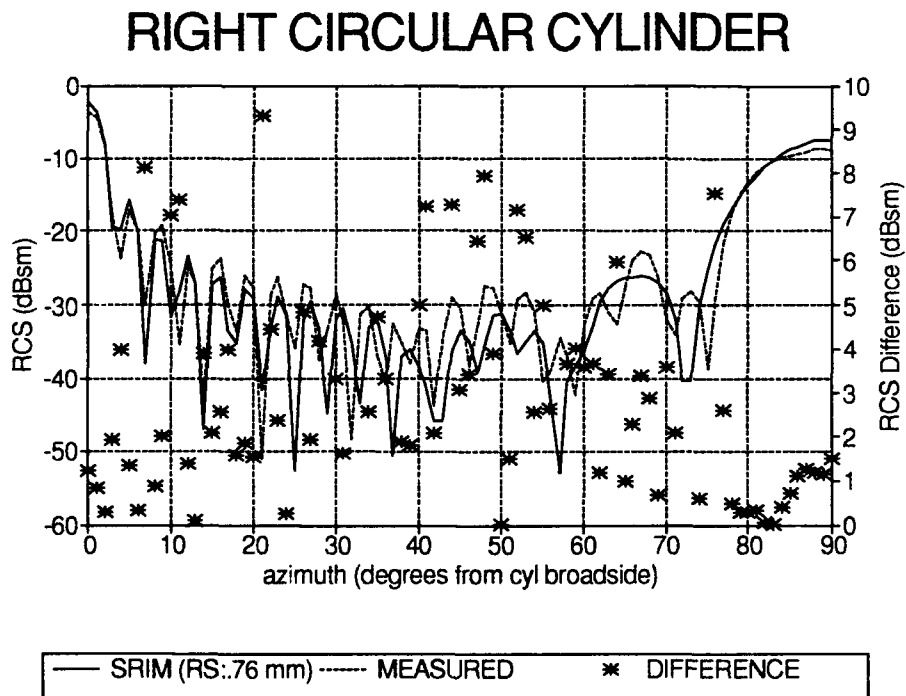


Figure 4.55. Circular Cylinder Along Y Axis - SRIM RCS Prediction & Measured Data, Radius=3.8 cm, Length=31.27 cm, F=8 GHz, P=HH, Elev=0°

Figure 4.55 shows the comparison for the solid right circular cylinder. SRIM sampled the target with .76 mm (1% of cylinder diameter) ray spacing. Broadside to the cylinder is at 0°. Generally, there is good agreement between the predictions and measured data at the peak points between 0° and 30° (less than a

3 dBsm difference). Between  $30^\circ$  and  $60^\circ$  the lobe positions are in slight disagreement. This is probably due to diffractions off both ends of the cylinder. Between  $60^\circ$  and  $90^\circ$  there is fairly good agreement. The difference at  $73^\circ$  is due to the traveling wave as predicted by Eq (4.1).

Figures 4.56 and 4.57 show the comparisons between the Ohio State University top hat experimental results and SRIM predictions. The thickness of the circular disk used in the experiment was not given in the report (11:858); therefore, the edge of the disk in the SRIM geometry file was made perfectly absorbing. The target was sampled with 2 mm ray spacing (1% of cylinder diameter). The predictions and experimental results are in fairly good agreement at the end points. However, the predictions are approximately 4 dBsm lower than the measured data between  $10^\circ$  and  $80^\circ$ . The reason for the disparity is unknown.

Figure 4.58 shows the comparison for a cone that has a  $65.9^\circ$  angle between the base and sidewall (i.e.  $24.1^\circ$  cone half angle). The primitive used for modeling the cone was an elliptical hyperboloid (EHY). SRIM sampled the target with .38 mm ray spacing (.5% of base diameter). The predictions follow the same trend as the measured data, but are only in good agreement when normal to the cone base. Otherwise, the response is dominated by secondary effects such as tip and corner diffractions, and traveling and creeping waves. SRIM did accurately predict the location, but not the magnitude, of the specular flash off the side of the cone.

Figure 4.59 shows the comparison for a prolate spheroid. The primitive used for modeling was an ellipsoid of revolution (ELL). It was sampled with a ray spacing of .77 mm (1% of minor axis length). The predictions and data are in good agreement  $40^\circ$  either side of broadside.

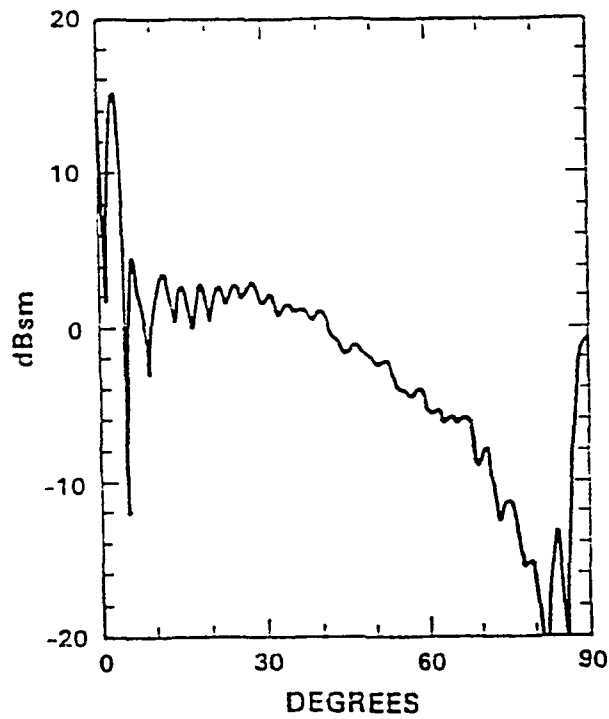


Figure 4.56. Top Hat Ohio State University RCS Measurements, Disk Radius=20.32 cm, Cyl Radius=10.16 cm, Cyl Length=20.32 cm, F=10 GHz, P=HII, Elev=0° (11:858)

### TOP HAT

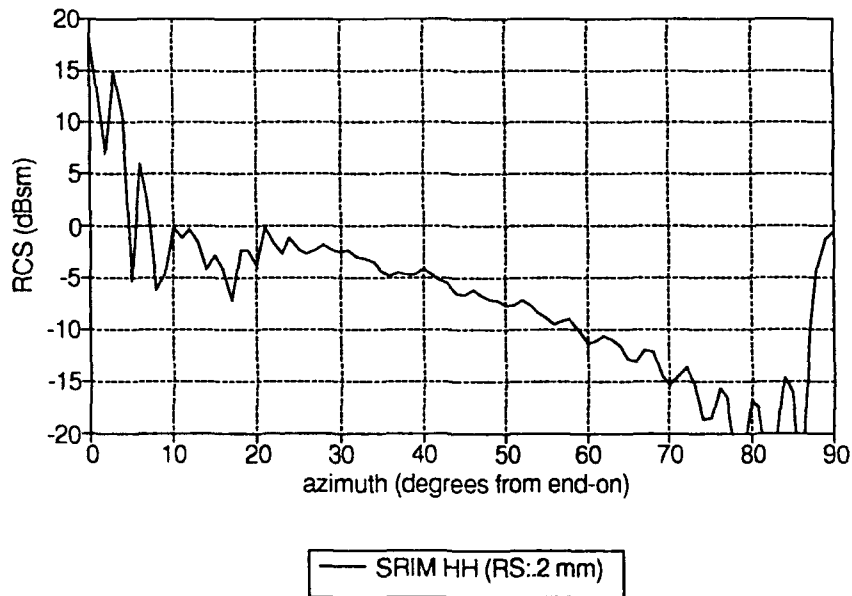


Figure 4.57. Top Hat - SRIM RCS Prediction, Disk Radius=20.32 cm, Cyl Radius=10.16 cm, Cyl Length=20.32 cm, F=10 GHz, P=HH, Elev=0°

# CONE

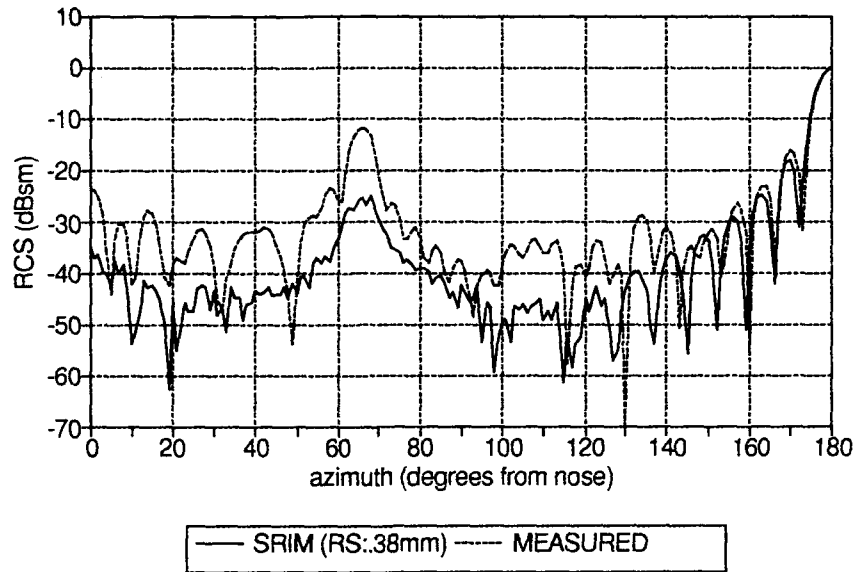


Figure 4.58. Cone - SRIM RCS Prediction & Measured Data, Base Radius=3.8 cm, Axial Length=8.5 cm, F=18 GHz, P=HH, Elev=0°

# PROLATE SPHEROID

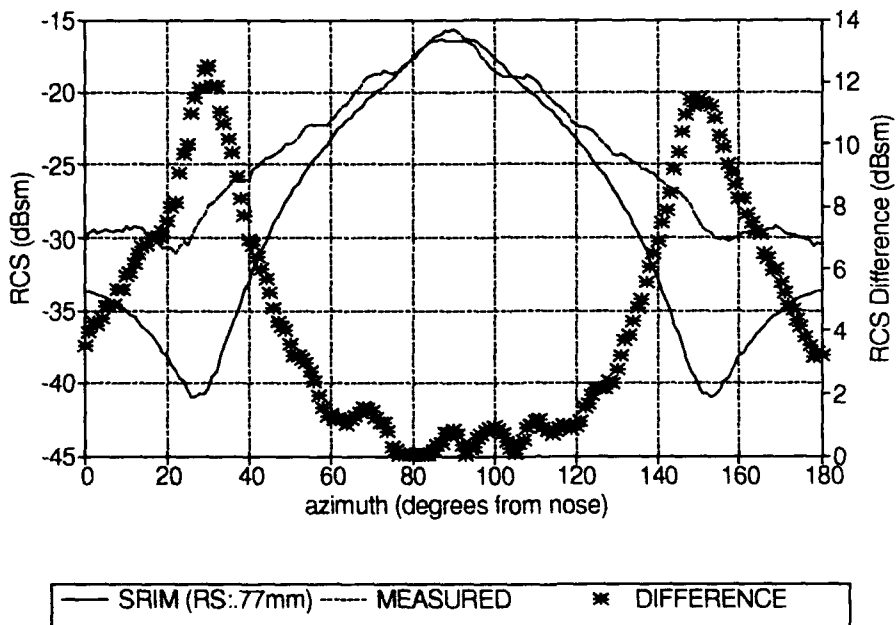


Figure 4.59. Prolate Spheroid - SRIM RCS Prediction & Measured Data, Major Axis Length=18 cm, Minor Axis Length=7.7 cm, F=10 GHz, P=HH, Elev=0°



A generic missile was used for the final comparisons. The primitives used to construct the geometry file were two right circular cylinders (RCC), an ellipsoid of revolution (ELL), and four arbitrary wedges (ARW). The target was sampled with .76 mm (1% of cylinder diameter) ray spacing. Figure 4.60 shows the RCS comparisons. As shown in the figure, the SRIM predictions follow the same trend as the measured data, but there are significant differences in the lobing structures. However, at all the major return locations there is good agreement.

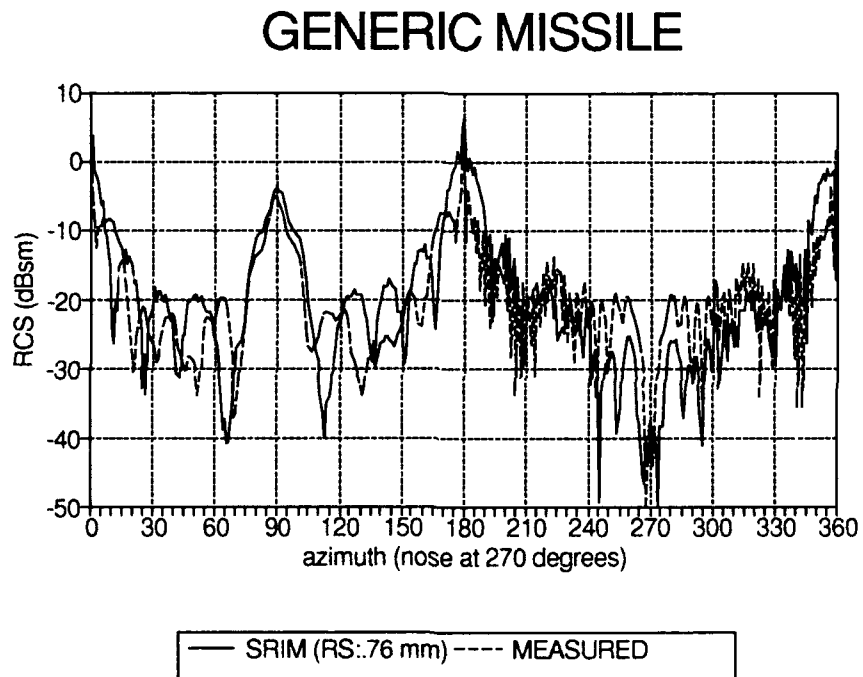


Figure 4.60. Generic Missile - SRIM RCS Prediction & Measured Data, Length=53.3 cm, Radius=3.81 cm, F=8.5 GHz, P=HH, Elev=0°

## ***Summary***

Table 4.4 lists the comparisons presented in this chapter. The data shows that SRIM RCS predictions are in good agreement with closed-form Physical Optics RCS equations for a sphere, rectangular plate, circular plate, cylinder, dihedral, and trihedral. There is a 5 dBsm disagreement in the comparisons for the top hat. SRIM predictions for the flat plates, cylinder, dihedral, trihedral, and top hat were also compared with other Physical Optics-based RCS prediction software codes. Except for the top hat, these comparisons show that SRIM performs as well, if not better, than the other codes.

Measured data comparisons show that SRIM is a good predictor of RCS values if restricted to the specular region of the target, or conversely, if the target has many specular regions. However, the accuracy is better for flat surfaces than for curved surfaces.

Table 4.4. Comparisons Used to Evaluate SRIM

Targets	Closed-Form PO Equations	Other Codes		Measured RCS Data
		SarTool	DBSM	
Sphere	X			
Flat Plate	X	X	X	X
Circular Plate	X	X	X	X
Solid Cylinder	X		X	X
Right Dihedral	X	X		
Square Trihedral	X	X		
Top Hat	X	X		X
Cone				X
Prolate Spheroid				X
Generic Missile				X

## ***V. Conclusions***

### ***Introduction***

The objective of this effort was to verify and validate the SRIM Version 4.0 radar cross section predictions for perfectly-conducting (electrically), smooth, canonical shapes. Shapes evaluated were a sphere, rectangular plate, circular disk, right circular cylinder, dihedral, trihedral, top hat, cone, prolate spheroid, and generic missile. The verification methodology consisted of comparing SRIM RCS predictions to Physical Optic equation predictions and the RCS prediction codes, SarTool and DBSM. Measured RCS data was used for validating the SRIM predictions.

Even though SRIM uses the physical optics surface current approximation, it relies on ray tracing to sample the *target geometry*. The user decides the sampling density. It was quite apparent from this effort that the SRIM predictions are extremely dependent upon the sampling density. The results show that as the distance between rays decreases (i.e. ray density increases), the better the SRIM predictions. Due to this dependency on ray spacing, the SRIM predictions must be stated as a function of ray spacing. In this study, the distance between rays was selected as a percentage of the shortest length of the target that is perpendicular to the radar line of sight when the target is normal to the radar. For example, a 5 cm plate (broadside to the radar) with 1% ray spacing was sampled every .5 mm (5 cm x .01). Quantification of the differences between SRIM predictions and the

other data were based on SRIM runs with 1% and .5% ray spacing for elevation angles of 0° and 35°, respectively.

### ***Verification Conclusions***

The differences between the SRIM predictions and other sources of data were virtually non-existent at the specular point of return for the plates and cylinder. However, there were major differences in the depth of the nulls. SRIM predictions for a sphere fluctuate around a level that is within .6 dBsm of the PO predictions. The fluctuations are not a concern because they are small and decrease even further as the ray density is increased. The SRIM predictions also match the Physical Optics equation predictions for the dihedral at 0° elevation (less than .25 dBsm difference even at the nulls). At 35° elevation, the differences are less than 1 dBsm at the peaks, but the depths and location of the nulls do not agree. Except for the nulls, all the trihedral predictions are within 1 dBsm of agreement. There is a 4-7 dBsm disagreement between the SRIM and PO equation predictions for the top hat. The cause of the disagreement was not determined

With the exception of the top hat results, this comparison study shows that SRIM correctly implements PO theory. Furthermore, since the predictions are accurate for dihedrals and trihedrals, the code correctly accounts for double and triple bounces between flat plates. The error in the top hat predictions indicate there may be a problem with SRIM's treatment of multiple bounces between a curved object and flat plate. The top hat target needs to be analyzed in greater depth before the SRIM RCS predictions can be given a blanket "verification completed".

### ***Validation Conclusions***

SRIM predictions for a square plate, circular plate, cylinder, cone, prolate spheroid, and generic missile were compared with measured RCS data acquired in the AFIT anechoic chamber. The SRIM predictions agreed remarkably well with the measured data, especially near the specular region of the target. For the flat plate, the disagreement was less than 1 dBsm out to 45° from the target normal. The circular disk comparisons also show good agreement (less than 1.5 dBsm disagreement) out to 45°. Between 45° and grazing incidence, the lobing patterns do not agree. The circular cylinder comparisons match fairly well (less than 2 dBsm disagreement) when broadside to the cylinder or end cap. In the off-specular region there are differences in the lobing structures. These differences are expected because the diffractions and traveling waves inherent in the cylinder response are not modeled in SRIM. The top hat predictions were compared with experimental data from Ohio State University. There is agreement only when broadside to the cylinder and disks, otherwise, there is a 4 dBsm difference. SRIM predictions for a cone agreed with measured data only in the specular region of the base. The difference between the SRIM predictions and measured data for a prolate spheroid was less than 2 dBsm 40° either side of broadside. There is remarkably good agreement between the SRIM predictions and measured data for the generic missile at all aspect angles (only analyzed 0° elevation plane).

This validation study shows that SRIM is a good predictor of RCS when restricted to the specular region of the target. Furthermore, as evidenced by the generic missile results, the restriction to the specular region can be relaxed as the number of specular regions in the target increase.

### ***Recommendations for Future Work***

To complete the RCS prediction verification and validation of SRIM, future work should expand upon this study by assessing the code's performance against complex (i.e. realistic) targets. Furthermore, the discrepancy between the predictions and measured data for a top hat should be analyzed. Also, the null location differences between SRIM, SarTool, and PO equations for the dihedral and trihedral should be evaluated. In addition, the interface between a graphical editor and SRIM should be developed to ease the construction of the target models and to enable a validation study of the code's imaging capabilities.

## ***Appendix A: GIFT and RADSIM Input File Requirements***

This appendix describes how to generate the input files required to run GIFT and RADSIM. Note: Running the automation technique described in Chapter 3 will create the radar.dat file.

### ***A.1 Geometry File, \*.cg***

The file consists of six records--title, control, primitive definition, region definition, delimiter, and region identification. SRIM reads the file with FORTRAN FORMAT statements, so the data in the file must be of the format shown in Table A.1.1.

The data required for the primitive definition record is listed in Table A.1.2 for the box, right circular cylinder, and sphere. Figure A.1.1 shows the geometry description for each of these primitives. Reference (4:Appendix A) has a complete listing of Tables A.1.1, A.1.2, and Figure A.1.1 for all the primitives SRIM supports. For each primitive, the vertex is positioned with respect to a common origin used for the entire target. The vectors of Figure A.1.1 are defined with respect to the vertex, not the origin.



Table A.1.1. Format for Geometry.cg File

RECORD TYPE	# OF LINES	COLUMNS	CONTENT
Title	1	1-2 6-65	Primitive Units (in,ft, mm, cm, m) Comments
Control	1	1-5 6-10	Number of Primitives Number of Regions
Primitive Definition (See Table A.1.2)	LAW Table A.1.2	1-5 6-8 11-70 71-80	Primitive Number (user's choice) Primitive Name See Table A.1.2 Comments
Region Definition	1 per Region	1-5 7-8 9-13 14-69  71-80	Region Number Boolean Operator Primitive Number Repeat Sequence for Boolean Operator (2 columns) and Primitive Number (5 columns) Comments
Delimiter	1	1-5	-1
Region Identification	1 per Region	1-5 6-10 11-15 31-80	Region Number 501 0 Comments

Table A.1.2. Primitive Definition Description

PRIMITIVE TYPE	PRIMITIVE NAME	COLUMNS					
		11-20	21-30	31-30	41-50	51-60	61-70
Box	BOX	Vx Wx	Vy Wy	Vz Wz	Hx Dx	Hy Dy	Hx Dz
Right Circular Cylinder	RCC	Vx R	Vy	Vz	Hx	Hy	Hx
Sphere	SPH	Vx	Vy	Vz	R		

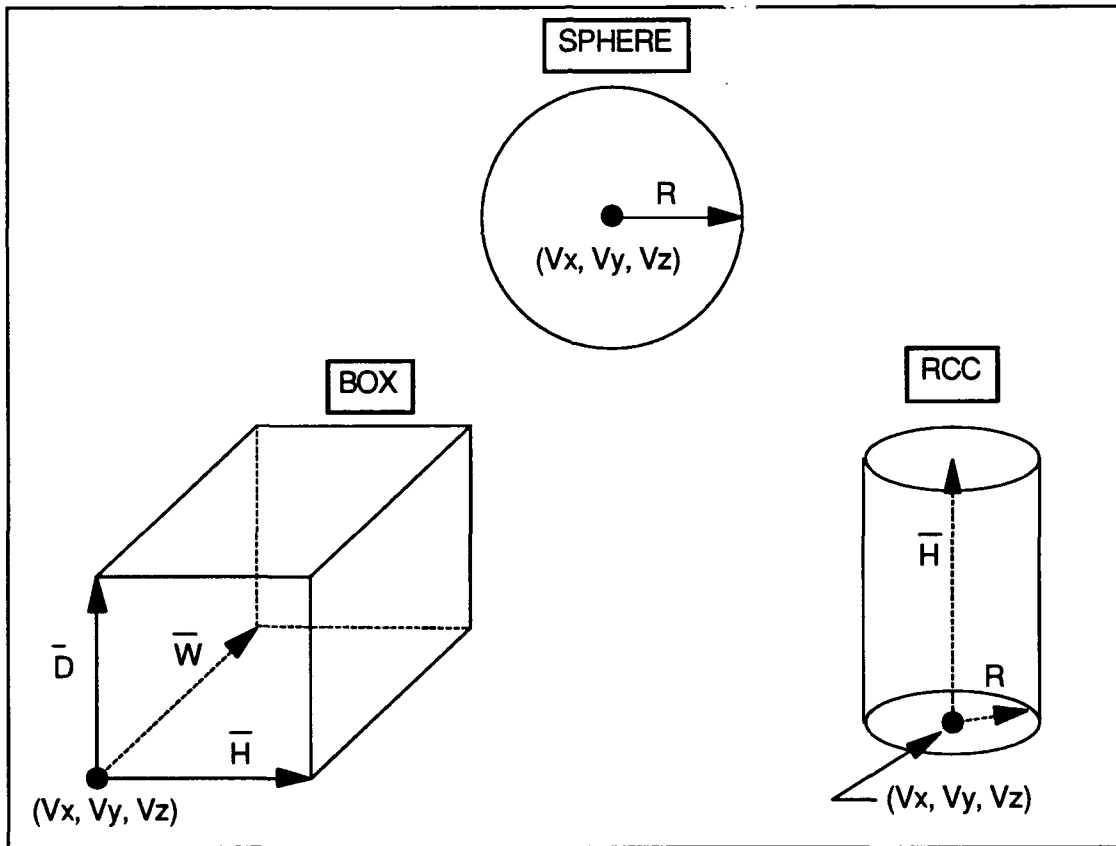


Figure A.1.1. Primitive Geometry

## ***A.2 Scattering Model File, Surface.dat***

A smooth reflective model is defined by assigning a model number to the percentage of absorption desired. Each model assignment record consists of the following three data items:

- Item 1) model number that is an integer value between 101 and 199
- Item 2) fraction of power to be absorbed by the surface (real number  $\leq 1.0$ )
- Item 3) comment ( $\leq 64$  characters)

Each record is free format so the data does not have to be in specific columns, as in the geometry file, but each entry must be separated by a comma, and Item 3 must be enclosed in single quotes. For example, to associate the model number 102 with a perfect reflector (no absorption) the data line would read: 102,0.0,'perfect reflector or any other comment'. In addition, the first line of the file must list the following two data items (separated by a comma):

- Item 1) left most digit (integer) of model number, hence, always a 1 for smooth reflective models
- Item 2) number (integer) of model identifiers contained within the file, or in other words, this number will be one less than the total number of lines contained in the file

## ***A.3 Surface Assignments File, \*.sur***

The file consists of two types of data records. One of the data records is an integer whose value corresponds to the number of explicit surface assignments made within the file (i.e. one less than the total number of lines contained in the file). This data record is the first line of the file.

The other type of data record consists of the following items:

- Item 1) region number (as defined in geometry file) of the primitive whose surface is being assigned a reflection model
- Item 2) primitive number (as defined in geometry file) whose surface is being assigned a reflection model

- Item 3) surface number (as defined in Table A.3.1) of the primitive being assigned a reflection model
- Item 4) scattering model number (as defined in scattering model file) that is being assigned to the primitive surface
- Item 5) optional comment field

Table A.3.1 gives the surface numbers required for Item 3 for the box, right circular cylinder, and sphere. V, D, W, and H (referred to in Table A.3.1) are identified in Figure A.1.1. Reference (4:Table A-3) gives a complete listing of surface numbers for all the primitives SRIM supports. Note, however, that the box surface numbers given in the reference is incorrect. The correct surface numbers given in Table A.3.1 were found by trial and error. As an additional note, zeros can be used as wildcards in Items 1, 2, and 3. For instance, a zero in Item 3 signifies that all surfaces of the primitive specified by Items 1 and 2 are assigned the scattering model listed in Item 4. Every surface not explicitly assigned a model number defaults to a perfect reflector.

Table A.3.1. Surface Identification Numbers

PRIMITIVE	SURFACE NUMBERS					
	1	2	3	4	5	6
BOX	plane with V,D, W	opposite surface 1	plane with V,H,D	opposite surface 3	plane with V,H, W	opposite surface 5
RCC	plane with V	opposite surface 1	cyl face	N/A	N/A	N/A
SPH	face	N/A	N/A	N/A	N/A	N/A

#### **A.4 Radar Data File, Radar.dat**

This is a free format file, so all character entries are in single quotes and each entry on a line is separated by commas. The following describes each data record:

- Line 1) 'WAVELEN', real value of the radar center wavelength in meters
- Line 2) 'TTAU', 90.0 for transmitter vertical polarization or 0.0 for transmitter horizontal polarization
- Line 3) 'TEPSI', 0.0 for transmitter vertical and horizontal polarizations
- Line 4) 'RTAU', same as Line 2 except applies to receiver
- Line 5) 'REPSI', same as Line 3, except applies to receiver
- Line 6) 'NOISE, ' ' name of file containing noise covariance matrix (file not used for RCS generation)
- Line 7) 'DIGITAL RESP.', ' ' name of file containing digitized impulse response data (file not used for RCS generation)
- Line 8) 'IPR PARAM.', 'RANGE', 'CROSS-RANGE', columns headings for Line 9
- Line 9) 'RESOL', real values for range and cross-range 3 dB width of the main lobe of the system impulse response in meters (used typical values of 3.0,3.0)
- Line 10) 'SPACEN', real values for image pixel size in meters (used typical values of 1.5,1.5)
- Line 11) 'NTAYLOR' real value signifying number of terms to use to a Taylor-weighted system response; if a one is used the system is unweighted and a sinc response is used (used 1.0,1.0)
- Line 12) 'SLL', real values of first sidelobe in the system impulse response in decibels (relative to the main lobe); ignored if a one is used in Line 11
- Line 13) 'CONV. CELLS', integer values signifying the number of pixels that will be spanned by the system response (used typical values of 29,29)

Line 14) 'QUANT. FACT', real values signifying the quantization factor  
for the system response (used 10.,10.)

## ***Appendix B: Actual SRIM Input Files Used***

This appendix lists the SRIM input files used during this study. Section B.1 gives typical examples of the gift.inp, radsim.inp, radar.dat, and surface.dat files described in Chapter 3 and Appendix A. Other than surface.dat, all these files are created by running the automation technique described in Chapter 3. The contents of surface.dat was the same for all the runs done in this study. Section B.2 lists the geometry and surface assignment files used for the SRIM data presented in Chapter 4. Table B.2.1 correlates the files with the Chapter 4 figures.

### ***B.1 Input File Examples***

In these examples, the blank lines indicate a carriage return.

#### ***Gift.inp.***

```
cp5.cg
0
radar
RCS
unclassified
25

16.0
0.0
-1
-1
n
1.00
1.00
n
0.00050
0.00050
c
end
```

***Radsim.inp.***

cp5.ray  
y  
cp5.sur  
y

y  
n  
n  
90.0  
200,200  
0.,0.

unclassified

y

***Radar.dat***

'WAVLEN',.02998  
'TTAU',0.  
'TEPSI',0.  
'RTAU',0.  
'REPSI',0.  
'NOISE','  
'DIGITAL RESP.','

'IPR PARAM.', 'RANGE', 'X-RANGE'  
'RESOL',5.0,5.0  
'SPACEN',.05,.05  
'NTAYLOR',1.,1.  
'SLL',-25.,-25.  
'CONV. CELLS',0.,0.  
'QUANT. FACT.',10.,10.

***Surface.dat***

1,2  
101,0.0,'perfect reflect'  
103,1.0,'perfect absorb'



## **B.2 Geometry and Surface Assignment Files**

### **Rectangular Plates**

#### ***rpcen.cg***

```
cm 10 x 15 cm flat plate centered in x-z plane, - 11 Oct
1 1
1 box 5.0 0.0 -7.5 -10.0 0.0 0.0
      0.0 -0.1 0.0 0.0 0.0 15.0
1 1
-1
1 501 0 object
```

#### ***rpcen.sur***

```
6
1, 1, 1, 103
1, 1, 2, 103
1, 1, 3, 101
1, 1, 4, 101
1, 1, 5, 103
1, 1, 6, 103
```

#### ***rpneg.cg***

```
cm flat plate in neg x-z plane,x=-10, y=-.1, z=-15 - 11 Oct
1 1
1 box 0.0 0.0 0.0 -10.0 0.0 0.0
      0.0 0.0 -15.0 0.0 -0.1 0.0
1 1
-1
1 501 0 object
```

#### ***rpneg.sur***

```
6
1, 1, 1, 103
1, 1, 2, 103
1, 1, 3, 103
1, 1, 4, 103
1, 1, 5, 101
1, 1, 6, 101
```

***rpx10z15.cg***

```
cm flat plate in x-z plane,x=10, y=.1, z=15 - 21 Aug
1 1
1 box 0.0 0.0 0.0 10.0 0.0 0.0
      0.0 -0.1 0.0 0.0 0.0 15.0
  1 1
-1
1 501 0          object
```

***rpx10z15.sur***

```
6
1, 1, 1, 103
1, 1, 2, 103
1, 1, 3, 101
1, 1, 4, 101
1, 1, 5, 103
1, 1, 6, 103
```

***rp6.cg***

```
cm 6 in plate centered in xz plane, 1/16 inch thick, - 11 Oct
1 1
1 box 7.62 0.0 -7.62 15.24 0.0 0.0
      0.0 -0.15875 0.0 0.0 0.0 15.24
  1 1
-1
1 501 0          object
```

***Sphere***

***sphere.cg***

```
cm 5 cm radius sphere - built 4 Sep 91
1 1
1 sph 0.00 0.00 0.00 5.00 0.00 0.00
  1 1
-1
1 501 0          sphere (ka=10.47 @ 10 GHz)
```

### **Circular Plates**

#### **cp5.cg**

mm 5 cm disk, 1 mm thick, origin centered, model built 18 Aug 91

```
1 1
1 rcc 0.00 -0.5 0.00 0.00 1.00 0.00
      50.0 0.00 0.00 0.00 0.00 0.00
  1 1
-1
1 501 0          object
```

#### **cp5.sur**

```
3          number of entries
1, 1, 1, 101 reflector-front face y<0
1, 1, 2, 101 reflector-back face y>0
1, 1, 3, 103 absorber-cylinder
```

#### **cp6.cg**

cm 6 in radius disk centered in xz plane, 1/16 in thick 11 Oct

```
1 1
1 rcc 0.00 0.00 0.00 0.00 -0.15875 0.00
      15.24 0.00 0.00 0.00 0.00 0.00
  1 1
-1
1 501 0          object
```

### **Right Circular Cylinders**

#### **rcc.cg**

cm 10.16 cm radius x 20.32 cm length solid cylinder -built 4 Sep 91

```
1 1
1 rcc 0.00 0.00 0.00 0.00 20.32 0.00
      10.16 0.00 0.00 0.00 0.00 0.00
  1 1
-1
1 501 0          cylinder
```

#### **cyl12.cg**

cm 31.27 cm x 3.8 cm rad solid cylinder, thick 11 Oct

```
1 1
1 rcc 0.00 -15.635 0.00 0.00 31.27 0.00
      3.80 0.00 0.00 0.00 0.00 0.00
  1 1
-1
1 501 0          object
```

## ***Dihedral***

### ***dihedral.cg***

```
cm 10x15 right dihedral- model built 3 Sep 91
2 2
1 box -0.1 -0.1 0.0 10.1 0.0 0.0
1 0.0 0.1 0.0 0.0 0.0 15.0
2 box -0.1 -0.1 0.0 0.1 0.0 0.0
2 0.0 10.1 0.0 0.0 0.0 15.0
1 1
2 2
-1
1 501 0 object
2 501 0 object
```

### ***dihedral.sur***

```
5 number of entries
1, 1, 3, 101 reflector-outside face xz plane
1, 1, 4, 101 reflector-inside face xz plane
2, 2, 1, 101 reflector-outside face yz plane
2, 2, 2, 101 reflector-inside face yz plane
0, 0, 0, 103 absorber-remaining edges (wildcard)
```

## ***Trihedral***

### ***trihedra.cg***

```
cm 10 cm square trihedral- model built 31 Aug 91
3 3
1 box -0.1 -0.1 -0.1 10.1 0.0 0.0
1 0.0 0.1 0.0 0.0 0.0 10.1
2 box -0.1 -0.1 -0.1 0.1 0.0 0.0
2 0.0 10.1 0.0 0.0 0.0 10.1
3 box -0.1 -0.1 -0.1 10.1 0.0 0.0
3 0.0 10.1 0.0 0.0 0.0 0.1
1 1
2 2
3 3
-1
1 501 0 plate in xz plane
2 501 0 plate in yz plane
3 501 0 plate in xy plane
```

**trihedra.sur**

7            number of entries  
1, 1, 3, 101 reflector-outside face xz plane  
1, 1, 4, 101 reflector-inside face xz plane  
2, 2, 1, 101 reflector-outside face yz plane  
2, 2, 2, 101 reflector-inside face yz plane  
3, 3, 5, 101 reflector-outside face xy plane  
3, 3, 6, 101 reflector-inside face xy plane  
0, 0, 0, 103 absorber-remaining edges (wildcard)

**Tophat**

**tophat.cg**

cm top hat 8" radius disk, 4" rad cylinder, 8" long built 10 Oct 91

2 2  
1 rcc -0.10 0.00 0.00 0.10 0.00 0.00  
          20.32 0.00 0.00 0.00 0.00 0.00  
2 rcc -0.10 0.00 0.00 20.42 0.00 0.00  
          10.16 0.00 0.00 0.00 0.00 0.00  
1 1  
2 2  
-1  
1 501 0            disk  
2 501 0            cylinder

**tophat.sur**

2            number of entries  
1, 1, 3, 103 absorber-cylinder face(edge) of disk  
0, 0, 0, 101 reflector-remaining surfaces(wildcard)

**Cone**

**cone.cg**

cm Cone - model built 19 Oct 91

1 1  
1 ehy 0.00 0.00 0.00 8.50 0.00 0.00  
          0.00 1.00 0.00 3.80 3.80 0.001  
1 1  
-1  
1 501 0            cone

**Prolate Spheroid**

*ps.cg*

```
cm Prolate Spheroid - model built 23 Oct 91
1 1
1 ell 0.00 0.00 0.00 9.00 0.00 0.00
      3.85 0.00 0.00 0.00 0.00 0.00
1 1
-1
1 501 0          prolate-spheroid
```

**Missile**

*missile.cg*

```
in Generic Missile - model built 19 Oct 91
7 6
1 epa 0.0 0.0 0.0 0.0 -3.50 0.0
1 1.0 0.0 0.0 1.5 1.5 0.0
2 rcc 0.0 0.0 0.0 0.0 18.0 0.0
2 1.5 0.0 0.0 0.0 0.0 0.0
3 rcc 0.0 0.0 0.0 0.0 18.0 0.0
3 1.1875 0.0 0.0 0.0 0.0 0.0
4 raw -0.125 18.0 1.5 0.0 -2.9375 0.0
4 0.25 0.0 0.0 0.0 0.0 2.9375
5 raw -.125 18.0 -1.5 0.0 0.0 -2.9375
5 .25 0.0 0.0 0.0 -2.9375 0.0
6 raw 1.5 18.0 -.125 0.0 -2.9375 0.0
6 2.9375 0.0 0.0 0.0 0.0 0.25
7 raw -1.5 18.0 -.125 -2.9375 0.0 0.0
7 0.0 -2.9375 0.0 0.0 0.0 0.25
1 1
2 2 -3
3 4
4 5
5 6
6 7
-1
1 501 0          nose
2 501 0          hollow cylinder
3 501 0          top fin
4 501 0          bottom fin
5 501 0          pilot's left fin
6 501 0          pilot's right fin
```

Table B.2.1. Correlation of Geometry Files and Chapter 4 Figures

Target	*.cg Filename	*.sur Filename	Chapter 4 Figure
sphere	sphere	NONE	2,3
rectangular flat plate	rpcen	rpcen	1
	rpneg	rpneg	1
	rpx10z15	rpx10z15	1,4-8,27,28,35,37
	rpx10z15	NONE	4
	rp6	NONE	53
circular flat plate	cp5	cp5	9-12,29,30,39,41
	cp6	NONE	54
right circular cylinder	rcc	NONE	13-16,31,32
	cyl12	NONE	55
dihedral	dihedral	dihedral	17-20,33,43,45
trihedral	trihedra	trihedra	21-24,47,49
tophat	tophat	tophat	25,26,51,57
cone	cone	NONE	58
prolate spheroid	ps	NONE	59
missile	missile	NONE	60

## ***Appendix C: GIFT and RADSIM Prompts***

In the process of running GIFT and RADSIM, the user must respond to 37 SRIM prompts. The following describes each prompt. The format is SRIM

PROMPT: explanation of expected response.

### ***C.1 GIFT Prompts***

1. **Geometry File Name:** enter the name of the geometry file, it must include the .cg extension.
2. **Main Option:** enter 0
3. **Application:** enter radar; there are two modules within GIFT--OPTIC and RADAR, the RADAR module creates the ray history file.
4. **Project Name:** anything goes
5. **Security Classification:** unclassified
6. **Maximum Number of Reflections:** each ray will be tracked until it leaves the target scene or encounters this number of reflections.
7. **Output File Name:** enter desired ray history file name; it must include .ray extension. If a carriage return is entered, the name will default to the same name used for the geometry file (.cg will be replaced with .ray)
8. **Comment:** any alpha-numeric comment up to 72 characters. The comment will appear in the header of the ray history file. For no comment enter carriage return.
9. **Azimuth:** enter a real value for angle (in degrees); corresponds to  $\phi$  in standard spherical coordinate system.
10. **Elevation:** enter a real value for angle (in degrees); corresponds to  $(90.^\circ - \theta)$  in standard spherical coordinate system.
11. **Maximum Horizontal Cells:** enter any negative or positive integer value. If a negative value is entered, SRIM will give a later prompt for the distance between rays. If a positive value is entered, GIFT will calculate the distance between rays based on the integer value and the size of the view plane rectangle. However, the size of the



rectangle depends upon the radar LOS. Therefore, if a positive integer value is used for this prompt, the distance between rays will depend upon the azimuth and elevation angles.

12. Maximum Vertical Cells: same as 11
13. Expand Emanation Plane: enter y to include ground to target interactions, otherwise, enter n.
14. Emanation plane horizontal scale factor: enter 1.0 for no scaling
15. Emanation plane vertical scale factor: same as 14
16. Suspend regions from view plane calculations: enter n; entering y gives the option of not including some target regions in the view plane calculations.
17. Horizontal Cell Size: enter the distance in meters between rays
18. Vertical Cell Size: same as 17
19. Enter (r), (c), or (e): enter c to continue
20. Enter application: enter end to stop program

## **C.2 RADSIM Prompts**

1. Ray history file name: enter the name of the ray history file, it must include the .ray extension.
2. Correct file: enter y or carriage return
3. Surface assignments file name: enter the name of the corresponding file, it must include the .sur extension; carriage return defaults to the geometry file name (.cg replaced with .sur).
4. Correct file: enter y or carriage return.
5. Radar system data file: enter the name of the corresponding file; carriage return defaults to radar.dat.
6. Correct file: enter y
7. Chip image for Embedding: enter n; not used for RCS calculations.
8. Polarimetric option: enter n (carriage return default); if a y is entered four cases of polarization is evaluated (HH, HV, VH, and VV).

9. Squint angle: enter 90.0 (carriage return default) for side looking SAR.
10. Image size (range, x-range): enter the number of pixels for images; not used for RCS calculations.
11. Offsets (range, x-range): enter 0.0,0.0; enables offsetting the position of the target in the image, not used in RCS calculations.
12. Image file name: carriage return defaults to the geometry file name with .cg replaced with .ci. This file is not used for RCS calculations.
13. Security classification: enter unclassified; only used for images.
14. Scattering model parameter name: enter the corresponding file name; carriage return defaults to surface.dat.
15. Correct file: enter y
16. Random number seed: enter carriage return for system generated number; not used for RCS calculations.
17. RCS log file name: enter the file name for storing the rcs data; carriage return defaults to the same name as the ray history file with the .ray replaced with .rcs. Entering an n will send the data to the screen, but not to a file.

## ***Appendix D: Automation Technique Programs***

This appendix lists the code used for the automation technique. As noted previously, running this set of codes will generate the radar.dat file. If any changes are made to the programs they must be recompiled. When recompiling, name the executable files AZIMUTH, AZIMVARY, or RCS PLOT, respectively. This is done by compiling with the command: `f77 -o filename filename.f`

### ***D.1 Rcs.script***

```
#!/bin/csh
echo '*****'
echo '      WELCOME TO RCS.SCRIPT'
echo '*****'
echo ''
echo 'This script file runs an azimuth RCS scan using SRIM.'
echo 'Please answer the following questions, pressing return'
echo 'after each response. HAVE FUN'
echo ''

# COMMENT: The FORTRAN program azimuth creates radsim.inp,
#          (the keyboard entries required to run RADSIM),
#          azimuth.cnt (file needed to run azimvary) and
#          rcsplot.inp (file needed to run rcsplot).
azimuth
echo 'Enter the number of data points shown above'
set points = $<
echo ''
echo 'Just a moment please while I do some computing.'

# COMMENT: The following while loop runs SRIM once for
#          each data point.

while ($points)

# COMMENT: The ascii geometry file must be removed each time
#          because SRIM creates a new file during each run.

# COMMENT: The FORTRAN program azimvary inputs the keyboard
#          entries for GIFT and increments the azimuth angle.
```

azimvary

# COMMENT: The following two statements run SRIM.

```
/shared/srim40/bin/gift < gift.inp > giftrad.log  
/shared/srim40/bin/radsim < radsim.inp > radsim.log
```

# COMMENT: The ray history and complex image files must be  
# removed after each SRIM run because they are created  
# during each run.

# COMMENT: The rcs output file created during each SRIM run is  
# being appended to the previous run. The file created  
# by SRIM is then removed prior to the next run.

```
cat *.rcs >> rcs.dat  
rm *.4  
rm *.ray  
rm *.ci  
rm *.rcs
```

# COMMENT: The while counter is being decremented by one.

@ points--  
end

# COMMENT: The FORTRAN program rcsplot strips the rcs output  
# file created by SRIM and only gives the frequency,  
# elevation, azimuth, polarization, and rcs (real and  
# complex) values.

```
rcsplot  
echo ''  
echo 'The RCS data is now stored under the output file name you'  
echo 'selected earlier. Thanks'  
echo ''
```

# COMMENT: The rcs.dat file is being removed in preparation  
# for the next input geometry file.

```
rm rcs.dat
```

## ***D.2 Azimuth.f***

### **PROGRAM AZIMUTH**

\* The purpose of this FORTRAN program (azimuth.f) is to  
\* select: 1) the input geometry file name, 2) output rcs data  
\* file name, 3) azimuth starting location (in degrees),  
\* 4) azimuth ending location (in degrees), and 5) the interval

```

* (in degrees) between azimuth data points. It also calculates
* the ray spacing (in meters) between rays for a user specified
* number of rays per wavelength, and outputs the radar.dat file
* used by RADSIM.

* Input files:
* none - only keyboard entries are required
* Output files produced:
* radsim.inp (unit 7) - this file is used to run RADSIM
* rcsplot.inp (unit 8) - this file is used for rcsplot.f
* azimuth.cnt (unit 9) - this file is used for azimvary.f
* radar.dat (unit 10) - this file is used for radsim
* Variable Declaration
  character*20 ray_history_file,surface_assign_file
  character*15 correct_file, radar_sys_file, polar_option,
+ chip_image, offset, image_size, security_class,
+ scattering_model_file, random_seed, rcs_log_file,
+ input_geometry_file, output_rcs_file
  character*3 ray, dot_sur_file
  real azimuth_start,azimuth_end,increment,number_points
  real freq,number_rays,ray_spacing,wavelength,elevation
* Open Files
  open(7,file='radsim.inp',status='unknown')
  open(8,file='rcsplot.inp',status='unknown')
  open(9,file='azimuth.cnt',status='unknown')
  open(10,file='radar.dat',status='unknown')

* Reading the input from the keyboard
  print *, 'What is the input geometry file name? It must be *.cg'
  read *,input_geometry_file
  print *, ''
  print *, 'If using a .sur file input y, otherwise press n'
  read *,dot_sur_file
  print *, ''
  print *, 'What is the radar frequency (GHz)?'
  read *, freq
  wavelength=2.9979*10.**8/(freq*10.**9)
  print *, ''
  print *, 'How many rays do you want per wavelength?'
  read *, number_rays
  print *, ''
  ray_spacing=wavelength/number_rays
  print *, 'What is the desired output RCS file name?'
  read *,output_rcs_file
  print *, ''
  print *, 'What is the elevation angle (measured from xy plane)?'
  read *,elevation
  print *, ''
  print *, 'What is the beginning azimuth location in degrees?'

```

```

read *, azimuth_start
print *, ''
print *, 'What is the ending azimuth location in degrees?'
read *, azimuth_end
print *, ''
print *, 'The increment (in degrees) between azimuth data
+      points'
print *, 'can be 1 or .5. Which increment do you desire?'
read *, increment
print *, ''

```

```

* Calculating the number of data points and printing to screen
  if (azimuth_end .lt. azimuth_start) then
    number_points=(360.+(azimuth_end-azimuth_start))/increment+1.
  else
    number_points=(azimuth_end-azimuth_start)*1./increment+1.
  endif
print *, '*****'
print *, 'The number of data points is: ', number_points
print *, '*****'
print *, ''

```

\* Variable Initialization for Radsim.inp.

```
ray='ray'
```

\* This do loop creates the ray history surface file names based on the  
\* input file name.

```

do 5 i=1,20
  if (input_geometry_file(i:i) .eq. '.') then
    ray_history_file=input_geometry_file(1:i)//ray
    surface_assign_file=input_geometry_file(1:i)//sur
    goto 6
  endif

```

```
5 continue
```

```
6 correct_file='y'
```

```
if (dot_sur_file .ne. 'y') surface_assign_file=''
```

```
radar_sys_file=''
```

```
polar_option='n'
```

```
chip_image='n'
```

```
squint_angle=90.
```

```
image_size='200,200'
```

```
offset='0.,0.'
```

```
image_file=''
```

```
security_class='unclassified'
```

```
scattering_model_file=''
```

```
random_seed=''
```

```
rsc_log_file=''
```

\* Output of RADSIM.INP

```
write(7,10) ray_history_file,correct_file,surface_assign_file,
```

```

+      correct_file,radar_sys_file,correct_file,
+      polar_option,chip_image,squint_angle,image_size,
+      offset,image_file,security_class,
+      scattering_model_file,correct_file,random_seed,
+      rcs_log_file
10 format(A/,A/,A/,A/,A/,A/,A/,A/,F6.1/,A/,A/,A/,A/,A/,A/,A/,A)

* Output of rcsplot.inp (unit 8) and azimuth.cnt (unit 9) files.
  write(8,20) azimuth_start,number_points,increment,
+      output_rcs_file
20 format(F6.1/,F4.0/,F4.2/,A)
  write(9,30) input_geometry_file,azimuth_start,elevation,
+      ray_spacing,increment
30 format(A/,F6.1/,F5.1/,F7.5/,F4.2)

* The following creates radar.dat. For vertical polarization,
* TTAU and RTAU must be set to 90.0 (as shown below). For
* horizontal polarization, change TTAU and RTAU to 0.0 and
* recompile the file.

* Output of Radar.dat
  write(10,40) "'WAVLEN'",wavelength,"'TTAU' 90.",
+      "'TEPSI' 0.",
+      "'RTAU' 90.", "'REPSI' 0.", "'NOISE' ", "' ",
+      "'DIGITAL RESP.'", "' ' ' ",
+      "'IPR PARAM.', 'RANGE', 'X-RANGE'",
+      "'RESOL' 5.0,5.0', 'SPACEN', .05,.05',
+      "'NTAYLOR' 1.,1.', 'SLL', -25.,-25.',
+      "'CONV. CELLS' 0.,0.', 'QUANT. FACT.' 10.,10.'
40 format(A,F6.5/,A/,A/,A/,A/,A/,A/,A/,A/,A/,A/,A/,A/,A/,A/)
  end

```

### ***D.3 Azimvary.f***

#### **PROGRAM AZIMVARY**

```

* The purpose of this FORTRAN program (azimvary.f) is
* to create the input file, gift.inp, (the keyboard
* entries) used by SRIM in the GIFT RADAR module.
* The sole function of the program is to create the
* file for each azimuth angle. As such, this program
* is run, via rcs.script, for each azimuth location.
* Input Required:
*   data file (unit 7) - azimuth.cnt
*   This contains input geometry file name, starting
*   azimuth location, ray spacing between rays (meters),
*   and the increment between data points, and elev.
* Output
*   data file (unit 9) - azimuth.cnt

```

```

*      Same as above except azimuth is incremented by
*      interval between data points
*      data file (unit 8) - gift.inp
*      Keyboard entries for GIFT RADAR
*****
*      Variable Declaration
character*20 geometry_input_file,application1,project,
+      classification,output_file,comment,
+      expand_eman_plane,suspend_regions,
+      run_cont_end,application2
integer option,reflections,max_horz_cells,max_vert_cells
real azimuth,elevation,eman_plane_horz_sf,eman_plane_vert_sf,
+  horz_cell_size,vert_cell_size,increment,ray_spacing

*      Open files
open(7,file='azimuth.cnt',status='old')
open(8,file='gift.inp',status='unknown')
open(9,file='azimuth.cnt',status='old')

*      Variable Initialization
read(7,*) geometry_input_file
option=0
application1='radar'
project='RCS'
classification='unclassified'
reflections=9
output_file=' '
comment=' '
read(7,*) azimuth
print *,'Azimuth: ',azimuth
read(7,*) elevation
max_horz_cells=-1
max_vert_cells=-1
expand_eman_plane='n'
eman_plane_horz_sf=1.
eman_plane_vert_sf=1.
suspend_regions='n'
read(7,*) ray_spacing
horz_cell_size=ray_spacing
vert_cell_size=ray_spacing
run_cont_end='c'
application2='end'
write(8,10)geometry_input_file,option,application1,project,
+      classification,reflections,output_file,comment,
+      azimuth,elevation,max_horz_cells,max_vert_cells,
+      expand_eman_plane,eman_plane_horz_sf,
+      eman_plane_vert_sf,suspend_regions,horz_cell_size,
+      vert_cell_size,run_cont_end,application2
10 format(A,I1/A/A/A,I3/A/A,F6.1/F6.1/I2/I2/A/

```



```

+      F4.2/,F4.2/,A/,F8.5/,F8.5/,A/,A)
  read(7,*) increment
  azimuth=azimuth+increment
  write(9,20) geometry_input_file,azimuth,elevation,ray_spacing,
+      increment
20 format(A/,F6.1/,F5.1/,F7.5/,F4.2)
  end

```

#### **D.4 Rcsplot.f**

```

PROGRAM RCSPLOT
* The purpose of this FORTRAN program (rcsplot.f) is to read
* in the rcs data file (rcs.dat) created by appending the
* azimuth SRIM RADSIM runs. The data file is stripped of
* everything except wavelength, elevation, azimuth,
* polarization, and target rcs values (real and complex).
* Input Files:
*   rcs.dat (unit 7)
*   This is the file created by appending each
*   of the SRIM rcs run
*   rcsplot.inp (unit 8)
*   This file is created by azimuth.f and contains
*   azimuth starting position (azimstart), the number
*   of data points (number_points), the increment
*   between data points (increment), and the output
*   rcs file name (output_rcs_file).
* Output File:
*   output_rcs_file (unit 9)
*   This is the file that contains the stripped
*   rcs data. The name of the file is read in from
*   rcsplot.inp.

* variable declaration
  character*10 ignore_a,receiver_polar,transmitter_polar
  character*20 output_rcs_file
  real azimstart,wavlen,azim,elev,ignore_r,rcstotmm,rcs_real,
+   freq,rcstotdb,rcs_imag,number_points,azimct,increment
  integer j

* Open data files.

  open(7,file='rcs.dat',status='old')
  open(8,file='rcsplot.inp',status='old')

* The following reads in the starting azimuth location, the number
* of data points, the increment between points, and the rcs output
* file name
  read(8,10) azimstart,number_points,increment,output_rcs_file

```

```
10 format(F6.1/,F4.0/,F4.2/,A)
```

- \* Opening the output data file  
    open(9,file=output\_rcs\_file,status='unknown')
- \* The following statements read in the data file created by RADSIM.
- \* ignore\_a and ignore\_r are dummy variables for info that is to be
- \* discarded. The total target rcs(m<sup>2</sup>) is stored in rcstotmm,
- \* the real part of the complex rcs is stored in rcs\_real, the
- \* imaginary part of the complex rcs is stored in rcs\_imag.

```
    read(7,*) ignore_a  
    read(7,*) ignore_a
```

- \* The if loop is required because if the azimuth angle is negative
- \* the negative sign is read into a memory cell  
    if (azimstart .lt. -1.) then  
        read(7,\*) ignore\_a,ignore\_a,wavlen,ignore\_a,ignore\_a,  
        +        ignore\_a,azim,ignore\_a,elev
- \* The negative sign on the azimuth angle is ignored and azimuth is
- \* multiplied by -1 for the correct sign.

```
        azim=azim*-1.  
    else  
        read(7,*) ignore_a,ignore_a,wavlen,ignore_a,  
        +        ignore_a,azim,ignore_a,elev  
    endif
```

```
    azimstart=azimstart+increment  
    read(7,*) ignore_a  
    read(7,*) ignore_a  
    read(7,*) ignore_a  
    read(7,*) ignore_a,ignore_r,rcstotmm  
    read(7,*) ignore_a  
    read(7,*) ignore_a  
    read(7,*) ignore_a  
    read(7,5) ignore_a,ignore_r,rcs_real,rcs_imag
```

- 5 format(/,T1,A3, e8.2, T23,e8.2, T32,e8.2)
- read(7,\*) ignore\_a  
    read(7,\*) ignore\_a  
    read(7,\*) ignore\_a  
    read(7,\*) ignore\_a  
    read(7,\*) ignore\_a,ignore\_r,ignore\_r,receiver\_polar  
    if (receiver\_polar .eq. 'H') then  
        receiver\_polar='Horizontal'
- else  
        receiver\_polar='Vertical'
- endif  
    read(7,\*) ignore\_a,ignore\_r,ignore\_r,transmitter\_polar

```

if (transmitter_polar .eq. 'H') then
  transmitter_polar='Horizontal'
else
  transmitter_polar='Vertical'
endif

* Calculate frequency in GHz and RCS in dBsm
freq=2.9979e8/wavlen/1.e9
rcstotdb=10*alog10(rcstotmm)

* Write out data
write(9,20) freq,wavlen,elev,receiver_polar,transmitter_polar
20 format('Frequency (GHz): ',F6.2/,'Wavelength (meters): ',
+F6.4/,'Elevation (degrees): ',F4.1/,
+'Receiver Polarization: ',A/,
+'Transmitter Polarization: ',A/)
write(9,30)
30 format(T1,'Azimuth(degrees)',T20,'RCSTOT(dBsm)',T35,
+'RCSTOT(mm)',T49,'Real RCS (mm)',T63,'Imag RCS(mm)',/)
write(9,40) azim,rcstotdb,rcstotmm,rcs_real,rcs_imag
40 format(T6,F5.1,T19,F10.5,T33,F10.5,T51,F6.3,T65,F6.3)

* Complete the following loop for subsequent azimuth runs
do 100 azimct=2,number_points
  read(7,*) ignore_a
  read(7,*) ignore_a
  if (azimstart .le. -1.) then
    read(7,*) ignore_a,ignore_a,ignore_r,ignore_a,
+      ignore_a,ignore_a,azim
    azim=azim*-1.
  else
    read(7,*) ignore_a,ignore_a,ignore_r,ignore_a,
+      ignore_a,azim
  endif
endif

* The following loop is required because SRIM gives erroneous
* negative azimuth locations if the azimuth values are not
* integers. SRIM gives azimuth locations between -1 and 0
* as 0.
  if (increment .ne. 1. .and. azimstart .le. 0.) then
    azim=azimstart
  endif

  azimstart=azimstart+increment
  read(7,*) ignore_a
  read(7,*) ignore_a
  read(7,*) ignore_a
  read(7,*) ignore_a,ignore_r,rcstotmm
  read(7,*) ignore_a

```

```
    read(7,*) ignore_a
    read(7,*) ignore_a
    read(7,5) ignore_a,ignore_r,rCS_real,rCS_imag
    do 50 j=1,6
        read(7,*) ignore_a
50    continue

* Calculate RCS in dBsm
    rcstotdb=10*alog10(rcstotmm)

* Write out data
    write(9,40) azim,rcstotdb,rcstotmm,rCS_real,rCS_imag

100 continue
end
```

## ***Appendix E: Closed Form PO Equations***

The FORTRAN programs written during this effort are presented in this section. The equation references are documented within each program.

### ***E.1 Sphere Source Code, Sphere.f***

#### **Program Sphere**

\*\*\*\*\*

\* This FORTRAN program (sphere.f) calculates the RCS of sphere  
\* using the PO equation given in Knott's textbook, Radar Cross  
\* Section, Page 129, Eq 5-31. In his development, the phase  
\* has been referenced to the center of the sphere.

\*

\* Input Files Required:

\* None. The program input parameters, sphere radius  
\* (meters) and radar frequency (GHz) are provided via  
\* program prompts.

\* Output Files Produced:

\* sphere.dat - this file contains the RCS in units of dBsm

\*\*\*\*\*

```
real a,freq,pi,c,k,ka2,rCS
complex j,sigma
open(7,file='sphere.dat',status='unknown')
print *,'Input the radius in meters'
read *,a
print *,'Input the frequency in GHz'
read *,freq
freq=freq*10.**9
pi=3.1416259
c=3.*10.**8
k=2.*pi*freq/c
ka2=2.*k*a
j=(0.0,1.0)
sigma=sqrt(pi)*a*((1.-j/ka2)*cexp(-j*ka2)+j/ka2)
```

\* Converting to dBsm. Using 20 because sigma is in terms of  
\* square root.

```
rCS=20.*alog10(cabs(sigma))
```

\* Printing output data to screen and to file sphere.dat

```
write(7,*) rCS
print *, rCS
end
```

## ***E.2 Rectangular Plate Source Code, Rplate.f***

```
PROGRAM rplate
*****
* This FORTRAN program (rplate.f) calculates the RCS (dBsm)
* of a rectangular flat plate oriented in the XZ plane in one
* degree increments from 0 to 90 degrees. The PO equation used
* was originally taken from Dennis Blejer's paper "Physical
* Optics Polarization Scattering Matrix of a Square Trihedral
* Corner Reflector," unpublished report No. 47PM-ADT-0059, MIT,
* Lincoln Laboratory, 30 Jun 87. Capt Dennis Young modified the
* equation in his thesis, "Primitive Target Validation for the
* Signature Prediction Tools Software Package,"
* AFIT/GE/ENG/90D-68, Dec 90.

* Input Files Required:
*   rplate.inp - this file must contain the following
*               parameters on separate lines
*               a - X dimension in cm
*               b - Z dimension in cm
*               f - radar frequency in GHz
*               elev - elevation angle measured from XY
*                   plane in degrees

* Output Files Produced:
*   rplate.dat - this file is a 2 column array with the first
*               column containing the azimuth angle and the
*               second column containing the RCS in dBsm
*****

* Variable Declaration
  real a,b,f,elev,theta,pi,sqrtpi,c,k,ka,kb,sqrtab
  real phir,thetar,cosphi,sinphi,costheta,sintheta
  real sincl,sinc2,rcs
  integer i
  complex j,jkab,expl,sigxz

* Open input and output files
  open(7,file='rplate.inp',status='old')
  open(8,file='rplate.dat',status='unknown')

* Read in parameters. Each entry must include decimal point.
  read(7,*) a,b,f,elev
  5 format(F9.6/,F9.6/,F7.4/,F5.1)
* Converting to meters and GHz
  a=a/100.
  b=b/100.
  f=f*10.**9
```

```

theta=90.-elev

* Variable Initialization
pi=3.1415926
sqrtpi=sqrt(pi)
c=2.998e8
j=(0.0,1.0)
k=2.*pi*f/c
ka=k*a
kb=k*b
jkab=j*k*a*b
sqrtab=sqrt(a*a+b*b)

* Begin phi loop
do 100 i=0,90
  phi=float(i)

* Avoidance of Singularities
  if (phi .eq. 0.0) phi=.001
  if (theta .eq. 0.0) theta=.001
  if (theta .eq. 90.) theta=89.999

* Converting to Radians
  phir=phi*pi/180.
  thetar=theta*pi/180.

  cosphi=cos(phir)
  sinphi=sin(phir)
  costheta=cos(thetar)
  sintheta=sin(thetar)

  sinc1=sin(kb*costheta)/(kb*costheta)
  sinc2=sin(ka*sintheta*cosphi)/(ka*sintheta*cosphi)
  exp1=cexp(j*k*sqrtab*(sintheta*cosphi+costheta))

* Equation 3.24 of Capt Young's Thesis
  sigxz=-jkab/sqrtpi*sintheta*sinphi*sinc1*sinc2*exp1

* Converting to dBsm. Using 20 because sigxz is in terms
* of square root
  rcs=20.*alog10(cabs(sigxz))

  write(8,10) float(i),rcs
100 continue
10 format(F4.1,3X,F10.5)
end

```

### ***E.3 Circular Plate Source Code, Cplate.f***

```
PROGRAM cplate
*****
* This FORTRAN program (cplate.f) calculates the rcs (dBsm) of
* a circular plate oriented in the xz plane. Values are calculated
* from 0 to 90 degrees in one degree increments. The PO equation for
* the disk is taken from Ruck's book, Radar Cross Section, page 513
* (slightly modified as shown in Chapter Two of this thesis).
*****
* Input Files Required:
*   None. The user provides the input parameters by answering
*   prompts. The input parameters are:
*       a - plate radius in centimeters;
*       freq - frequency in GHz;
*       elev - elevation angle in degrees measured from the xy
*             plane.
* Output Files Produced:
*   cplate.dat - includes the azimuth angle and RCS (dBsm).
*****

* Variable Declaration
  real pi,area,a,freq,theta,elev,k,lambda,phi,r,kasinpsi
  real rcsdb,sigma,bes1,arg,bessel,sintheta,sinphi,sinpsi
  real cospsi

* Open output file
  open(7,file='cplate.dat',status='unknown')

* Variable Initialization and user inputs
  pi=3.1415926
  r=pi/180.
  print *,'Input the disk radius in centimeters'
  read *,a
  a=a/100.
  area=pi*a**2
  print *,'Input the radar frequency in GHz'
  read *,freq
  freq=freq*10**9
  lambda=3.*10.**8/freq
  k=2.*pi/lambda
  print *,'Input the elevation angle in degrees (measured'
  print *,'from xy plane)'
  read *,elev
  theta=90.-elev
  sintheta=sin(theta*r)

* Phi Loop
  do 10 phi=0.,90.
```



```

sinphi=sin(phi*r)
sinpsi=sin(acos(sintheta*sinphi))
kasinpsi=k*a*sinpsi
cospsi=sintheta*sinphi
arg=2.*kasinpsi
if (kasinpsi .eq. 0.) kasinpsi=1.
bessel=bes1(arg,1)/kasinpsi
if (bessel .eq. 0.) bessel=1.
sigma=4.*pi*area**2*cospsi**2/lambda**2
+      *bessel**2
rcsdb=10*alog10(sigma)
write(7,100) phi,rcsdb
10 continue
100 format(F5.1,3x,F14.5)
end

```

```

*****
* This function computes the Bessel function of the first
* kind and order zero or one. The input parameters are
*   x = argument of Bessel function
*     (real and greater than or equal to zero)
*   n = order of Bessel function
*     (zero or one)
*
* This coded function was given to the students in the AFIT
* EENG630 course during the Winter 91 term
*****

```

```

function bes1(x,n)
y=(x/3.0)**2
if(x.gt.3.0) go to 1
if(n.eq.0) then
a0=1.0
a1=-2.2499997
a2=1.2656208
a3=-0.3163866
a4=0.0444479
a5=-0.0039444
a6=0.0002100
else
a0=0.5
a1=-0.56249985
a2=0.21093573
a3=-0.03954289
a4=0.00443319
a5=-0.00031761
a6=0.00001109
endif
bes1=a0+y*(a1+y*(a2+y*(a3+y*(a4+y*(a5+y*a6))))))

```

```

    if(n.eq.1) bes1=x*bes1
    return
1  y=3.0/x
    sx=sqrt(x)
    if(n.eq.0) then
        a1=-0.00000077
        a2=-0.00552740
        a3=-0.00009512
        a4=0.00137237
        a5=-0.00072805
        a6=0.00014476
        b0=-0.78539816
        b1=-0.04166397
        b2=-0.00003954
        b3=0.00262573
        b4=-0.00054125
        b5=-0.00029333
        b6=0.00013558
    else
        a1=0.00000156
        a2=0.01659667
        a3=0.00017105
        a4=-0.00249511
        a5=0.00113653
        a6=-0.00020033
        b0=-2.35619449
        b1=0.12499612
        b2=0.00005650
        b3=-0.00637879
        b4=0.00074348
        b5=0.00079824
        b6=-0.00029166
    endif
    f=0.79788456+y*(a1+y*(a2+y*(a3+y*(a4+y*(a5+y*a6))))
    thta=x+b0+y*(b1+y*(b2+y*(b3+y*(b4+y*(b5+y*b6))))
    bes1=f*cos(thta)/sx
    return
end

```

#### ***E.4 Solid Right Circular Cylinder Source Code, Rccpo.f***

```

PROGRAM rccpo
*****
* This fortran program (rccpo.f) calculates the rcs (dBsm) of
* a solid right circular cylinder axially aligned with the
* Y axis. The PO solutions for a circular disk and cylinder
* are blended together for the final solution (i.e. they are
* not coherently summed). Values are calculated from 0 to 90

```

```

* degrees in one degree increments. The equation for the
* cylinder is taken from Knott's textbook, Radar Cross Section,
* page 126. The equation for the disk is taken from Ruck's
* book, Radar Cross Section, page 513 (slightly modified as
* shown in Chapter Two of this thesis.
* Input Files Required:
*   rccpo.inp - this file must contain the following
*               information on subsequent lines:
*               a - radius in cm
*               l - length of cylinder in cm
*               f - frequency in GHz
*               elev - elevation in degrees measured from
*                   XY plane
* Output Files Produced:
*   rccpo.dat - this file contains a two column array with the
*               first column containing the azimuth and the
*               second column containing the rcs (dBsm)
*****

* Variable Declaration
  real pi,r,a,l,freq,elev,area,lambda,k,theta,sintheta
  real phi,sinphi,sinpsi,kasinpsi,cospsi,arg,bessel
  real sigmapl,sinc,sigmacyl,sigmatot,rcstotdb
  integer i

* Open output file
  open(7,file='rccpo.inp',status='old')
  open(8,file='rccpo.dat',status='unknown')

* Read in data. Each parameter must have a decimal point
  read(7,5) a,l,freq,elev
5  format(F9.5/,F9.5/,F4.1/,F4.1)
  a=a/100.
  l=l/100.
  freq=freq*10.**9
  theta=90.-elev

* Variable Initialization
  pi=3.1415926
  r=pi/180.
  area=pi*a**2
  lambda=3.*10.**8/freq
  k=2.*pi/lambda
  sintheta=sin(theta*r)

* Phi Loop
  do 10 i=0,90
    phi=float(i)

```

```

* Cylinder Calculations
  sinphi=sin(phi*r)
  if (k*l*sinphi .eq. 0.) then
    sinc=1.
  else
    sinc=sin(k*l*sinphi)/(k*l*sinphi)
  endif
  if (phi .le. 45.) then
    sigmacyl=k*a**2*cos(phi*r)**2*sinc**2
  else
    sigmacyl=0.
  endif

* Circular Plate Calculations
  sinpsi=sin(acos(sintheta*sinphi))
  kasinpsi=k*a*sinpsi
  cospsi=sintheta*sinphi
  arg=2.*kasinpsi
  if (kasinpsi .eq. 0.) kasinpsi=1.
  besse1=bes1(arg,1)/kasinpsi
  if (besse1 .eq. 0.) besse1=1.
  if (phi .ge. 46) then
    sigmapl=4.*pi*area**2*cospsi**2/lambda**2
+      *besse1**2
  else
    sigmapl=0.
  endif

* Combining the results
  sigmatot=sigmapl+sigmacyl
  rcstotdb=10.*alog10(sigmatot)
  write(8,100) phi,rcstotdb
10 continue
100 format(F5.1,3x,F14.5)
end

```

```

*****
* This function computes the Bessel function of the first
* kind and order zero or one. The input parameters are
*   x = argument of Bessel function
*   (real and greater than or equal to zero)
*   n = order of Bessel function
*   (zero or one)
*
* This coded function was given to the students in the AFIT
* EENG630 course during the Winter 91 term
*****

```

```

function bes1(x,n)
y=(x/3.0)**2
if(x.gt.3.0) go to 1
if(n.eq.0) then
  a0=1.0
  a1=-2.2499997
  a2=1.2656208
  a3=-0.3163866
  a4=0.0444479
  a5=-0.0039444
  a6=0.0002100
else
  a0=0.5
  a1=-0.56249985
  a2=0.21093573
  a3=-0.03954289
  a4=0.00443319
  a5=-0.00031761
  a6=0.00001109
endif
bes1=a0+y*(a1+y*(a2+y*(a3+y*(a4+y*(a5+y*a6))))
if(n.eq.1) bes1=x*bes1
return
1 y=3.0/x
sx=sqrt(x)
if(n.eq.0) then
  a1=-0.00000077
  a2=-0.00552740
  a3=-0.00009512
  a4=0.00137237
  a5=-0.00072805
  a6=0.00014476
  b0=-0.78539816
  b1=-0.04166397
  b2=-0.00003954
  b3=0.00262573
  b4=-0.00054125
  b5=-0.00029333
  b6=0.00013558
else
  a1=0.00000156
  a2=0.01659667
  a3=0.00017105
  a4=-0.00249511
  a5=0.00113653
  a6=-0.00020033
  b0=-2.35619449
  b1=0.12499612
  b2=0.00005650

```

```

      b3=-0.00637879
      b4=0.00074348
      b5=0.00079824
      b6=-0.00029166
endif
f=0.79788456+y*(a1+y*(a2+y*(a3+y*(a4+y*(a5+y*a6))))
thta=x+b0+y*(b1+y*(b2+y*(b3+y*(b4+y*(b5+y*b6))))
bes1=f*cos(thta)/sx
return
end

```

### ***E.5 Right Dihedral Source Code, Dihedral.f***

#### **PROGRAM DIHEDRAL**

```

*****
* This Fortran program (dihedral.f) uses Physical Optics to
* calculate the HH and VV components of a right trihedral. The
* dihedral is aligned so the faces are in the xz and yz planes,
* with the dihedral corner being along the z axis. The program
* calculates the the RCS values for a user specified theta angle
* for phi angles from 0 to 45 in one degree increments.
* Symmetry is then used for phi angles greater than 45.

* This program is based on the equations in 'Physical Optics
* Polarization Scattering Matrix and Impulse Response of a
* Right Dihedral Corner Reflector', dated 22 Dec 86, authored
* by D. J. Blejer, of MIT Lincoln Laboratory, ADT Project
* Memorandum No. 47PM-ADT-0052. Hereafter, referred to as
* Reference 1.

* Furthermore, this is a modified version of the fortran code
* from Capt Dennis Young's thesis, 'Primitive Target Validation
* for the Signature Prediction Tools Software Package',AFIT/
* GE/ENG/90D-68.

* Input File Required
*   dihed.inp - this data file must contain one entry per line
*               consisting of the following:
*               a - dimension in cm of dihedral width (X,Y axes)
*               b - dimension in cm of dihedral length (Z axis)
*               f - radar frequency in GHz
*               elev - angle of incidence in degrees,
*                   referenced from XY plane
* Output File Produced
*   dihed.dat - this is a three column data file listing
*               the azimuth angle, HH (dBsm), and VV (dBsm),
*               respectively
*****

```

```

* Variable Declaration
  real a,b,f,theta,pi,sqrtpi,c,k,ka,kb,phi,phir,thetar
  real cosphi,sinphi,costheta,sintheta,d,kd,HHrcs,VVrcs
  real sinc1,sinc2,sinc3,sinc4,elev
  integer i
  complex j,abjk,HH,VV,exp1,exp2,exp3,exp4

* Open input and output files
  open(7,file='dihed.inp',status='old')
  open(8,file='dihed.dat',status='unknown')

* Read in parameters. Each entry must include decimal point.
  read(7,5) a,b,f,elev
5 format(F9.6/F9.6/F7.4/F5.1)
  a=a/100.
  b=b/100.
  f=f*10.**9
  theta=90.-elev

* Variable Initialization
  pi=3.1415926
  sqrtpi=sqrt(pi)
  c=2.998e8
  j=(0.0,1.0)
  k=2.*pi*f/c
  ka=k*a
  kb=k*b
  abjk=a*b*j*k
  HH=(0.0,0.0)
  VV=(0.0,0.0)

* Begin phi loop
  do 100 i=0,90
    phi=float(i)

* Symmetry being used for phi>45 or theta>90
    if (phi .gt. 45.) phi=90.-phi
    if (theta .gt. 90.) theta=180.-theta

* Avoidance of Singularities
    if (phi .eq. 0.0) phi=.001
    if (theta .eq. 0.0) theta=.001
    if (theta .eq. 90.) theta=89.999

* Conversion to Radians
  phir=phi*pi/180.
  thetar=theta*pi/180.

```

\* More Variable Initialization

```

cosphi=cos(phir)
sinphi=sin(phir)
costheta=cos(thetar)
sintheta=sin(thetar)

```

\* Equation (3) from Reference 1

```

d=a*costheta/(cosphi*sintheta)
kd=k*d

```

```

sinc1=sin(kb*costheta)/(kb*costheta)
sinc2=sin(ka*cosphi*sintheta)/(ka*cosphi*sintheta)
sinc3=sin(ka*sinphi*sintheta)/(ka*sinphi*sintheta)
sinc4=sin(kd*costheta)/(kd*costheta)

```

```

exp1=cexp(j*ka*cosphi*sintheta)
exp2=cexp(j*ka*sinphi*sintheta)
exp3=cexp(-j*kb*costheta)
exp4=cexp(-j*kd*costheta)

```

Comment: In the following two statements, Capt Young's program c had a minus sign before the sinc2 and cosphi terms. In c Reference 1 the minus signs are not present.

\* Equation (5) from Reference 1

```

HH=abjk/sqrtpi*(sintheta*sinc1*(sinc2*sinphi*exp1
+   +cosphi*sinc3*exp2+2.*sinphi)+sintheta*sinphi
+   *exp3/(j*kb*costheta)*(1.-exp4*sinc4))

```

\* Equation (6) from Reference 1

```

VV=abjk/sqrtpi*(sintheta*sinc1*(sinc2*sinphi*exp1
+   +cosphi*sinc3*exp2-2.*sinphi)-sintheta*sinphi
+   *exp3/(j*kb*costheta)*(1.-exp4*sinc4))

```

\* Converting to dBsm. Using 20 because HH and VV are in terms of square root

```

HHrcs=20.*alog10(cabs(HH))
VVrcs=20.*alog10(cabs(VV))
write(8,10) float(i),HHrcs,VVrcs
100 continue
10 format(F4.1,3X,F10.5,3X,F10.5)
end

```



## ***E.6 Square Trihedral Source Code, Trihed.f***

```
PROGRAM trihed
*****
* This Fortran program (trihed.f) uses Physical Optics to
* calculate the HH and VV RCS (dBsm) components of a square
* trihedral. The trihedral is aligned so the plates are in
* the xz, yz, and xy planes. The vertical crease is along
* the z axis. The program calculates the RCS for a user
* specified theta, frequency, and length for phi angles from
* 0 to 45 in one degree increments. Symmetry is then used
* for phi angles from 46 to 90 degrees.

* This program is based on the equations in three papers
* submitted by D. J. Blejer of MIT Lincoln Laboratory. The
* paper titles, dates, and ADT Project Memorandum Numbers are:
* 'Physical Optics Polarization Scattering Matrix of a Square
* Trihedral Corner Reflector', 30 Jun 87, 47PM-ADT-0059;
* 'Physical Optics Polarization Scattering Matrix and Impulse
* Response of a Right Dihedral Corner Reflector', 22 Dec 86,
* 47PM-ADT-0052; 'Scattering Matrix Transformations for
* Polarimetric Target Modeling, 20 May 87, 47PM-ADT-0057.
* Hereinafter, the papers are referred to as References 1, 2,
* and 3, respectively.

* Furthermore, this is a modified version of the fortran code
* from Capt Dennis Young's thesis, 'Primitive Target Validation
* for the Signature Prediction Tools Software Package', AFIT/
* GE/ENG/90D-68.
*
* Input File Required
*   trihed.inp - this data file must contain one entry per
*               line listed in the following order:
*               L   - dimension in cm of trihedral
*               f   - radar frequency in GHz
*               elev - angle of incidence in degrees,
*                   measured from XY plane
*
* Output File Produced
*   trihed.dat - this is a seven column data file listing
*               the azimuth angle, HH RCS (dBsm), and VV RCS
*               (dBsm), Single bounce (dBsm), Triple bounce
*               (dBsm), HH Double bounce (dBsm), and VV
*               Double bounce (dBsm), respectively.
*****

* Variable Declaration
  real L,f,elev,theta,pi,sqrtpi,c,k,kL,phi,rcsHH,rcsVV
  real rcsSB,rcsTB,rcsDBHH,rcsDBVV
```

```
complex j,Ss,St,sigDBHH,SdHH,SdVV
complex RTpdotRpHH,RTpdotRpVV,RTppdotRppHH,RTppdotRppVV
complex HHtot,VVtot
integer i
```

\* Common Variables Passed to Subroutines

```
common pi,sqrtpi,k,L,kL,j
```

\* Open input and output files

```
open(7,file='trihed.inp',status='old')
open(8,file='trihed.dat',status='unknown')
```

\* Read in Parameters. Each entry must include a decimal point.

```
read(7,5) L,f,elev
5 format(F9.6,/F9.6/,F5.1)
L=L/100.
f=f*10.**9
theta=90.-elev
```

\* Initialize Variables

```
pi=3.1415926
sqrtpi=sqrt(pi)
c=2.998e8
j=(0.0,1.0)
k=2.0*pi*f/c
kL=k*L
```

\* Begin phi loop

```
do 10 i=0,90
  phi=float(i)
```

\* Using Symmetry for phi>45 and theta>90

```
if (phi .gt. 45.0) phi=90.-phi
if (theta .gt. 90.) theta=180.-theta
```

\* Avoiding Singularities

```
if (phi .eq. 0.0) phi=.001
if (phi .eq. 90.0) phi=89.999
if (theta .eq. 0.0) theta=.001
if (theta .eq. 90.0) theta=89.999
```

\* Call Subroutines for the Polarization Scattering Matrix

```
call Singlebounce(theta,phi,Ss)
call Triplebounce(theta,phi,St)
call Doublebounce(theta,phi,sigDBHH)
```

```

* Equation 34 of Reference 1
  SdHH=sigDBHH*cexp(j*k*L*cos(theta*pi/180.))
  SdVV=-SdHH

* Call Subroutines for the Horizontal Dihedral Scattering
* Matrices
  call HoriYdihed(theta,phi,RTpdotRpHH,RTpdotRpVV)
  call HoriXdihed(theta,phi,RTppdotRppHH,RTppdotRppVV)

* Equation 26 of Reference 1
  HHtot=Ss+St+SdHH+RTpdotRpHH+RTppdotRppHH
  VVtot=Ss+St+SdVV+RTpdotRpVV+RTppdotRppVV

* Converting to dBsm. Multiplied by 20 instead of 10 because
* complex square root RCS is used in the Reference 1 equations
* Calculating total response and single bounce, triple bounce,
* and double bounce responses
  rcsHH=20.*alog10(cabs(HHtot))
  rcsVV=20.*alog10(cabs(VVtot))
  rcsSB=20.*alog10(cabs(Ss))
  rcsTB=20.*alog10(cabs(St))
  rcsDBHH=20.*alog10(cabs(SdHH+RTpdotRpHH+RTppdotRppHH))
  rcsDBVV=20.*alog10(cabs(SdVV+RTpdotRpVV+RTppdotRppVV))

* RCS (dBsm) output to trihed.dat file. Three column data file
* listing the azimuth angle, HH, and VV, respectively
  write(8,20) float(i),rcsHH,rcsVV,rcsSB,rcsTB,rcsDBHH,
+      rcsDBVV
  10 continue
  20 format(F4.1,6(3x,F10.5))
  end

  Subroutine Singlebounce(theta,phi,Ss)
*****
* This subroutine calculates the single bounce scattering matrix
* elements for each trihedral plate. The HH and VV response is
* the same.
*****

* Variable Declaration
  real theta,phi,pi,sqrtpi,k,L,kL,sqrt2,phir,thetar,cosphi

  real sinphi,costheta,sintheta,sinc1,sinc2,sinc3
  complex Ss,jkLL,jkL,exp1,exp2,exp3,sigxy,sigyz,sigxz

* Common Variables
  common pi,sqrtpi,k,L,kL,j

```

```

* Variable Initialization
  jkLL=j*k*L**2
  jkL=j*k*L
  sqrt2=sqrt(2.)

* Converting to Radians
  phir=phi*pi/180.
  thetar=theta*pi/180.

  cosphi=cos(phir)
  sinphi=sin(phir)
  costheta=cos(thetar)
  sintheta=sin(thetar)

  sinc1=sin(kL*sintheta*cosphi)/(kL*sintheta*cosphi)
  sinc2=sin(kL*sintheta*sinphi)/(kL*sintheta*sinphi)
  sinc3=sin(kL*costheta)/(kL*costheta)

  exp1=cexp(sqrt2*jkL*sintheta*(sinphi+cosphi))
  exp2=cexp(sqrt2*jkL*(sintheta*sinphi+costheta))
  exp3=cexp(sqrt2*jkL*(sintheta*cosphi+costheta))

* Equations 28 -30 of Reference 1
  sigxy=-jkLL/sqrtpi*costheta*sinc1*sinc2*exp1
  sigyz=-jkLL/sqrtpi*sintheta*cosphi*sinc3*sinc2*exp2
  sigxz=-jkLL/sqrtpi*sintheta*sinphi*sinc3*sinc1*exp3

* Equation 31 of Reference 1
  Ss=sigxy+sigyz+sigxz

  return
  end

```

Subroutine Triplebounce(theta,phi,St)

\*\*\*\*\*

```

* This subroutine calculates the triple bounce scattering matrix
* elements for a square trihedral. The HH and VV response is
* the same.

```

\*\*\*\*\*

```

* Variable Declaration
  real theta,phi,pi,sqrtpi,k,L,kL,phir,thetar,cosphi,sinphi
  real costheta,sintheta,cospsix,cospsiy,cospsiz,al,an
  real temp1,temp2,temp3,am,area
  complex St,j

```

```

* Common Variables
  common pi,sqrtpi,k,L,kL,j

```

\* Converting to Radians and Variable Initialization

```
phir=phi*pi/180.  
thetar=theta*pi/180.
```

```
cosphi=cos(phir)  
sinphi=sin(phir)  
costheta=cos(thetar)  
sintheta=sin(thetar)
```

\* Equations 22-24 of Reference 1

```
cospsix=sintheta*cosphi  
cospsiy=sintheta*sinphi  
cospsiz=costheta
```

\* Equations 19-21 of Reference 1

```
al=min(cospsix,cospsiy,cospsiz)  
an=max(cospsix,cospsiy,cospsiz)
```

\* This calculates mid(cospsix,cospsiy,cospsiz)

```
temp1=max(cospsix,cospsiy)  
temp2=max(cospsiy,cospsiz)  
temp3=max(cospsix,cospsiz)  
am=min(temp1,temp2,temp3)
```

\* Equations 17 & 18 of Reference 1

```
if (am .le. an/2.) then  
  area=4.*al*am*L**2/an  
else  
  area=al*(4.-an/am)*L**2  
endif
```

\* Equations 45 & 46 of Reference 1

```
St=j*k*area/sqrtpi
```

```
return  
end
```

Subroutine Doublebounce(theta,phi,sigDBHH)

\*\*\*\*\*

\* This subroutine calculates the doublebounce scattering matrix

\* elements of the vertical dihedral. The HH and VV responses

\* are of opposite polarity.

\*\*\*\*\*

\* Variable Declaration

```
real theta,phi,pi,sqrtpi,k,L,kL,sqrt2,phir,thetar  
real cosphi,sinphi,costheta,sintheta,d,sinc1,sinc2  
complex sigDBHH,jkL,jkLL,jkd,exp1,exp2
```

```

* Common Variables
  common pi,sqrtpi,k,L,kL,j

* Variable Initialization
  jkL=j*k*L
  jkLL=j*k*L**2
  sqrt2=sqrt(2.)

* Converting to radians
  phir=phi*pi/180.
  thetar=theta*pi/180.

  cosphi=cos(phir)
  sinphi=sin(phir)
  costheta=cos(thetar)
  sintheta=sin(thetar)

* Equation 3 of Reference 2 (a=L)
  d=L*costheta/(sintheta*cosphi)
  jkd=j*k*d

  sinc1=sin(kL*costheta)/(kL*costheta)
  sinc2=sin(k*d*costheta)/(k*d*costheta)
  exp1=cexp(-jkL*costheta)
  exp2=cexp(-jkd*costheta)

* Equations 32 & 33 of Reference 1
  sigDBHH=jkLL/sqrtpi*sintheta*sinphi
  +      *(2.*sinc1+exp1/(jkL*costheta)*(1.-exp2*sinc2))

  return
  end

```

```

Subroutine HoriYdihed(theta,phi,RTpdotRpHH,RTpdotRpVV)
*****
* This subroutine calculates the elements of the double bounce
* scattering matrix elements for the horizontal dihedral with
* its crease along the y axis. The HH and VV elements are of
* opposite polarity. The Doublebounce subroutine provides the
* response due to the vertical dihedral.
*****

```

```

* Variable Declaration
  real theta,phi,pi,sqrtpi,k,L,kL,phir,thetar,cosphi,sinphi
  real costheta,sintheta,phip,thetap,denom,smHH,smVV
  complex RTpdotRpHH,RTpdotRpVV,j,sigDBHH

```

```

* Common Variables
  common pi,sqrtpi,k,L,kL,j

* Converting to Radians and Variable Initialization
  phir=phi*pi/180.
  thetar=theta*pi/180.

  cosphi=cos(phir)
  sinphi=sin(phir)
  costheta=cos(thetar)
  sintheta=sin(thetar)

  denom=1.-sintheta**2*sinphi**2

* Avoiding Singularity
  if (denom .eq. 0.) denom=.001

  smHH=cosphi**2-costheta**2*sinphi**2
  smVV=costheta**2*sinphi**2-cosphi**2

* Equations 48 & 46 of Reference 3 (converted to degrees)
  phip=asin(costheta/sqrt(denom))*180./pi
  thetap=asin(sqrt(denom))*180./pi

* Avoiding Singularities
  if (thetap .eq. 0.0) thetap=.001
  if (thetap .eq. 90.0) thetap=89.999
  if (phip .eq. 0.0) phip=.001
  if (phip .eq. 90.0) phip=89.999

  call Doublebounce(thetap,phip,sigDBHH)

* Equation 35 of Reference 1
  RTppdotRppHH=-sigDBHH/denom*smHH*cexp(j*kL*sintheta*sinphi)
  RTppdotRppVV=-sigDBHH/denom*smVV*cexp(j*kL*sintheta*sinphi)

  return
  end

```

```

Subroutine HoriXdihed(theta,phi,RTppdotRppHH,RTppdotRppVV)
*****
* This subroutine calculates the elements of the double bounce
* scattering matrix elements for the horizontal dihedral with
* its crease along the x axis. The HH and VV elements are of
* opposite polarity. The Doublebounce subroutine provides the
* response due to the vertical dihedral.
*****

```

```

* Variable Declaration
  real theta,phi,pi,sqrtpi,k,L,kL,phir,thetar,cosphi,sinphi
  real costheta,sintheta,phidp,thetadp,denom,smHH,smVV
  complex RTppdotRppHH,RTppdotRppVV,j,sigDBHH

* Common Variables
  common pi,sqrtpi,k,L,kL,j

* Converting to Radians and Variable Initialization
  phir=phi*pi/180.
  thetar=theta*pi/180.

  cosphi=cos(phir)
  sinphi=sin(phir)
  costheta=cos(thetar)
  sintheta=sin(thetar)

  denom=1.-sintheta**2*cosphi**2

* Avoiding Singularity
  if (denom .eq. 0.) denom=.001

  smHH=costheta**2*cosphi**2-sinphi**2
  smVV=sinphi**2-costheta**2*cosphi**2

* Equations 62 & 60 of Reference 3 (converted to degrees)
  phidp=asin(costheta/sqrt(denom))*180./pi
  thetadp=asin(sqrt(denom))*180./pi

* Avoiding Singularities
  if (thetadp .eq. 0.0) thetadp=.001
  if (thetadp .eq. 90.0) thetadp=89.999
  if (phidp .eq. 0.0) phidp=.001
  if (phidp .eq. 90.0) phidp=89.999

  call Doublebounce(thetadp,phidp,sigDBHH)

* Equation 36 of Reference 1
  RTppdotRppHH=sigDBHH/denom*smHH*cexp(j*kL*sintheta*cosphi)
  RTppdotRppVV=sigDBHH/denom*smVV*cexp(j*kL*sintheta*cosphi)

  return
  end

```



## E.7 Top Hat Source Code, Tophat.f

### Program tophat

```
*****
* This FORTRAN program (tophat.f) uses Physical Optics
* to calculate the polarization scattering matrix of
* a top hat with cylinder length L, cylinder radius a, and
* circular disk radius b. The cylinder axis coincides with the
* Y axis and the circular disk lies in the XZ plane. The HH and
* VV RCS components are output as a function of phi in dBsm from
* 0 to 90 degrees in one degree increments.

* This program is based on the equations in 'Physical Optics
* Polarization Scattering Matrix for a Top Hat Reflector',
* D.J. Blejer, IEEE Transactions on Antennas and Propagation,
* Vol. 39, No. 6, June 1991, pp 857-859. Hereinafter, referred
* to as Reference 1.

* Furthermore, this is a modified version of the Fortran code
* from Capt Dennis Young's thesis, 'Primitive Target Validation
* for the Signature Prediction Tools Software Package',
* AFIT/GE/ENG/90D-68

* Input File Required:
*   tophat.inp - this data file must contain one entry per
*               line in the following order:
*               a - dimension in cm of cylinder radius
*               b - dimension in cm of disk radius
*               L - dimension in cm of cylinder length
*               f - radar frequency in GHz
* Output File Produced:
*   tophat.dat - this is a three column data file listing
*               phi, HH (dBsm), and VV (dBsm), respectively
*****

* Variable Declaration
  real a,b,L,f,pi,sqrtpi,c,ka,kb,kL,phi,phir
  real cosphi,sinphi,cotphi,kasnphi,ka2snphi
  real kb2snphi,sinc1,J1kb2snphi,J1ka2snphi
  real Heff,rcsHH,rcsVV
  integer i
  complex j,exp1,exp2,Ecyl,Egp,Etop,Ems,SHH,SVV

* Open Input and Output Files
  open(7,file='tophat.inp',status='old')
  open(8,file='tophat.dat',status='unknown')
```

```

* Read in Parameters. Each parameter must include decimal point
  read(7,5) a,b,L,f
  5 format(F9.5/F9.5/F9.5/F5.1)
  a=a/100.
  b=b/100.
  L=L/100.
  f=f*10.**9

* Variable Initialization
  pi=3.1415926
  sqrtpi=sqrt(pi)
  c=2.998e8
  j=(0.0,1.0)
  k=2.*pi*f/c
  ka=k*a
  kb=k*b
  kL=k*L

* Begin Phi Loop
  do 100 i=0,90
    phi=float(i)

* Avoiding Singularities
    if (phi .eq. 0.0) phi=.001
    if (phi .eq. 90.) phi=89.999

* Converting to Radians and Continuing Variable Initialization
    phir=phi*pi/180.
    cosphi=cos(phir)
    sinphi=sin(phir)
    cotphi=cosphi/sinphi

    kasnphi=ka*sinphi
    ka2snphi=2.*kasnphi
    kb2snphi=2.*kb*sinphi

    exp1=cexp(j*ka2snphi+j*pi/4.)
    exp2=cexp(j*kL*cosphi)

    sinc1=sin(kL*cosphi)/(kL*cosphi)

    J1kb2snphi=bes1(kb2snphi,1)
    J1ka2snphi=bes1(ka2snphi,1)

* Equation 2 from Reference 1
  Heff=min(2.*L,2.*(b-a)*cotphi)

```

```

* Equation 3 from Reference 1
  Ectl=-sqrt(kasnphi)*exp1*L*sinc1*exp2
  Egp=-j*k*sqrtpi*cosphi*(2.*b**2*J1kb2snphi/
+   kb2snphi-2.*a**2*J1ka2snphi/ka2snphi)
  Etop=-j*k*sqrtpi*cosphi*(2.*a**2*J1ka2snphi/
+   ka2snphi)*exp2
  Ems=sqrt(kasnphi)*exp1*Heff

* Equation 1 from Reference 1
  SHH=Ectl+Egp+Etop+Ems
  SVV=Ectl+Egp+Etop-Ems

* Converting to dBsm. Multiplied by 20 instead of 10 because
* complex square root RCS was used to derive the equations.
  rcsHH=20.*alog10(cabs(SHH))
  rcsVV=20.*alog10(cabs(SVV))
  write(8,150) phi,rcsHH,rcsVV
100 continue
150 format(F4.1,5x,F14.6,5x,F14.6)
  end
  -

*****
* This function computes the Bessel function of the first
* kind and order zero or one. The input parameters are
*   x = argument of Bessel function
*   (real and greater than or equal to zero)
*   n = order of Bessel function
*   (zero or one)
*
* This coded function was given to the students in the AFIT
* EENG630 course during the Winter 91 term
*****

function bes1(x,n)
y=(x/3.0)**2
if(x.gt.3.0) go to 1
if(n.eq.0) then
  a0=1.0
  a1=-2.2499997
  a2=1.2656208
  a3=-0.3163866
  a4=0.0444479
  a5=-0.0039444
  a6=0.0002100
else
  a0=0.5
  a1=-0.56249985
  a2=0.21093573
  a3=-0.03954289

```

```

    a4=0.00443319
    a5=-0.00031761
    a6=0.00001109
endif
bes1=a0+y*(a1+y*(a2+y*(a3+y*(a4+y*(a5+y*a6))))
if(n.eq.1) bes1=x*bes1
return
1  y=3.0/x
   sx=sqrt(x)
   if(n.eq.0) then
     a1=-0.00000077
     a2=-0.00552740
     a3=-0.00009512
     a4=0.00137237
     a5=-0.00072805
     a6=0.00014476
     b0=-0.78539816
     b1=-0.04166397
     b2=-0.00003954
     b3=0.00262573
     b4=-0.00054125
     b5=-0.00029333
     b6=0.00013558
   else
     a1=0.00000156
     a2=0.01659667
     a3=0.00017105
     a4=-0.00249511
     a5=0.00113653
     a6=-0.00020033
     b0=-2.35619449
     b1=0.12499612
     b2=0.00005650
     b3=-0.00637879
     b4=0.00074348
     b5=0.00079824
     b6=-0.00029166
   endif
f=0.79788456+y*(a1+y*(a2+y*(a3+y*(a4+y*(a5+y*a6))))
thta=x+b0+y*(b1+y*(b2+y*(b3+y*(b4+y*(b5+y*b6))))
bes1=f*cos(thta)/sx
return
end

```

## ***Appendix F: Data Acquisition Programs***

Section F.1 lists the data transfer program used to transfer the AFIT anechoic chamber data from the Hewlett Packard 9000 Series 236 computer system to an IBM compatible computer. Section F.2 lists the FORTRAN source code for shifting the AFIT data in azimuth to a user specified azimuth location.

### ***F.1 Data Transfer***

```
10  ! TRANSFER VERSION 1.0 AUG 1991
20  ! LISTS, PRINTS, OR TRANSFERS DATA FILES
21  ! RUN "ARMS" first by loading and running "AUTOPROG", and
22  ! then escape back to basic (K7)
24  ! Load and run TRANSFER located on the BDATLIST disk
30  OPTION BASE 1
40  MASS STORAGE I ":INTERNAL,4,0"
50  Azimf$=""
51  PRINT "THIS PROGRAM WILL LIST, PRINT, OR TRANSFER AN
AZIMUTH DATA FILE WITH 360 DATA POINTS. THE DATA WILL NOT BE
SHIFTED."
52  PRINT ""
53  PRINT "ENSURE THE DATA DISK IS INSTALLED."
54  PRINT ""
60      INPUT "TYPE IN THE NAME OF THE AZIMUTH FILE",Azimf$
61      DIM E data(360)
62      ASSIGN @dT to Azimf$
63      ENTER @Dt;E_data(*)
64      ASSIGN @Dt TO *
100  Pt$=""
102  PRINT "TYPE IN AN UPPER CASE C FOR A CRT LISTING, AN UPPER
CASE P FOR A PRINTER LISTING OR AN UPPER CASE T FOR A FILE
TRANSFER."
110  INPUT "TYPE IN AN UPPER CASE C, P, OR T",Pt$
120  IF Pt$="C" THEN
121      PRINTER IS CRT
136          PRINT "LISTING OF DATA FROM DATA FILE",Azimf$
137          PRINT E_data(*)
141  END IF
142  IF Pt$="P" THEN
150      PRINTER IS 701
151      PRINT ""
152      PRINT ""
```

```

165             PRINT "LISTING OF DATA FROM DATA FILE",Azimf$
166             PRINT E_data(*)
170         END IF
171         IF Pt$="T" THEN
181             PRINT ""
182             PRINT " The BAUD rate is set at 4800."
183             PRINT ""
184             PRINT "1) Insure that the RS-232 cable is connected."
185             PRINT "2) On the Z-248, enter the SMARTCOM directory"
186             PRINT "3) Execute SCOM"
187             PRINT "4) Choose 1 BEGIN COMMUNICATION"
188             PRINT "5) Choose A<NSWER"
189             PRINT "6) Choose C"
190             PRINT "7) Press F1"
191             PRINT "8) Choose 4 RECEIVE FILE"
192             PRINT "9) Press Enter"
193             PRINT "10) Enter File Name"
194             PRINT ""
196             PRINT "PRESS" CHR$(129);"CONTINUE";CHR$(128);"WHEN THE Z-248
IS READY TO RECEIVE DATA"
197             PAUSE
198             PRINT "TRANSFERRING DATA"
199             ASSIGN @Rs232 TO 9;FORMAT ON
200             CONTROL 9,3;4800
201             CONTROL 9,4;3
204                 FOR I=1 TO 360
205                     OUTPUT 9;E_data(I)
206                 next I
213             END IF
215             PRINT "FINISHED"
216             PRINT ""
218             PRINTER IS CRT
219             PRINT "DO YOU WANT TO RUN THIS PROGRAM AGAIN?"
220             INPUT "TYPE IN AN UPPER CASE Y FOR YES OR N FOR NO",Repeat$
221             IF Repeat$="Y" THEN
222                 GOTO 60
223             ELSE
224                 GOTO 226
225             END IF
226             END

```

## ***F.2. Data Shift Source Code, Shiftpk.f***

```
PROGRAM SHIFTPK
*****
* This FORTRAN program reads in a 360 point data file (the RCS
* values for a 360 point azimuth scan) and shifts the peak to
* a user specified azimuth location.
*
* Input files Required:
*   User specified data file name (unit 7) - must be
*   a 360 data point array.
* Output files Produced:
*   Input file name preceded with an s (unit 8) -
*   a 360 column x 2 row array. Column one sequentially
*   lists 0 to 359 (corresponding to azimuth). Column
*   two is the shifted input file.
*****

* Variable Declaration
  real RCS_array(360,2),RCS_sorted(360,2),maximum
  integer i,num_points,azim_max,azim_desired,shift
  character*25 file_name
  num_points=360

* Keyboard Inputs
  print *, '*****'
  print *, '* This program enables the user to read in a *'
  print *, '* 360 point azimuth scan and shift the data *'
  print *, '* so the maximum occurs at a user specified *'
  print *, '* azimuth location. *'
  print *, '*****'
  print *, ''
  print *, 'Enter the data file name'
  read *, file_name
  print *, ''
  print *, 'Enter the azimuth position at which the maximum'
  print *, 'RCS value should be located'
  read *, azim_desired

* Opening input and output files
  open(7,file=file_name,status='old')
  open(8,file='s'//file_name,status='unknown')

* Reading in the data file and sequentially numbering the points
* from 0 to the number of data points being read in minus one
  do 10 i=0,num_points-1
    read(7,*) RCS_array(i,2)
    RCS_array(i,1)=i
  10 continue
```

```

* Finding the azimuth location where the maximum RCS value
* is located (-200 is an arbitrarily chosen small value)
  maximum=-200.
  do 20 i=0,num_points-1
    if(maximum .lt. RCS_array(i,2)) then
      azimuth_max=RCS_array(i,1)
      maximum=RCS_array(i,2)
    endif
  20 continue

* Determining how many cells the data should be shifted
  shift=azimuth_desired-azimuth_max
  if (shift .lt. 0) then
    shift=num_points+shift
  endif

* Shifting the data
  do 30 i=0,num_points-1
    if (i+shift .lt. num_points) then
      RCS_sorted(shift+i,2)=RCS_array(i,2)
      RCS_sorted(shift+i,1)=RCS_array(i,1)
    else
      RCS_sorted(i-(num_points-shift),2)=RCS_array(i,2)
      RCS_sorted(i-(num_points-shift),1)=RCS_array(i,1)
    endif
  30 continue

* Writing out the output file
  write(8,*) 'AZIMUTH   RCS(dBsm)'
  do 40 i=0,num_points-1
    write(8,50) i,RCS_sorted(i,2)
  40 continue
  50 format(I3,5x,F10.6)

  print *,''
  print *,'The data has been shifted and is in a file'
  print *,'called ',s//file_name
  end

```



## ***Bibliography***

1. Anderson, W. C. "Consequences of Nonorthogonality on the Scattering Properties of Dihedral Reflectors," *IEEE Transactions on Antennas and Propagation*, AP-35:1154-1159 (October 1987).
2. Ando, Makoto and Teruhiro Kinoshita. "PO and PTD Analyses in Polarization Prediction for Plane Wave Diffraction from a Large Circular Disk," *89 IEEE AP/S Symposium*. 1282-1285. San Jose CA, June 1989.
3. Arnold, Charles L. Jr. *SRIM User's Manual, Release Version 2.2*. US Army Intelligence and Threat Analysis Center Contract MDA 903-86-C-0286 with Environmental Research Institute of Michigan. Bldg 203, Stop 314, Washington Navy Yard, Washington D.C., September 1987.
4. Arnold, Charles L. Jr. and Mervyn Walsh. *SRIM User Reference Manual, Release 4.0*. Wright Research and Development Center Contract F33615-88-C-1749 with Environmental Research Institute of Michigan. Wright-Patterson AFB OH, January 1991.
5. Avionics Laboratory, Wright Research and Development Center, Air Force Systems Command. "Synthetic Aperture Radar Imaging Technology Program, Volume 1, Task 6, Foliage Penetration Radar Study," Contract F33615-88-C-1716 with Environmental Research Institute of Michigan. Wright-Patterson AFB OH, July 1990.
6. Balanis, Constantine A. *Advanced Engineering Electromagnetics*. New York: John Wiley & Sons, 1989.
7. Bechtel, M. E. "Application of Geometric Diffraction Theory to Scattering from Cones and Disks," *Proceedings of the IEEE*, 53:877-882 (August 1985).
8. Blejer, Dennis J. *Physical Optics Polarization Scattering Matrix and Impulse Response of a Right Dihedral Corner Reflector*. Unpublished report No. 47PM-ADT-0052. Massachusetts Institute of Technology (MIT), Lincoln Laboratory, Lexington MA, 22 December 1986.
9. Blejer, Dennis J. *Scattering Matrix Transformations for Polarimetric Target Modeling*. Unpublished report No. 47PM-ADT-0057. MIT, Lincoln Laboratory, Lexington MA, 20 May 1987.
10. Blejer, Dennis J. *Physical Optics Polarization Scattering Matrix of a Square Trihedral Corner Reflector*. Unpublished report No. 47PM-ADT-0059. MIT, Lincoln Laboratory, Lexington MA, 30 June 1987.

11. Blejer, Dennis J. "Physical Optics Polarization Scattering Matrix for a Top Hat Reflector," *IEEE Transactions on Antennas and Propagation*, 39: 857-859 (June 1991).
12. Choi, J. and others. "Near-axial Backscattering from a Cone-Sphere," *Radio Science*, 25:427-434 (July-August 1990).
13. Crispin, J. W. Jr. and K. M. Siegel. *Methods of Radar Cross-Section Analysis*. New York: Academic Press, 1968.
14. Faison, Joseph C. "The Electromagnetic Code Consortium," *IEEE Antennas and Propagation Society Magazine*, 19-23 (February 1990).
15. Herrick D. F. and I. J. LaHaie. *SRIM Polarimetric Signature Modeling*, Internal Research and Development Final Report, October 1987-September 1988. Environmental Research Institute of Michigan Report No. 675805-1-F, 13 December 1988.
16. Knott, Eugene F. "RCS Reduction of Dihedral Corners," *IEEE Transactions on Antennas and Propagation*, AP-25: 406-409 (May 1977).
17. Knott, Eugene F. and others. *Radar Cross Section*. Norwood MA: Artech House, Inc., 1985.
18. Knott, Eugene F. "A Progression of High-Frequency RCS Prediction Techniques," *Proceedings of the IEEE*, 73:252-264 (February 1985).
19. Lee, S. W. and others. "Twelve Versions of Physical Optics: How do they Compare?" *88 IEEE AP/S Symposium*. 408-411. Syracuse NY, June 1988.
20. Marsland, Diane P. "Higher Order Diffractions from a Circular Disk," *IEEE Transactions on Antennas and Propagation*, AP-35:1436-1443 (December 1987).
21. Mensa, Dean. *High Resolution Radar Cross Section Imaging*. Four tutorial VCR tapes, not dated.
22. Miller, Edmund K. "A Selective Survey of Computational Electromagnetics," *IEEE Transactions on Antennas and Propagation*, 36:1281-1305 (September 1988).
23. Peterson, Andrew F., Professor of Electrical Engineering. Georgia Institute of Technology, Atlanta GA, August 91.
24. Ross, R. A. "Radar Cross Section of Rectangular Flat Plates as a Function of Aspect Angle," *IEEE Transactions on Antennas and Propagation*, AP-14:329-335 (May 1966).

25. Ruck, George T. *Radar Cross Section Handbook*, Volume 2. New York: Plenum Press, 1970.
26. Sargent, Robert G. "A Tutorial on Validation and Verification of Simulation Models," *Proceedings of the 1988 Winter Simulation Conference*. 33-38. San Diego, December 1988.
27. Tuley, M. T. *Radar Cross Section Model Evaluation, Draft Final Report*. Contract DAAH01-89-D-A003, Georgia Technology Research Institute Project A-8638. Georgia Institute of Technology, Atlanta GA, March 1991.
28. Young, Capt Dennis John. *Primitive Target Validation for the Signature Prediction Tools Software Package*. MS thesis, AFIT/GE/ENG/90D-68. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1990.