PLANE-WAVE SCATTERING OF A PERIODIC CORRUGATED CYLINDER

by

Samuel Garcia

A Dissertation Submitted to the Faculty of

College of Engineering & Computer Science

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

Florida Atlantic University

Boca Raton, FL

May 2017

PLANE-WAVE SCATTERING OF A PERIODIC CORRUGATED CYLINDER

by

Samuel Garcia

This dissertation was prepared under the direction of the candidate's dissertation advisor. Dr. Jonathan Bagby, Department of Computer & Electrical Engineering and Computer Science, and has been approved by the members of his supervisory committee. It was submitted to the faculty of the College of Engineering & Computer Science and was accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

SUPERVISORY COMMITTEE:

Jonathan Bagby, Ph D.
Dissertation Advisor

Vichate Ungvichian, Ph D.

Valentine Aalo, Ph D.

William Rhodes, Ph D.

Grigoriy Kreymerman, Ph D.

Nurgun Erdol, Ph.D.
Chair, Department of Computer & Electrical Engineering & Computer Science

Mohammad Ilyas, Ph.D.
Dean, College of Engineering & Computer Science

Deborah L. Floyd, Ed.D.
Dean, Graduate College

April 12, 2017
Date

ACKNOWLEDGEMENTS

This dissertation could not have been accomplished without the supervision and guidance of Dr. Jonathan Bagby, the Committee Chairperson and dissertation advisor. I would like to thank Dr. Bagby for his continued support throughout my PhD. His vast knowledge on this research material is something I admire. My sincere gratitude goes out to him for the guidance and advice he has given me.

I wish to thank the Committee Members, Dr. Valentine Aalo, Dr. Grigoriy Kreymerman, Dr. William Rhodes and Dr. Vichate Ungvichian for their time serving as members of the committee and providing their educated assessment to this dissertation. I would like to give an extra thanks Dr. Vichate Ungvichian, who stepped in to advise me for a semester. I owe him a debt of gratitude for that time and for his guidance and support.

I want to give a special thanks to my girlfriend, Ivette Morazzani, who’s pushed me and advised me along this journey. She gave me constant encouragement and has shown an enormous amount of patience with me. She has also directly contributed to my success in helping me develop and learn the coding language used, along with running and reviewing countless simulations, lines of code, formulas and this dissertation itself.

I’m also greatly thankful for my family in supporting me through this journey. Their strength and support in me encouraged me to push through and achieve this goal of completing my dissertation.

ABSTRACT

Author: Samuel Garcia

Thesis: Plane-Wave Scattering of a Periodic Corrugated Cylinder

Institution: Florida Atlantic University

Dissertation Advisor: Dr. Jonathan Bagby

Degree: Doctorate of Philosophy

Year: 2016

In this dissertation, a novel approach to modeling the scattered field of a periodic corrugated cylinder, from an oblique incident planewave, is presented. The approach utilizes radial waveguide approximations for fields within the corrugations, which are point matched to approximated scattered fields outside of the corrugation to solve for the expansion coefficients. The point matching is done with TM_z and TE_z modes simultaneously, allowing for hybrid modes to exist.

The derivation of the fields and boundary conditions used are discussed in detail. Axial and radial propagating modes for the scattered fields are derived and discussed. Close treatment is given to field equations summation truncation and conversion to matrix form, for numerical computing. A detailed account of the modeling approach using Mathematica® and NCAlgebra for the noncommutative algebra, involved in solving for the expansion coefficients, are also given.

The modeling techniques offered provide a full description and prediction of the scattered field of a periodic corrugated cylinder. The model is configured to approximate a smooth cylinder, which is then compared against that of a textbook standard smooth cylinder. The methodology and analysis applied in this research provide a solution for computational electromagnetics, RF communications, Radar systems and the like, for the design, development, and analysis of such systems. Through the rapid modeling techniques developed in this research, early knowledge discovery can be made allowing for better more effective decision making to be made early in the design and investigation process of an RF project.

DEDICATION

First and foremost, this manuscript is dedicated to my late father. It's because of him I had any interest in becoming an electrical engineering. He always encouraged me to study and work hard. Growing up, he always gave me electrical projects to work on and challenge me. He started this PhD journey with me, but now, wherever he is, I know he'd be proud.

I also dedicated this to my girlfriend, Ivette Morazzani, who not only spent many years putting up with my research, she's also spent a lot of time reviewing my work and helping me code my work.

I would also like to dedicate this to my mom Mayte, my brother Carlos, my sister-in-law Paula and my nephews Carlos and Bryan. They have all supported me through this journey and have brought me joy in my toughest moments.

PLANE-WAVE SCATTERING OF A PERIODIC CORRUGATED CYLINDER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 Objective

The scattered field due to a large periodic corrugated cylinder of quasi-infinite length is formulated and numerically determined in this dissertation. It is assumed that the boundaries and structure of the corrugated cylinder are constructed of Perfect Electrical Conductors (PECs). The present research provides an alternate technique for evaluating near-field scattering of a corrugated cylinder due to an incident plane wave while improving the computational efficiency of infinite array scattering using Floquet modes. The effect on the total electric field from variations of relative dimensions of the corrugated cylinder is also investigated.

Problems of scattering from periodic corrugated cylinders have been treated in the past. There has been investigation into the use of radial waveguide representation of the corrugated region [1] [2], asymptotic corrugation boundary conditions [3] [4], metallic ring representation [5], tensor permeability and tensor permittivity [6], and surface roughness function for corrugation representation [7]. However, treatment of this problem utilizing radial waveguide representation for the region within the corrugations and simultaneously working with Transverse Magnetic (TM) and Transverse Electric (TE) modes appears to be a new contribution to the field of computational electromagnetics, specifically with regards to scattering structures.

1.2 Background

In today's modern era of increasingly use of high-tech radio frequency (RF) communications and radar systems, it has become increasingly important to understand and mitigate the effects of RF or electromagnetic field scattering from common objects and geometries involved in those systems. That's because the more bandwidth and the greater sensitivity these instruments require, the more impactful inadvertent scattering off of nearby objects can be. The effect would cause an increase to the noise floor and distortion of intended signals, which reduces the overall quality and bandwidth of the data sent through the system. In the case of a 'stealth' type aircraft, unintended scattering would increase the aircraft's Radar Cross Section (RCS) which would improve an adversarial radar system's ability to detect that aircraft.

A common shape seen across all these systems is the cylinder. The fuselage of an aircraft and many external payloads are approximately cylindrical. The supporting struts for the transmitter/receiver on parabolic dishes (Figure 1-1), as such used in satellite communication, radio astronomy, etc. also tend to be cylindrical in shape. The present research provides a method of analysis for a modified geometry of these cylindrical structures, utilizing a periodic corrugation, in effect to optimize the design of them through assessment of their scattering.

Figure 1-1 Example of a parabolic dish used in communications

1.3 Chapter-Wise Organization

In order to provide a coherent and orderly research documentation, this dissertation has been organized categorically into chapters, which are summarized here:

- Chapter 1 Introduction

  This chapter (current) is an introduction to the present research. It provides context and motivation for the field of research. It also provides a summary for the structure of the dissertation.

- Chapter 2 Periodic Corrugated Cylinder: Physical Description And Electromagnetic Application

  This chapter describes the problem space in which the research is set. A detailed description of the geometry and electromagnetic treatment are discussed. The characteristics of the problem space that are varied are also described. Floquet modes are invoked to represent the phase shift of the fields due to the periodicity of the structure. The nomenclature for the components, regions and other aspects of the problem space are established. Also, well established electromagnetic waveguide equations are represented and discussed. The overall approach for achieving solutions is also discussed as well as its validity.

- Chapter 3 Incident Field

  The incident plane-wave for TM & TE modes is described. The fields are further derived into their respective cylindrical coordinate components for both E-fields and H-fields.

- Chapter 4 Region I – Radial Waveguide Field Equations

  The fields between the corrugations are examined. Radial waveguide field equations were derived, based on Maxwell's equations. Boundary conditions are established as part of the derivation and further reduction of the equations. Finally, equations are developed for each respective cylindrical coordinate component for both E-fields and H-fields. Unknown coefficients are identified for solving in subsequent chapters.

- Chapter 5 Region II – Scattered Field Equations

  The scattered fields are derived and boundary conditions are established. Equations are developed for each respective cylindrical coordinate component for both E-fields and H-fields. Unknown coefficients are identified for solving in the subsequent chapter.

- Chapter 6 Boundary Conditions & Point Matching Method

  A point matching method is established for solving all the unknown coefficients. Equation sets are established for boundary conditions intended for point matching. All the field equations' summations are truncated from $+\infty$ and $-\infty$, to integers.

- Chapter 7 Results, Comparisons and Future Research

  Results are presented and discussed. General description of mathematical software tool and use are provided. Expressions identified for full description of solution space. A description of future research is also discussed.

CHAPTER 2 PERIODIC CORRUGATED CYLINDER: PHYSICAL DESCRIPTION AND ELECTROMAGNETIC APPLICATION

2.1 Physical Description

The subject of this paper is on the scatterer shown in Figure 2-1. It is a periodic corrugated cylinder of approximately infinite length, discussed further in the following section. The different dimensions of its components are referenced in Figure 2-1(b) with letter reference designators.

The corrugated cylinder is lined vertically with its center axis being on the z-axis. The inner radius of the corrugation is ρ_1 and the outer radius is ρ_2 along the ρ axis. The corrugated cylinder is symmetrical all along the φ axis. These corrugation pieces, also referred to here as discs, have a thickness (height in z direction) of b. The spacing between the corrugation (or discs) have a dimension of a. The values of these dimensions are discussed further in later chapters and will be described in terms of λ, wavelength of the incident plane-wave.

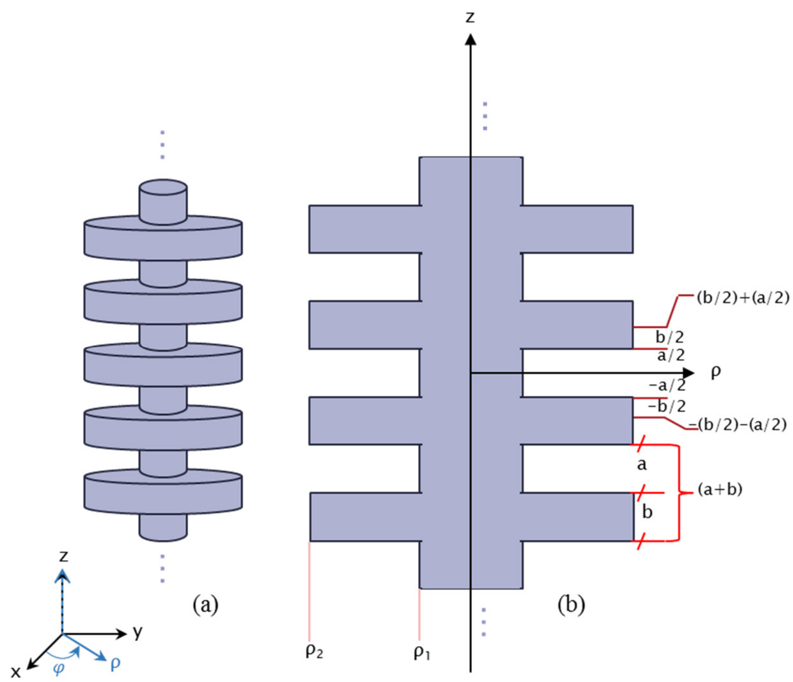


Figure 2-1 Representation of a Segment of a Periodic Corrugated Cylinder: (a) 3D view of the Periodic Corrugated Cylinder (b) Cross-sectional view of the Periodic Corrugated Cylinder with Referenced Dimensions

2.2 Infinite Length Approximation

When a periodic structure of infinite length, in this case a corrugated cylinder, is radiated by an incident field, the scattered field produced will contain multiple modes or space harmonics, which are coupled to the boundary conditions in which they must satisfy [8, p. 625]. These spatially periodic fields can be represented through Floquet modes. Floquet modes are modes of propagating waves that take on the symmetry of the periodic structure that wave has interfaced with. This is based on Floquet theory, in which a single period of the periodic structure is used to define the wave and accounting for the phase shift along the axis of propagation, which is also the axis of periodicity of the structure. A further description of Floquet modes and Floquet theory can be found in [9, pp. 264-266] and [8, pp. 605-608].

Practically, one works with smaller and finite length periodic structures and not infinitely long ones. However, sufficiently long periodic scatterers can be approximated as infinitely long allowing for simplification for mathematical models of the scattering fields [10], [11].

2.3 Problem Space

Consider a PEC periodic corrugated cylinder of infinite length as shown in Figure 2-2. Now consider that same corrugated cylinder in the presence of an incident planewave. That corrugated cylinder will perturb the incident planewave by behaving as a scatterer. Many have approached the problem of calculating the scattered field, such as Manara [1] [2], Kishk [4], Freni [5] and Hillion [7].

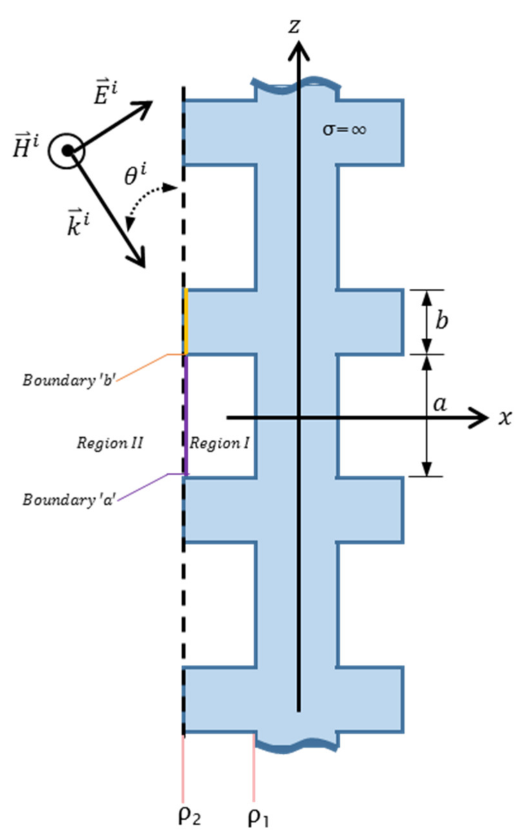


Figure 2-2 PEC periodic corrugated cylinder of infinite length, radiated by incident planewave

However, the method proposed in this paper describes the simultaneous use of the Transverse Magnetic (TM_z) and the Transverse Electric (TE_z) field modes, with respect to the z-axis, to solve the problem of predicting the scattered fields. As implied by Constantine A. Balanis in his work on describing scattering by a conducting circular cylinder of an oblique planewave, a cylinder structure that deviates from a smooth cylinder can experience depolarizations of the fields due to the scattering [12, p. 615]. Based on this statement, this work makes the assumption of a hybrid mode when scattering from a corrugated cylinder. Therefore, the TM_z and the TE_z modes shall be accounted for simultaneously and depolarization of the fields from one mode to another can occur.

Along with TM_z and the TE_z modes, this approach also utilizes a radial waveguide representation of the fields within the corrugations (region I) which the full solution for is formulated in conjunction with the fields outside of the corrugations (region II). The radial waveguide method has been investigated by Manara [1]. The combination of TM_z and the TE_z modes with the radial waveguide representation provides a more complete description of the scattered field from a periodic corrugated cylinder, that is novel and has not been observed by the author in previous literature.

2.4 Solution Approach

Having been presented with a description of the problem space, the reader can now follow along with the solution approach, a step-by-step guide on how to utilize the problem space components provided in order to solve the problem (predicting the scattered fields) of a known PEC periodic corrugated cylinder.

Step 1: Identify the incident planewaves

Step 2: Derive solutions to Helmholtz equations for the fields in region I

Step 3: Derive solutions to Helmholtz equations for the scattered fields in region II

Step 4: Truncate all summation equations to finite ranges and represent in matrix form

Step 5: Apply boundary conditions for fields in region I and region II to solve for unknown coefficients

Step 6: Symbolically and numerically compute solutions for unknown coefficients

This is the solution approach that is represented in this paper. Steps 1, 2 and 3 are described in chapters 3, 4 and 5 respectively. Steps 4, 5 and 6 are covered in chapter 6. Step 6 is where a great deal of time and effort was spent by the author in developing the appropriate numerical computing approach.

In this paper, the fields and incident planewaves are considered to be time harmonic, as shown in equations (2-1) and (2-2). Only the vector phasor component, $\vec{E}(x,y,z)$, will be considered and the $e^{j\omega t}$ will be omitted for simplification.

$$\vec{E}(x,y,z,t) = Re\left[\vec{E}(x,y,z)\right]e^{j\omega t} \tag{2-1}$$

$$\vec{H}(x,y,z,t) = Re\left[\vec{H}(x,y,z)\right]e^{j\omega t}. \tag{2-2}$$

The field equations and solutions to them, presented in this paper will be in the cylindrical coordinate form (i.e. $\hat{\rho}$, $\hat{\varphi}$, and $\hat{z}$ axes), with the exception of the introduction of the incident field in Chapter 3.

CHAPTER 3 INCIDENT FIELD

Now consider an unbounded medium, referred to here as region II, which is a near infinite media and void of any scatterers. This description allows for the easy modeling of a propagating incident planewave. An idealistic view as such is incomplete, with regards to the scattered field, in the presence of a scatterer (as in the case of the periodic corrugated cylinder). In order to appropriately describe the total fields (E^t, H^t), superposition can be used, as shown in equations (3.1) and (3.2), in which the incident field (E^i, H^i) is added to the scattered field (E^s & H^s, field created by a scatterer in the presence of a propagating field) in order to get the total field. Deriving the scattered field is the subject of chapter 5. This chapter will focus on the incident field.

$$E^t = E^i + E^s \qquad (3\text{-}1)$$

$$H^t = H^i + H^s. \qquad (3\text{-}2)$$

The problem space and solution approach require that the solution set be driven by two sets of modes, the TM and TE modes. Separate treatment will be given to these modes when describing the field components. However, when solving for boundary conditions at the point matching phase, these modes will be combined through invoking superposition. For now, the driving incident field will be separated in its TM and TE modes. It's also important to note here that the present work is done on oblique incident planewaves and not normal incident planewaves.

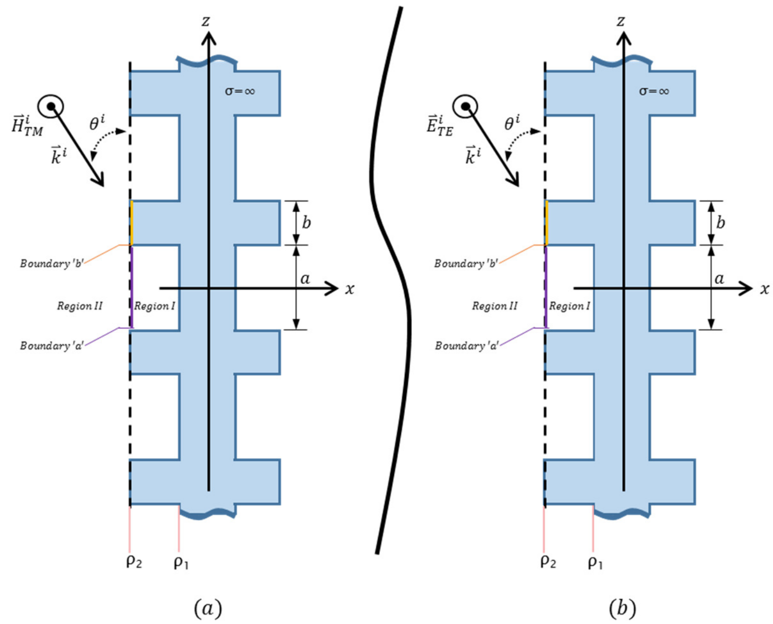


Figure 3-1 Incident field shown with respect to corrugated cylinder for TM_z mode (a) and TE_z mode (b)

Depicted in Figure 3-1(a) is a TM_z mode planewave, in region II, which is incident to a periodic corrugated cylinder. In Figure 3-1(b), the same structure is shown only now with a TE_z mode planewave that's incident. In both scenarios, the planewave is traveling in the same direction, as indicated by θ^i (angle of incidence) and the wave vector $\vec{k}^i$, which is given by

$$\vec{k}^i = k[\hat{x}\, sin\, \theta^i - \hat{z}\, cos\, \theta^i] \qquad (3\text{-}3)$$

where $k = \omega\sqrt{\mu\varepsilon}$ is the wavenumber. In region II, as is the case for the incident field, $k = \omega\sqrt{\mu_{II}\varepsilon_{II}}$ where $\mu = \mu_{II}$ and $\varepsilon = \varepsilon_{II}$. In region I, covered in detail in Chapter 4, $k = \omega\sqrt{\mu_I\varepsilon_I}$ where $\mu = \mu_I$ and $\varepsilon = \varepsilon_I$.

Here is a brief description of TM_z and TE_z modes of a propagating wave, to aid the reader through this paper. A transverse magnetic mode, with respect to the z-axis (TM_z), of a propagating wave, is that of a wave that has its magnetic field components in a plane that is perpendicular (transverse) to the z-axis. Therefore, there is no magnetic field component on the z-axis, or $H_z=0$, for a TM_z mode.

A transverse electric mode, with respect to the z-axis (TE_z), of a propagating wave, follows the same logic as that of the transverse magnetic, but rather now it's the electric field component that is perpendicular (transverse) to the z-axis. In this case of a TE_z mode, there is no electric field component on the z-axis, or $E_z=0$.

3.1 Incident Field TM_z mode

In the case of the TM_z incidence, the equation for the electric field is given by equation (3.4) for rectangular coordinates and (3.5) for cylindrical coordinates. The component $Z = \sqrt{\frac{\mu_{II}}{\varepsilon_{II}}}$ is intrinsic impedance of region II. Conversion of a vector field between coordinate system types can be found in [12, pp. 920-923] .

$$\vec{E}^i = E_0[\hat{x}\cos\theta^i + \hat{z}\sin\theta^i]e^{-jk[x\sin\theta^i - z\cos\theta^i]} \quad (3\text{-}4)$$

$$\vec{E}^i = E_0[\hat{\rho}\cos\varphi\cos\theta^i - \hat{\varphi}\sin\varphi\cos\theta^i + \hat{z}\sin\theta^i]e^{jkz\cos\theta^i}\sum_{n=-\infty}^{\infty} j^{-n}J_n(k\rho\sin\theta^i)e^{jn\varphi} \quad (3\text{-}5)$$

The corresponding magnetic fields are given using and expanding on Maxwell's curl equations, which a full example can be found in [12, pp. 616-618] to get equations (3-6) for rectangular coordinates and (3-7) for cylindrical coordinates.

$$\begin{aligned}\vec{H}^i &= \frac{1}{Z}\,\hat{k}\times\vec{E}^i = \frac{E_0}{Z}[\hat{x}\sin\theta^i - \hat{z}\cos\theta^i]\times[\hat{x}\cos\theta^i + \hat{z}\sin\theta^i]e^{-jk[x\sin\theta^i - z\cos\theta^i]} \\ &= -\hat{y}\frac{E_0}{Z}e^{jkz\cos\theta^i}\sum_{n=-\infty}^{\infty} j^{-n}J_n(k\rho\sin\theta^i)e^{jn\varphi}\end{aligned} \quad (3\text{-}6)$$

$$\vec{H}^i = \frac{E_0}{Z}[-\hat{\rho}\sin\varphi - \hat{\varphi}\cos\varphi]e^{jkz\cos\theta^i}\sum_{n=-\infty}^{\infty} j^{-n}J_n(k\rho\sin\theta^i)e^{jn\varphi} \quad (3\text{-}7)$$

Further breaking down the electric and magnetic field equations into their coordinate constituents, we get equations

$$E^i_{z_{II}} = E_0 \sin\theta^i\, e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho\sin\theta^i) e^{jn\varphi} \tag{3-8}$$

$$E^i_{\rho\_TM^{II}} = E_0 \cos\varphi \cos\theta^i\, e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho\sin\theta^i) e^{jn\varphi} \tag{3-9}$$

$$E^i_{\varphi\_TM^{II}} = -E_0 \sin\varphi \cos\theta^i\, e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho\sin\theta^i) e^{jn\varphi} \tag{3-10}$$

$$H^i_{\rho_{TM^{II}}} = -\frac{E_0}{Z} \sin\varphi\, e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho\sin\theta^i) e^{jn\varphi} \tag{3-11}$$

$$H^i_{\varphi\_TM^{II}} = -\frac{E_0}{Z} \cos\varphi\, e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho\sin\theta^i) e^{jn\varphi} \tag{3-12}$$

which now provides a more manageable description of the TM_z mode incident field as will be seen in the later chapters.

3.2 Incident Field TE_z mode

In the case of the TE_z incidence, the equation for the magnetic field is given by equation (3-13) for rectangular coordinates and (3-14) for cylindrical coordinates.

$$\vec{H}^i = H_0[\hat{x}\cos\theta^i + \hat{z}\sin\theta^i] e^{-jk[x\sin\theta^i - z\cos\theta^i]} \tag{3-13}$$

$$\vec{H}^i = H_0[\hat{\rho}\cos\varphi\cos\theta^i - \hat{\varphi}\sin\varphi\cos\theta^i + \hat{z}\sin\theta^i] e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho\sin\theta^i) e^{jn\varphi}. \tag{3-14}$$

The corresponding electric fields are given by (3-15) for rectangular coordinates and (3-16) for cylindrical coordinates.

$$\vec{E}^i = Z\vec{H}^i \times \hat{k} = ZH_0[\hat{x}\cos\theta^i + \hat{z}\sin\theta^i] \times [\hat{x}\sin\theta^i - \hat{z}\cos\theta^i] e^{-jk[x\sin\theta^i - z\cos\theta^i]} = \hat{y} Z H_0 e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho\sin\theta^i) e^{jn\varphi} \tag{3-15}$$

$$\vec{E}^i = ZH_0[\hat{\rho}\sin\varphi + \hat{\varphi}\cos\varphi] e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho\sin\theta^i) e^{jn\varphi} \tag{3-16}$$

As was done in the incident Field TM_z mode section, a further breakdown of the TE_z mode field equations into their coordinate constituents gives

$$H^i_{z^{II}} = H_0 \sin\theta^i\, e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \quad (3\text{-}17)$$

$$E^i_{\rho\_TE^{II}} = ZH_0 \sin\varphi\, e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \quad (3\text{-}18)$$

$$E^i_{\varphi\_TE^{II}} = ZH_0 \cos\varphi\, e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \quad (3\text{-}19)$$

$$H^i_{\rho\_TE^{II}} = H_0 \cos\varphi \cos\theta^i\, e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \quad (3\text{-}20)$$

$$H^i_{\varphi\_TE^{II}} = -H_0 \sin\varphi \cos\theta^i\, e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \quad (3\text{-}21)$$

completing the full component breakdown description of the incident field.

3.3 Total Incident Field

Due to the hybrid nature of the corrugated cylinder, both TM_z and TE_z modes will be used simultaneously. Therefore, the TM_z and TE_z mode fields in the cylindrical coordinate orientation will be combined to form

$$E^i_{z^{II}} = E_0 \sin\theta^i\, e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \quad (3\text{-}22)$$

$$E^i_{\rho^{II}} = [E_0 \cos\varphi \cos\theta^i + ZH_0 \sin\varphi] e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \quad (3\text{-}23)$$

$$E^i_{\varphi^{II}} = [-E_0 \sin\varphi \cos\theta^i + ZH_0 \cos\varphi] e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \quad (3\text{-}24)$$

$$H^i_{z^{II}} = H_0 \sin\theta^i\, e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \quad (3\text{-}25)$$

$$H^i_{\rho^{II}} = \left[-\frac{E_0}{Z} \sin\varphi + H_0 \cos\varphi \cos\theta^i\right] e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \quad (3\text{-}26)$$

$$H^i_{\varphi^{II}} = \left[-\frac{E_0}{Z} \cos\varphi - H_0 \sin\varphi \cos\theta^i\right] e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi}. \quad (3\text{-}27)$$

CHAPTER 4 REGION I – RADIAL WAVEGUIDE FIELD EQUATIONS

This chapter will focus on deriving the field equations for region I, the region between the corrugations as can be seen in Figure 3-1. The approach to deriving the equations shown is to start with the Fundamental Equations of Guided Waves (FEGW) in cylindrical form, given by

$$E_\rho = \frac{-j}{k_\rho^2}\left[k_z \frac{\partial E_z}{\partial \rho} + \frac{\omega\mu}{\rho}\frac{\partial H_z}{\partial \varphi}\right] \tag{4-1}$$

$$E_\varphi = \frac{-j}{k_\rho^2}\left[\frac{k_z}{\rho}\frac{\partial E_z}{\partial \varphi} - \omega\mu\frac{\partial H_z}{\partial \rho}\right] \tag{4-2}$$

$$H_\rho = \frac{j}{k_\rho^2}\left[\frac{\omega\varepsilon}{\rho}\frac{\partial E_z}{\partial \varphi} - k_z\frac{\partial H_z}{\partial \rho}\right] \tag{4-3}$$

$$H_\varphi = \frac{-j}{k_\rho^2}\left[\omega\varepsilon\frac{\partial E_z}{\partial \rho} + \frac{k_z}{\rho}\frac{\partial H_z}{\partial \varphi}\right] \tag{4-4}$$

where

$$k_\rho = \sqrt{k^2 - k_z^2} \tag{4-5}$$

and as mentioned in Chapter 3, $k = \omega\sqrt{\mu_I \varepsilon_I}$, when referring to the wavenumber k in equations from region I. These equations are derived from Maxwell's equations and can be found in any electromagnetic textbook [13, p. 118].

Note that these FEGWs are in terms of H_z and E_z. When utilizing these equations for TM_z or TE_z modes, H_z or E_z respectively can be set to zero further simplifying the equations. This will be useful in the subsequent sections, in order to derive the appropriate form for many of the equations.

4.1 TM_z Mode Equations for Region I

For the TMz mode, the FEGWs can be rewritten as

$$E_\rho = \frac{-j}{k_\rho^2}\left[k_z \frac{\partial E_z}{\partial \rho}\right] \qquad (4\text{-}6)$$

$$E_\varphi = \frac{-j}{k_\rho^2}\left[\frac{k_z}{\rho}\frac{\partial E_z}{\partial \varphi}\right] \qquad (4\text{-}7)$$

$$H_\rho = \frac{j}{k_\rho^2}\left[\frac{\omega\varepsilon}{\rho}\frac{\partial E_z}{\partial \varphi}\right] \qquad (4\text{-}8)$$

$$H_\varphi = \frac{-j}{k_\rho^2}\left[\omega\varepsilon \frac{\partial E_z}{\partial \rho}\right] \qquad (4\text{-}9)$$

by substituting in H_z=0.

It is now apparent that all the FEGWs equations are in terms of E_z. The only term now left to develop an equation for is E_z. In order to derive a vector field wave equation for E_z, region I will be assumed to be lossless and source free, in order to simplify the mathematics. This will allow for the use of the Helmholtz vector wave equation in the form

$$\nabla^2 E_z + k^2 E_z = 0 \qquad (4\text{-}10)$$

which can be used to derive an equation for E_z. Further detail and derivation of the Helmholtz equations can be found in [13, p. 116].

The other assumption is that E_z, being composed of all the cylindrical field components, or $E_z = E_z(\rho, \varphi, z)$, is separable into its constituent components, such that

$$E_z = R_z(\rho)\phi_z(\varphi)Z_z(z)\,. \qquad (4\text{-}11)$$

Now, substituting equation (4-11) into equation (4-10) yields 3 sets of differential second order equations, one for each coordinate component. Since this is a common process in the field of electromagnets, details of these steps are will not be shown here but can be found in [13, p. 118]. However, solutions to these second order homogenous differential equations take the form of

$$R_z(\rho) = \begin{cases} A_1 J_n(k_\rho \rho) + B_1 Y_n(k_\rho \rho) \leftarrow Standing\ Wave \\ C_1 H_n^{(1)}(k_\rho \rho) + D_1 H_n^{(2)}(k_\rho \rho) \leftarrow Traveling\ Wave \end{cases} \quad (4\text{-}12)$$

$$\phi_z(\varphi) = \begin{cases} A_2 \cos n\varphi + B_2 \sin n\varphi \leftarrow Standing\ Wave \\ C_2 e^{-jn\varphi} + D_2 e^{+jn\varphi} \leftarrow Traveling\ Wave \end{cases} \quad (4\text{-}13)$$

$$Z_z(z) = \begin{cases} A_3 \cos k_z z + B_3 \sin k_z z \leftarrow Standing\ Wave \\ C_3 e^{-jk_z z} + D_3 e^{+jk_z z} \leftarrow Traveling\ Wave \end{cases} . \quad (4\text{-}14)$$

It is important to note the geometry of the structure of region I, which behaves as a radial waveguide, in order to appropriately select the standing wave or traveling wave solution for each of the E_z subcomponents. The structure allows for traveling waves in the ρ and φ direction. However, in the z direction, only a standing wave can exist as depicted in Figure 4-1. Therefore, the solutions for each of the E_z subcomponents will be selected as

$$R_z(\rho) = C_1 H_n^{(1)}(k_\rho \rho) + D_1 H_n^{(2)}(k_\rho \rho) \leftarrow Traveling\ Wave \quad (4\text{-}15)$$

$$\phi_z(\varphi) = C_2 e^{-jn\varphi} + D_2 e^{+jn\varphi} \leftarrow Traveling\ Wave \quad (4\text{-}16)$$

$$Z_z(z) = A_3 \cos k_z z + B_3 \sin k_z z \leftarrow Standing\ Wave \quad (4\text{-}17)$$

which leads to

$$E_z = \left(C_1 H_n^{(1)}(k_\rho \rho) + D_1 H_n^{(2)}(k_\rho \rho)\right)(C_2 e^{-jn\varphi} + D_2 e^{+jn\varphi})(A_3 \cos k_z z + B_3 \sin k_z z) . \quad (4\text{-}18)$$

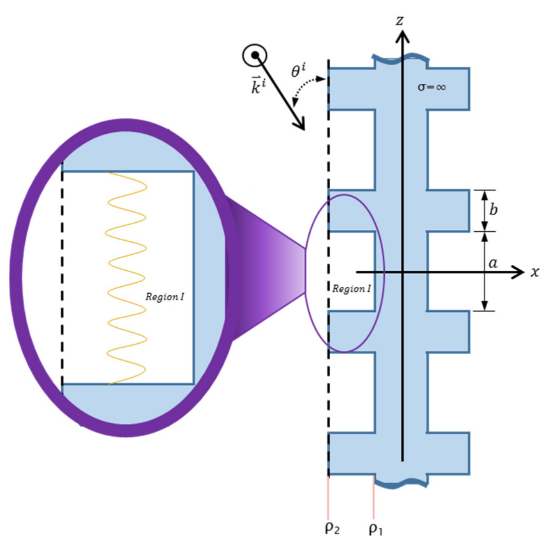


Figure 4-1 Standing wave depicted in region I along the z-axis

4.1.1 Deriving $R_z(\rho)$ of E_z in Region I

From here, the next step is to reduce each E_z subcomponents starting with $R_z(\rho)$. This can be done with using the using boundary condition $\rho=\rho_1$ where $R_z(\rho) = 0$ or

$$C_1 H_n^{(1)}(k_\rho \rho_1) + D_1 H_n^{(2)}(k_\rho \rho_1) = 0 \qquad (4\text{-}19)$$

Now solve for D_1 which will give

$$D_1 = -C_1 \frac{H_n^{(1)}(k_\rho \rho_1)}{H_n^{(2)}(k_\rho \rho_1)}. \qquad (4\text{-}20)$$

Plug D_1 back into equation (4-15) to give

$$C_1 H_n^{(1)}(k_\rho \rho) - C_1 \frac{H_n^{(1)}(k_\rho \rho_1)}{H_n^{(2)}(k_\rho \rho_1)} H_n^{(2)}(k_\rho \rho) = C_1 \left(H_n^{(1)}(k_\rho \rho) - \frac{H_n^{(1)}(k_\rho \rho_1)}{H_n^{(2)}(k_\rho \rho_1)} H_n^{(2)}(k_\rho \rho) \right). \qquad (4\text{-}21)$$

4.1.2 Deriving $\Phi_z(\varphi)$ of E_z in Region I

Simplifying $\Phi_z(\varphi)$ is a little bit easier, as it requires very little manipulation. Since the two terms are exponentials each multiplied by a coefficient, have the same exponents but just with opposite signs, and will be incorporated into a summation that has 'n' going from -∞ to +∞, the two terms can be written as a single term inside a summation. Doing so yields

$$\sum_{n=-\infty}^{\infty} A_n e^{jn\varphi} \qquad (4\text{-}22)$$

where A_n is the coefficient with an 'n' subscript representing the index where, 'n' represents the circumferential (φ) variations.

4.1.3 Deriving $Z_z(z)$ of E_z in Region I

The remaining term (z), for E_z, Z_z, can be simplified using the boundary condition of z=±a/2 where $\frac{\partial Z_z(z)}{\partial z}$ is set equal to 0. This is possible due to the boundary condition of E_ρ and E_φ being equal to zero at z=±a/2 and both having components of $\partial E_z/\partial z$ as shown by

$$E_\varphi \sim E_\rho \sim \frac{\partial Z_z(z)}{\partial z} = \frac{\partial (A_3 \cos k_z z + B_3 \sin k_z z)}{\partial z} = Z_z'(z) = -A_3 k_z \sin k_z z + k_z B_3 \cos k_z z \qquad (4\text{-}23)$$

which will need to be set to zero for this boundary condition. As shown in Figure 4-2, E_φ and E_ρ are tangential to the PEC at z=±a/2 and therefore equal to zero, leading to

$$-A_3 k_z \sin k_z z + k_z B_3 \cos k_z z = 0 = \begin{cases} -A_3 k_z \sin\left(k_z \frac{a}{2}\right) + B_3 k_z \cos\left(k_z \frac{a}{2}\right) = 0 & , \ z = \frac{a}{2} \\ -A_3 k_z \sin\left(-k_z \frac{a}{2}\right) + B_3 k_z \cos\left(-k_z \frac{a}{2}\right) = 0 & , \ z = -\frac{a}{2} \end{cases}. \qquad (4\text{-}24)$$

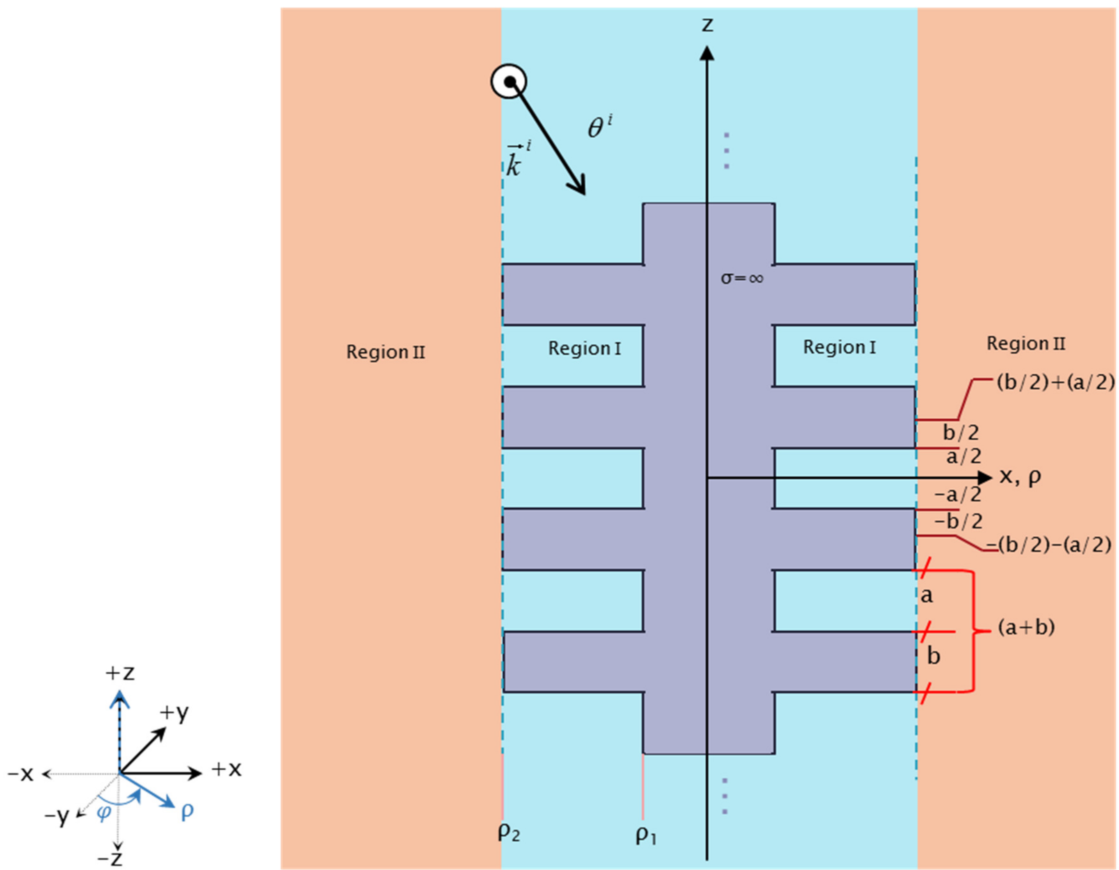


Figure 4-2 Cross-sectional view of corrugated cylinder with multiple boundaries identified

Choosing the equation form where $z = \frac{a}{2}$, the coefficient A_3 can be solved for, which yields

$$A_3 = \frac{B_3 k_z \cos\left(k_z \frac{a}{2}\right)}{k_z \sin\left(k_z \frac{a}{2}\right)}. \tag{4-25}$$

Plug A_3 back into the Z_z (z) equation to get

$$Z_z(z) = \frac{B_3 \cancel{k_z} \cos\left(k_z \frac{a}{2}\right)}{\cancel{k_z} \sin\left(k_z \frac{a}{2}\right)} \cos k_z z + B_3 \sin k_z z = B_3 \left(\sin k_z z + \frac{\cos\left(k_z \frac{a}{2}\right)}{\sin\left(k_z \frac{a}{2}\right)} \cos k_z z \right). \tag{4-26}$$

Since k_z is anticipated to be a constant, a new constant $B_3' = B_3 / \sin k_z \frac{a}{2}$ can be defined and plugged back into Z_z (z) equation giving

$$Z_z(z) = B_3' \left(\sin k_z \frac{a}{2} \sin k_z z + \cos k_z \frac{a}{2} \cos k_z z \right). \tag{4-27}$$

This expression can be simplified further using the trigonometric identity

$$\cos(x - y) = (\sin x \sin y + \cos x \cos y) \tag{4-28}$$

which gives

$$Z_z(z) = B_3' \cos\left(k_z z - k_z \frac{a}{2}\right) \tag{4-29}$$

$$\frac{\partial Z_z(z)}{\partial z} = Z_z'(z) = -B_3' k_z \sin\left(k_z z - k_z \frac{a}{2}\right). \tag{4-30}$$

Up to this point in region I, the k_z has been left in its general form. Now, k_z needs to be defined so that $\frac{\partial Z_z(z)}{\partial z}$ equals zero whenever $z = \pm \frac{a}{2}$, which leads to

$$k_{z_m}\left(z - \frac{a}{2}\right) = k_{z_m}\left(\pm \frac{a}{2} - \frac{a}{2}\right) = -k_{z_m} a = 0 \tag{4-31}$$

$$0 = m\pi \rightarrow k_{z_m} a = -m\pi \rightarrow k_{z_m} = \frac{-m\pi}{a} \tag{4-32}$$

and therefore, provides the final form of the Z_z (z) equation as

$$Z_z(z) = B_3' \cos k_{z_m}\left(z - \frac{a}{2}\right) \tag{4-33}$$

Note that axial wavenumber k_z is now k_{z_m} where the 'm' sub-subscript identifies propagating mode for region I in integer intervals, much the way a classic waveguide does. This will be used in the TE_z mode as well.

4.1.4 Deriving TM_z Mode Equations for Region I

Now that the subcomponents to E_z have been defined, they can come together into a summation as given by

$$E_z^I = \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{m=-\infty}^{\infty} \left(A_{nm} \cos k_{Z_m}\left(z - \frac{a}{2}\right)\right)\left(H_n^{(1)}(k_{\rho_m}\rho) - \frac{H_n^{(1)}(k_{\rho_m}\rho_1)}{H_n^{(2)}(k_{\rho_m}\rho_1)} H_n^{(2)}(k_{\rho_m}\rho)\right) \quad (4\text{-}34)$$

where A_{nm} is the combined coefficients and $k_z = k_{z_m}$, $k_\rho = k_{\rho_m}$ and the radial wavenumber, k_{ρ_m}, is derived as $k_{\rho_m} = \sqrt{k^2 - k_{Z_m}^2}$. From here, E_z is plugged into the modified FEGWs given by equations (4-6) through (4-9) to yield

$$E_{\rho\_TM}^I = -j \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{k_{z_m}}{k_{\rho_m}^2}\left(A_{nm} \cos k_{Z_m}\left(z - \frac{a}{2}\right)\right)\Bigg(\Big(k_{\rho_m} H_{n-1}^{(1)}(k_{\rho_m}\rho) - \frac{n}{\rho} H_n^{(1)}(k_{\rho_m}\rho)\Big) - \frac{H_n^{(1)}(k_{\rho_m}\rho_1)}{H_n^{(2)}(k_{\rho_m}\rho_1)}\Big(k_{\rho_m} H_{n-1}^{(2)}(k_{\rho_m}\rho) - \frac{n}{\rho} H_n^{(2)}(k_{\rho_m}\rho)\Big)\Bigg) \quad (4\text{-}35)$$

$$E_{\varphi\_TM}^I = \frac{1}{\rho} \sum_{n=-\infty}^{\infty} n e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{k_{z_m}}{k_{\rho_m}^2}\left(A_{nm} \cos k_{Z_m}\left(z - \frac{a}{2}\right)\right)\left(H_n^{(1)}(k_{\rho_m}\rho) - \frac{H_n^{(1)}(k_{\rho_m}\rho_1)}{H_n^{(2)}(k_{\rho_m}\rho_1)} H_n^{(2)}(k_{\rho_m}\rho)\right) \quad (4\text{-}36)$$

$$H_{\rho\_TM}^I = -\frac{\omega\varepsilon}{\rho} \sum_{n=-\infty}^{\infty} n e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{1}{k_{\rho_m}^2}\left(A_{nm} \cos k_{Z_m}\left(z - \frac{a}{2}\right)\right)\left(H_n^{(1)}(k_{\rho_m}\rho) - \frac{H_n^{(1)}(k_{\rho_m}\rho_1)}{H_n^{(2)}(k_{\rho_m}\rho_1)} H_n^{(2)}(k_{\rho_m}\rho)\right) \quad (4\text{-}37)$$

$$H_{\varphi\_TM}^I = -j\omega\varepsilon \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{1}{k_{\rho_m}^2}\left(A_{nm} \cos k_{Z_m}\left(z - \frac{a}{2}\right)\right)\Bigg(\Big(k_{\rho_m} H_{n-1}^{(1)}(k_{\rho_m}\rho) - \frac{n}{\rho} H_n^{(1)}(k_{\rho_m}\rho)\Big) - \frac{H_n^{(1)}(k_{\rho_m}\rho_1)}{H_n^{(2)}(k_{\rho_m}\rho_1)}\Big(k_{\rho_m} H_{n-1}^{(2)}(k_{\rho_m}\rho) - \frac{n}{\rho} H_n^{(2)}(k_{\rho_m}\rho)\Big)\Bigg) . \quad (4\text{-}38)$$

4.2 TE_z Mode Equations for Region I

The TE_z mode will follow the same approach as with the TM_z mode starting with rewriting the FEGWs as

$$E_\rho = \frac{-j}{k_\rho^2}\left[\frac{\omega\mu}{\rho}\frac{\partial H_z}{\partial \varphi}\right] \quad (4\text{-}39)$$

$$E_\varphi = \frac{-j}{k_\rho^2}\left[-\omega\mu\frac{\partial H_z}{\partial \rho}\right] \quad (4\text{-}40)$$

$$H_\rho = \frac{j}{k_\rho^2}\left[-k_z\frac{\partial H_z}{\partial \rho}\right] \quad (4\text{-}41)$$

$$H_\varphi = \frac{-j}{k_\rho^2}\left[\frac{k_z}{\rho}\frac{\partial H_z}{\partial \varphi}\right] \quad (4\text{-}42)$$

by substituting in E_z=0.

In the case of the TM_z mode, the FEGWs were in terms of E_z. For the TE_z case, they're in terms of H_z. Just as in the TM_z mode, the Helmholtz vector wave equation will be used, but in the form

$$\nabla^2 H_z + k^2 H_z = 0 \quad (4\text{-}43)$$

along with the assumption of $H_z = H_z(\rho, \varphi, z)$, and that it is separable into its constituent components, such that

$$H_z = R_z(\rho)\phi_z(\varphi)Z_z(z)\ . \quad (4\text{-}44)$$

Using the same solutions from equations (4-15) through (4-17), H_z can be found to be

$$H_z = \left(C_1 H_n^{(1)}(k_\rho\rho) + D_1 H_n^{(2)}(k_\rho\rho)\right)(C_2 e^{-jn\varphi} + D_2 e^{+jn\varphi})(A_3 \cos k_z z + B_3 \sin k_z z)\ . \quad (4\text{-}45)$$

4.2.1 Deriving $R_z(\rho)$ of H_z in Region I

Now, the next step is to reduce each H_z subcomponents starting with $R_z(\rho)$. This approach is similar to the approach taking for E_z. However, a boundary condition for H_z is not directly available but one can be derived from the relationship $E_\varphi \sim \frac{\partial H_z}{\partial \rho} \sim \frac{\partial R_z(\rho)}{\partial \rho} = 0$ at the boundary $\rho=\rho_1$, which is given by

$$\frac{\partial R_z(\rho)}{\partial \rho} = C_1\left(k_{\rho_m} H_{n-1}^{(1)}(k_{\rho_m}\rho) - \frac{n}{\rho} H_n^{(1)}(k_{\rho_m}\rho)\right) + D_1\left(k_{\rho_m} H_{n-1}^{(2)}(k_{\rho_m}\rho) - \frac{n}{\rho} H_n^{(2)}(k_{\rho_m}\rho)\right) = 0\ . \quad (4\text{-}46)$$

This equation can be rearranged to become

$$D_1 = -C_1 \frac{\left(k_{\rho_m} H_{n-1}^{(1)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(1)}(k_{\rho_m}\rho_1)\right)}{\left(k_{\rho_m} H_{n-1}^{(2)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(2)}(k_{\rho_m}\rho_1)\right)} \tag{4-47}$$

which can be plugged into $R_z(\rho)$ to give

$$R_z(\rho) = C_1 H_n^{(1)}(k_{\rho_m}\rho) - C_1 \frac{\left(k_{\rho_m} H_{n-1}^{(1)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(1)}(k_{\rho_m}\rho_1)\right)}{\left(k_{\rho_m} H_{n-1}^{(2)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(2)}(k_{\rho_m}\rho_1)\right)} H_n^{(2)}(k_{\rho_m}\rho) =$$
$$C_1 \left(H_n^{(1)}(k_{\rho_m}\rho) - \frac{\left(k_{\rho_m} H_{n-1}^{(1)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(1)}(k_{\rho_m}\rho_1)\right)}{\left(k_{\rho_m} H_{n-1}^{(2)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(2)}(k_{\rho_m}\rho_1)\right)} H_n^{(2)}(k_{\rho_m}\rho) \right). \tag{4-48}$$

4.2.2 Deriving Φ_z (φ) of H_z in Region I

Deriving Φ_z (φ) H_z follows the same process as E_z. Both terms can be combined within a summation forming

$$\sum_{n=-\infty}^{\infty} B_n e^{jn\varphi}. \tag{4-49}$$

4.2.3 Deriving Z_z (z) of H_z in Region I

The last term for H_z, $Z_z(z)$, will require a boundary condition in order for it to be simplified. $Z_z(z)$ is found to have a relationship with $E_\varphi \sim Z_z(z)$, and E_φ is equal to zero at the boundary where z=±a/2. This boundary condition will be sufficient to simplify $Z_z(z)$. First, the relationship of $E_\varphi \sim Z_z(z)$ needs to be established by using the definitions of E_ρ from equation (4-39) and

$$H_\varphi = \frac{1}{k_{\rho_m}^2 \rho} \frac{\partial^2 H_z}{\partial\varphi \partial z} \tag{4-50}$$

which is an alternate form of a FEGW as described in [14, p. 202]. E_ρ and H_φ can be related as

$$\frac{\partial E_\rho}{\partial z} = -j\omega\mu H_\varphi \tag{4-51}$$

and therefore $\frac{\partial E_\rho}{\partial z} \sim \frac{\partial H_z}{\partial z}$ and $E_\rho \sim H_z$.

Now $Z_z(z)$, which takes the form of equation (4-17), can be equated to zero at the prescribed boundary condition yielding

$$A_3 \cos k_z z + B_3 \sin k_z z = 0 = \begin{cases} A_3 \cos\left(k_z \frac{a}{2}\right) + B_3 \sin\left(k_z \frac{a}{2}\right) = 0 \quad , \quad z = \frac{a}{2} \\ A_3 \cos\left(-k_z \frac{a}{2}\right) + B_3 \sin\left(-k_z \frac{a}{2}\right) = 0 \quad , \quad z = -\frac{a}{2} \end{cases} \tag{4-52}$$

which can now be solved for A_3 in the form

$$A_3 = \frac{-B_3 \sin\left(k_z \frac{a}{2}\right)}{\cos\left(k_z \frac{a}{2}\right)}, z = \frac{a}{2}. \tag{4-53}$$

Plugging A_3 back into $Z_z(z)$ yields

$$Z_z(z) = \frac{-B_3 \sin\left(k_z \frac{a}{2}\right)}{\cos\left(k_z \frac{a}{2}\right)} \cos k_z z + B_3 \sin k_z z = B_3 \left(\sin k_z z - \frac{\sin\left(k_z \frac{a}{2}\right)}{\cos\left(k_z \frac{a}{2}\right)} \cos k_z z\right). \tag{4-54}$$

Since k_z is anticipated to be a constant as was in the derivation for E_z, a new constant $B_3' = B_3/\sin k_z \frac{a}{2}$ can be defined and plugged back into $Z_z(z)$ equation giving

$$Z_z(z) = B_3' \left(\cos\left(k_z \frac{a}{2}\right) \sin k_z z - \sin\left(k_z \frac{a}{2}\right) \cos k_z z\right). \tag{4-55}$$

The trigonometric identity

$$\sin(x - y) = (\sin x \cos y - \cos x \sin y) \tag{4-56}$$

is then used to simplify the $Z_z(z)$ to give the form

$$Z_z(z) = B_3' \sin\left(k_z z - k_z \frac{a}{2}\right) = B_3' \sin k_z \left(z - \frac{a}{2}\right) \tag{4-57}$$

and since $k_z = k_{z_m}$

$$Z_z(z) = B_3' \sin k_{z_m} \left(z - \frac{a}{2}\right). \tag{4-58}$$

4.2.4 Deriving TE_z Mode Equations for Region I

The subcomponents for H_z have been defined and are combined as

$$H_z^I = \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{m=-\infty}^{\infty} \left(B_{nm} \sin k_{Z_m}\left(z - \frac{a}{2}\right)\right)\left(H_n^{(1)}(k_{\rho_m}\rho) - \frac{k_{\rho_m} H_{n-1}^{(1)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(1)}(k_{\rho_m}\rho_1)}{k_{\rho_m} H_{n-1}^{(2)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(2)}(k_{\rho_m}\rho_1)} H_n^{(2)}(k_{\rho_m}\rho)\right) \tag{4-59}$$

within summation form. Just as in the TM_z mode, B_{nm} is the combined coefficients and $k_z = k_{z_m}$, $k_\rho = k_{\rho_m}$ and $k_{\rho_m} = \sqrt{k^2 - k_{Z_m}^2}$.

Following the same steps as in the TM_z mode, H_z is plugged into the modified FEGWs given by equations (4-39) through (4-42) to yield

$$E_{\rho\_TE}^I = \frac{\omega\mu}{\rho} \sum_{n=-\infty}^{\infty} n e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{1}{k_{\rho_m}^2} \left(B_{nm} \sin k_{Z_m}\left(z - \frac{a}{2}\right)\right)\left(H_n^{(1)}(k_{\rho_m}\rho) - \frac{k_{\rho_m} H_{n-1}^{(1)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(1)}(k_{\rho_m}\rho_1)}{k_{\rho_m} H_{n-1}^{(2)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(2)}(k_{\rho_m}\rho_1)} H_n^{(2)}(k_{\rho_m}\rho)\right) \tag{4-60}$$

$$E_{\varphi\_TE}^I = j\omega\mu \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{1}{k_{\rho_m}^2} \left(B_{nm} \sin k_{Z_m}\left(z - \frac{a}{2}\right)\right)\left(\left(k_{\rho_m} H_{n-1}^{(1)}(k_{\rho_m}\rho) - \frac{n}{\rho} H_n^{(1)}(k_{\rho_m}\rho)\right) - \frac{k_{\rho_m} H_{n-1}^{(1)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(1)}(k_{\rho_m}\rho_1)}{k_{\rho_m} H_{n-1}^{(2)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(2)}(k_{\rho_m}\rho_1)}\left(k_{\rho_m} H_{n-1}^{(2)}(k_{\rho_m}\rho) - \frac{n}{\rho} H_n^{(2)}(k_{\rho_m}\rho)\right)\right) \tag{4-61}$$

$$H_{\rho\_TE}^I = -j \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{k_{z_m}}{k_{\rho_m}^2} \left(B_{nm} \sin k_{Z_m}\left(z - \frac{a}{2}\right)\right)\left(\left(k_{\rho_m} H_{n-1}^{(1)}(k_{\rho_m}\rho) - \frac{n}{\rho} H_n^{(1)}(k_{\rho_m}\rho)\right) - \frac{k_{\rho_m} H_{n-1}^{(1)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(1)}(k_{\rho_m}\rho_1)}{k_{\rho_m} H_{n-1}^{(2)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(2)}(k_{\rho_m}\rho_1)}\left(k_{\rho_m} H_{n-1}^{(2)}(k_{\rho_m}\rho) - \frac{n}{\rho} H_n^{(2)}(k_{\rho_m}\rho)\right)\right) \tag{4-62}$$

$$H_{\varphi\_TE}^{I} = \frac{1}{\rho}\sum_{n=-\infty}^{\infty} n e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{k_{z_m}}{k_{\rho_m}^2}\left(B_{nm} \sin k_{Z_m}\left(z-\frac{a}{2}\right)\right)\left(H_n^{(1)}(k_{\rho_m}\rho) - \frac{k_{\rho_m} H_{n-1}^{(1)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(1)}(k_{\rho_m}\rho_1)}{k_{\rho_m} H_{n-1}^{(2)}(k_{\rho_m}\rho_1) - \frac{n}{\rho_1} H_n^{(2)}(k_{\rho_m}\rho_1)} H_n^{(2)}(k_{\rho_m}\rho)\right) \quad . \tag{4-63}$$

CHAPTER 5 REGION II – SCATTERED FIELD EQUATIONS

This chapter will focus on the derivation of the scattered field in region II. This is the same region in which the incident fields were described in Chapter 3. By the end of this chapter, all the fields in region II would will be covered.

According to [12, p. 615], a perfectly smooth cylinder that is infinitely long and a PEC does not depolarize an incident wave. Balanis continues on to describe that deviations from this can cause depolarization of the incident wave [12, p. 615]. Though not explicitly stated by Balanis, his statement can be interpreted to be applied to that of the periodic corrugated cylinder which would depolarize an incident plane wave. Therefore, as stated in section 2.3 , a hybrid mode of TM_z and TE_z can exist for the periodic corrugated cylinder and will be examined as such. The field equations for the TM_z and TE_z modes will be derived separately, but through the principle of superposition will be combined in Chapter 5 when finding solutions for the fields.

When deriving the field equations for the scattered field, the FEGWs presented in Chapter 4, equations (4-1) through (4-4), will be made much use of. Also from Chapter 4, deriving the E_z and H_z scattered fields will use the same Helmholtz equation representation from (4-10) and (4-43) respectively, as well as the same coordinate constituent separable equations from (4-11) and (4-44) respectively. Use of the known solutions for the Helmholtz equation described in equations (4-12) through (4-14) will also be made in this chapter.

5.1 Equations for the TM_z Mode Scattered Field of Region II

The steps to derive the equations representing the TM_z mode of the scattered field in region II will be the same as that for the TM_z mode in region I. This starts with choosing the appropriate field representation from equations (4-12) through (4-14) based on geometry and expected behavior in order to describe E_z. Since there are no restrictive boundaries in region II, the traveling wave representation is chosen for each constituent and replaced into (4-11) to give

$$E_z = \left(C_1 H_n^{(1)}(k_\rho \rho) + D_1 H_n^{(2)}(k_\rho \rho)\right)(C_2 e^{-jn\varphi} + D_2 e^{+jn\varphi})(C_3 e^{-jk_z z} + C_3 e^{jk_z z})\,. \quad (5\text{-}1)$$

Now, the first and second term of Ez, which make up $R_z(\rho)$ represent inward and outward traveling waves respectively. Since at the boundary of region II, whether with the conductive surface or with region I, there is no scattered field expected, the inward traveling wave portion (the first term) can be eliminated leaving

$$E_z = D_1 H_n^{(2)}(k_\rho \rho)(C_2 e^{-jn\varphi} + D_2 e^{+jn\varphi})(C_3 e^{-jk_z z} + C_3 e^{jk_z z})\,. \quad (5\text{-}2)$$

This equation can be further simplified by bringing it into summation terms as was done in Chapter 4 yielding

$$E_{z_{II}}^{s} = \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{l=-\infty}^{\infty} C_{nl} e^{-jk_{z_l} z} H_n^{(2)}(k_{\rho_l} \rho)\,. \quad (5\text{-}3)$$

Note that axial wavenumber k_z has been replaced by k_{z_l} and radial wavenumber k_ρ has been replaced by k_{ρ_l}, where $k_{\rho_l} = \sqrt{k^2 - k_{z_l}^2}$. Here, the sub-subscript 'l' identifies the propagating mode for region II in the same way 'm' is for region I. To understand the relationship of the propagating mode with the geometry, a derivation of k_{z_l}, is required. This is due to the fact that a propagating wave interfacing with a structure, in this case the scattered field with the periodic corrugated cylinder, takes on the symmetry of said structure as depicted in Figure 5-1. This k_{z_l} is known as the Floquet harmonic or Floquet wavenumber [9, p. 265] or a Bloch wavenumber [15] because of the non-uniqueness and ability to represent a periodic medium.

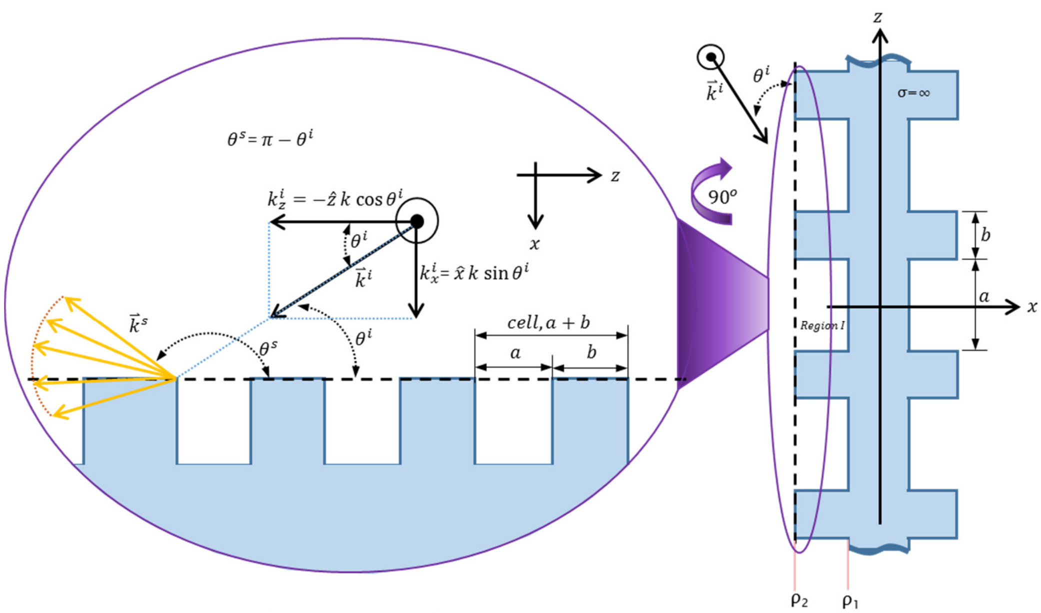


Figure 5-1 A depiction of the vector decomposition of the wavenumber in region II

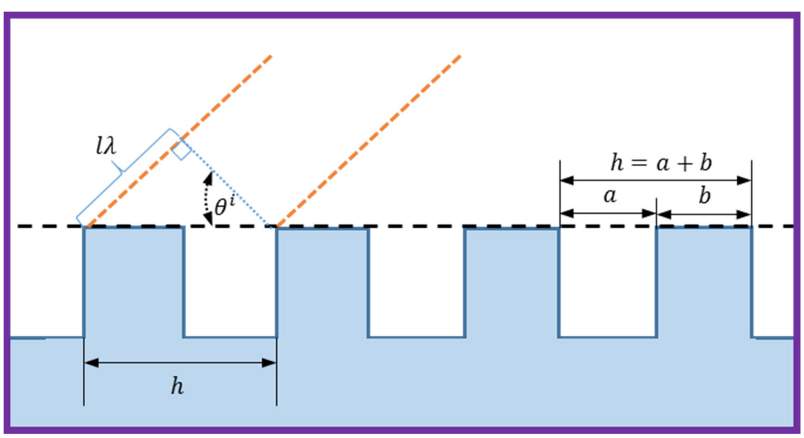


Figure 5-2 A depiction of constructive interference for a propagating wave along a periodic surface

First, for wave modes to exist, a constructive interference relationship needs to be established along the path of propagation. This is how wave modes are established within waveguides and periodic structures. It can be seen from Figure 5-2 that an integer multiple ‘l’ to the wavelength ‘λ’ would provide a description of the propagating modes along the prescribed surface where constructive interface would occur. Also, the relationship

$$l\lambda = (a+b)\,sin\,\theta^i \qquad (5\text{-}4)$$

is established. The relationship for the scattered field wave vector

$$\vec{k}^s = -\hat{z}\,k\,cos\,\theta^s + \hat{x}\,k\,sin\,\theta^s \qquad (5\text{-}5)$$

is established from Figure 5-1 where the magnitude of $\vec{k}^s$, which will be referred to as k_{z_l}, the axial wavenumber in region II, is

$$k_{z_l} = k\,cos\,\theta^i + k\,sin\,\theta^i. \qquad (5\text{-}6)$$

when substituting in the relationship of $\theta^s = \pi - \theta^i$ as per [12, p. 615]. Substituting in the definition $k = \frac{2\pi}{\lambda}$ for the second term yields

$$k_{z_l} = k\,cos\,\theta^i + \frac{2\pi}{\lambda}\,sin\,\theta^i \qquad (5\text{-}7)$$

which can further be reduced by using equation (5-4) to give

$$\frac{l\lambda}{(a+b)} = sin\,\theta^i \qquad (5\text{-}8)$$

$$k_{z_l} = k\,cos\,\theta^i + \frac{2\pi}{\lambda}\frac{l\lambda}{(a+b)} = k\,cos\,\theta^i + \frac{2\pi l}{(a+b)} \qquad (5\text{-}9)$$

which completes the derivation for k_{z_l}. This also completes the definition for $E^s_{z^{II}}$.

Next, the remaining TM_z mode scattered field equations are derived. This is done by substituting $E^s_{z^{II}}$ into equations (4-6) through (4-9) to produce

$$E^s_{\rho\_TM^{II}} = -j\sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{k_{z_l}}{k_{\rho_l}^2} C_{nl} e^{-jk_{z_l}z} \left(k_{\rho_l} H_{n-1}^{(2)}(k_{\rho_l}\rho) - \frac{nH_n^{(2)}(k_{\rho_l}\rho)}{\rho} \right) \qquad (5\text{-}10)$$

$$E^s_{\varphi\_TM^{II}} = \frac{1}{\rho}\sum_{n=-\infty}^{\infty} ne^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{k_{z_l}}{k_{\rho_l}^2} C_{nl} e^{-jk_{z_l}z} H_n^{(2)}(k_{\rho_l}\rho) \qquad (5\text{-}11)$$

$$H^s_{\rho\_TM^{II}} = -\frac{\omega\varepsilon}{\rho}\sum_{n=-\infty}^{\infty} ne^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{1}{k_{\rho_l}^2} C_{nl} e^{-jk_{z_l}z} H_n^{(2)}(k_{\rho_l}\rho) \qquad (5\text{-}12)$$

$$H^s_{\varphi\_TM^{II}} = -j\omega\varepsilon \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{1}{k_{\rho_l}^2} C_{nl} e^{-jk_{z_l}z} \left(k_{\rho_l} H_{n-1}^{(2)}(k_{\rho_l}\rho) - \frac{nH_n^{(2)}(k_{\rho_l}\rho)}{\rho} \right). \qquad (5\text{-}13)$$

5.2 Equations for the TE$_z$ Mode Scattered Field of Region II

For the TE$_z$ mode, H$_z$ is derived much the same way as was in the TE$_z$ mode in region I. Once applying the same assumptions from the geometry and expected field behavior described in section 5.1 for the TM$_z$ mode case, onto the equations (4-12) through (4-14) and to equation (4-44), H$_z$ is defined as

$$H^s_{z^{II}} = \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{l=-\infty}^{\infty} D_{nl} e^{-jk_{z_l}z} H_n^{(2)}(k_{\rho_l}\rho)\,. \qquad (5\text{-}14)$$

where D$_{nl}$ represents the combined coefficients and k_{z_l} and k_{ρ_l} are the axial and radial wavenumbers defined in section 5.1 respectively.

Now with a fully defined $H^s_{z^{II}}$, the remaining equations can be found by plugging $H^s_{z^{II}}$ into equations (4-39) through (4-42) to yield

$$E^s_{\rho\_TE^{II}} = -j\omega\mu * j \sum_{n=-\infty}^{\infty} n e^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{1}{k_{\rho_l}^2 \rho} D_{nl} e^{-jk_{z_l}z} H_n^{(2)}(k_{\rho_l}\rho) \qquad (5\text{-}15)$$

$$E^s_{\varphi\_TE^{II}} = j\omega\mu \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{1}{k_{\rho_l}^2} D_{nl} e^{-jk_{z_l}z} \left(k_{\rho_l} H_{n-1}^{(2)}(k_{\rho_l}\rho) - \frac{nH_n^{(2)}(k_{\rho_l}\rho)}{\rho} \right) \qquad (5\text{-}16)$$

$$H^s_{\rho\_TE^{II}} = -j \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{k_{z_l}}{k_{\rho_l}^2} D_{nl} e^{-jk_{z_l}z} \left(k_{\rho_l} H_{n-1}^{(2)}(k_{\rho_l}\rho) - \frac{nH_n^{(2)}(k_{\rho_l}\rho)}{\rho} \right) \qquad (5\text{-}17)$$

$$H^s_{\varphi\_TE^{II}} = \frac{1}{\rho} \sum_{n=-\infty}^{\infty} n e^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{k_{z_l}}{k_{\rho_l}^2} D_{nl} e^{-jk_{z_l}z} H_n^{(2)}(k_{\rho_l}\rho)\,. \qquad (5\text{-}18)$$

CHAPTER 6 BOUNDARY CONDITIONS & POINT MATCHING METHOD

At this point, the equations for region I and region II have been defined. However, the equations for region I and the scattered field equations for region II have unknown coefficients that need to be solved for in order to have a complete representation, and therefore to be able to calculate and predict the scattered field. In order to achieve this, boundary conditions need to be established to provide equation sets that allow for algebraic solving of the unknown coefficients. The equation relationships are established by point matching, equating points from different equations, that establishes an equal number of point matches to unknowns.

6.1 General Boundary Conditions

There are two main boundary regions that are utilized in this paper for point matching, as can be observed from Figure 6-1, which are boundary 'a' and boundary 'b'.

Boundary 'a':

- Boundary between region I and II at $\rho = \rho_2$ for a z range of $\frac{-a}{2} \leq z \leq \frac{a}{2}$.
- The available matching points, $p_j(\rho, \varphi, z)$, of a quantity 'j' are: $\rho = \rho_2, \frac{-a}{2} \leq z \leq \frac{a}{2}$ and $0 \leq \varphi \leq 2\pi$.
- Tangential electric fields from region II, E_z^{II} and E_φ^{II}, are equal to tangential electric fields from region I, E_z^I and E_φ^I, hence $E_z^{II}=E_z^I$ and $E_\varphi^{II}=E_\varphi^I$ [16, p. 178].
- Tangential magnetic fields from region II, H_z^{II} and H_φ^{II}, are equal to tangential magnetic fields from region I, H_z^I and H_φ^I, hence $H_z^{II}=H_z^I$ and $H_\varphi^{II}=H_\varphi^I$ when the

conductivity is finite, hence, not a PEC which is the case at this boundary [16, p. 234].

- Used to solve for the unknown expansion coefficients A_{nm} and B_{nm} from region I.
- Used to solve for the unknown expansion coefficients C_{nl} and D_{nl} from region II, which are distinct for each boundary, therefore these will have superscript of (a) to identify them: $C_{nl}^{(a)}$ and $D_{nl}^{(a)}$.

Boundary 'b':

- Boundary between region II and outer conducting surface of corrugated cylinder at $\rho = \rho_2$ for a z range of $\frac{a}{2} \leq z \leq \frac{a}{2} + b$.
- The available matching points, $p_j(\rho, \varphi, z)$, of a quantity 'j' are: $\rho = \rho_2, \frac{a}{2} \leq z \leq \frac{a}{2} + b$ and $0 \leq \varphi \leq 2\pi$.
- Tangential electric fields from region II are equal to zero based on the Dirichlet boundary condition [9, pp. 97-100]: $E_z^{II} = 0$ and $E_\varphi^{II} = 0$.
- Used to solve for the unknown expansion coefficients C_{nl} and D_{nl} from region II, which are distinct and will have superscript (b) to identify them, $C_{nl}^{(b)}$ and $D_{nl}^{(b)}$.

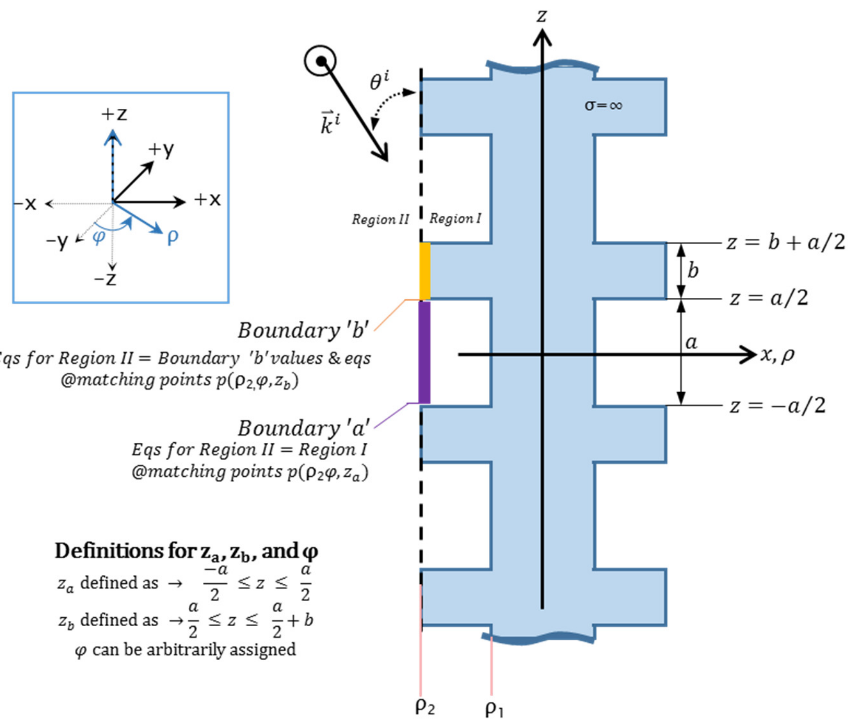


Figure 6-1 A depiction for the general boundary conditions utilized to solve for the unknown coefficients

6.2 Matrix Form of Field Equations

In order to effectively apply the boundary conditions and to solve for the unknown expansion coefficients, the field equations described in Chapter 3, Chapter 4 and Chapter 5 will be represented in matrix form. This will aid in the algebraic manipulation and solving of the coefficients.

There are a few things to note about the matrix form and indices. The expansion coefficients of each region need to have the same shape as each other for point matching, therefore the $n \times m$ size matrices of region I need to match the $n \times l$ matrices of region II. This is accomplished by making "n" of each region equal to each other and having "m" of region I equal to "l" of region II.

The matrices that contain a subscript "j" indicate that there are z's and φ's referenced in them, where ρ is always ρ_2 for point matches. Each "j" is a matching point, which is provided by z's and φ's variation on the specified boundary with a fixed ρ_2 and z range indicated. The number of matching points, which is the number of "j" points, is equal to $n * m$ or $n * l$, thus there's a matching point for every expanded "n" and "m" or "n" and "l". Also, the number of equation matches equals the number of expansion coefficients. The number of matching points times the number of equation matches equals the total number of unknown expanded coefficients which provides a complete system of equations.

In order to separate the coefficients from the rest of the terms for each equation, the matrices of the unknown expansion coefficients are flattened into an array of a single column with $n * m$ or $n * l$ rows and the remainder of the term can stay in a matrix form of "j" rows and $n * m$ or $n * l$ columns. When multiplied back together, it produces the summation equation. The incident fields are also "j" rows, but of a single column of which each entry is a full summation solution.

6.2.1 Matrix form of Region I Equations

The following are the matrix form of the subcomponents of the region I equations described in Chapter 4. The equations listed below capture the fields E_z^I, H_z^I, E_φ^I, H_φ^I, E_ρ^I and H_ρ^I in matrix form. For each equation in the list, the corresponding matrix form in which it's used is provided. Also, the equation's substituted form is identified and located within the matrix form through the use of highlighting and bolding the font.

$$[h_{nmj}] = \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{m=-\infty}^{\infty} \left(\cos k_{Z_m}\left(z - \frac{a}{2}\right)\right)\left(H_n^{(1)}(k_{\rho_m}\rho) - \frac{H_n^{(1)}(k_{\rho_m}\rho_1)}{H_n^{(2)}(k_{\rho_m}\rho_1)} H_n^{(2)}(k_{\rho_m}\rho)\right) \quad (6\text{-}1)$$

$$\leftarrow From\ E_z^I = [\boldsymbol{h_{nmj}}][A_{nm}]$$

$$[i_{nmj}] = \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{m=-\infty}^{\infty} \left(\sin k_{z_m}\left(z-\frac{a}{2}\right)\right)\left(H_n^{(1)}(k_{\rho m}\rho) - \frac{k_{\rho m}H_{n-1}^{(1)}(k_{\rho m}\rho_1)-\frac{n}{\rho_1}H_n^{(1)}(k_{\rho m}\rho_1)}{k_{\rho m}H_{n-1}^{(2)}(k_{\rho m}\rho_1)-\frac{n}{\rho_1}H_n^{(2)}(k_{\rho m}\rho_1)} H_n^{(2)}(k_{\rho m}\rho)\right) \leftarrow \text{From } H_z^I = [\boldsymbol{i_{nmj}}][B_{nm}] \tag{6-2}$$

$$[k_{nmj}] = \frac{1}{\rho}\sum_{n=-\infty}^{\infty} n e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{k_{zm}}{k_{\rho m}^2}\left(\cos k_{z_m}\left(z-\frac{a}{2}\right)\right)\left(H_n^{(1)}(k_{\rho m}\rho) - \frac{H_n^{(1)}(k_{\rho m}\rho_1)}{H_n^{(2)}(k_{\rho m}\rho_1)} H_n^{(2)}(k_{\rho m}\rho)\right) \leftarrow \text{From } E_\varphi^I = [\boldsymbol{k_{nmj}}][A_{nm}] + [l_{nmj}][B_{nm}] \tag{6-3}$$

$$[l_{nmj}] = j\omega\mu \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{1}{k_{\rho m}^2}\left(\sin k_{z_m}\left(z-\frac{a}{2}\right)\right)\left(\left(k_{\rho m}H_{n-1}^{(1)}(k_{\rho m}\rho) - \frac{n}{\rho}H_n^{(1)}(k_{\rho m}\rho)\right) - \frac{k_{\rho m}H_{n-1}^{(1)}(k_{\rho m}\rho_1)-\frac{n}{\rho_1}H_n^{(1)}(k_{\rho m}\rho_1)}{k_{\rho m}H_{n-1}^{(2)}(k_{\rho m}\rho_1)-\frac{n}{\rho_1}H_n^{(2)}(k_{\rho m}\rho_1)}\left(k_{\rho m}H_{n-1}^{(2)}(k_{\rho m}\rho) - \frac{n}{\rho}H_n^{(2)}(k_{\rho m}\rho)\right)\right) \leftarrow \text{From } E_\varphi^I = [k_{nmj}][A_{nm}] + [\boldsymbol{l_{nmj}}][B_{nm}] \tag{6-4}$$

$$[o_{nmj}] = -j\omega\varepsilon \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{1}{k_{\rho m}^2}\left(\cos k_{z_m}\left(z-\frac{a}{2}\right)\right)\left(\left(k_{\rho m}H_{n-1}^{(1)}(k_{\rho m}\rho) - \frac{n}{\rho}H_n^{(1)}(k_{\rho m}\rho)\right) - \frac{H_n^{(1)}(k_{\rho m}\rho_1)}{H_n^{(2)}(k_{\rho m}\rho_1)}\left(k_{\rho m}H_{n-1}^{(2)}(k_{\rho m}\rho) - \frac{n}{\rho}H_n^{(2)}(k_{\rho m}\rho)\right)\right) \leftarrow \text{From } H_\varphi^I = [\boldsymbol{o_{nmj}}][A_{nm}] + [p_{nmj}][B_{nm}] \tag{6-5}$$

$$[p_{nmj}] = \frac{1}{\rho}\sum_{n=-\infty}^{\infty} n e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{k_{zm}}{k_{\rho m}^2}\left(\sin k_{z_m}\left(z-\frac{a}{2}\right)\right)\left(H_n^{(1)}(k_{\rho m}\rho) - \frac{k_{\rho m}H_{n-1}^{(1)}(k_{\rho m}\rho_1)-\frac{n}{\rho_1}H_n^{(1)}(k_{\rho m}\rho_1)}{k_{\rho m}H_{n-1}^{(2)}(k_{\rho m}\rho_1)-\frac{n}{\rho_1}H_n^{(2)}(k_{\rho m}\rho_1)} H_n^{(2)}(k_{\rho m}\rho)\right) \leftarrow \text{From } H_\varphi^I = [o_{nmj}][A_{nm}] + [\boldsymbol{p_{nmj}}][B_{nm}] \tag{6-6}$$

$$[r_{nmj}] = -j \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{k_{zm}}{k_{\rho m}^2}\left(\cos k_{z_m}\left(z-\frac{a}{2}\right)\right)\left(\left(k_{\rho m}H_{n-1}^{(1)}(k_{\rho m}\rho) - \frac{n}{\rho}H_n^{(1)}(k_{\rho m}\rho)\right) - \frac{H_n^{(1)}(k_{\rho m}\rho_1)}{H_n^{(2)}(k_{\rho m}\rho_1)}\left(k_{\rho m}H_{n-1}^{(2)}(k_{\rho m}\rho) - \frac{n}{\rho}H_n^{(2)}(k_{\rho m}\rho)\right)\right) \leftarrow \text{From } E_\rho^I = [\boldsymbol{r_{nmj}}][A_{nm}] + [s_{nmj}][B_{nm}] \tag{6-7}$$

$$[s_{nmj}] = \frac{\omega\mu}{\rho}\sum_{n=-\infty}^{\infty} n e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{1}{k_{\rho m}^2}\left(\sin k_{z_m}\left(z-\frac{a}{2}\right)\right)\left(H_n^{(1)}(k_{\rho m}\rho) - \frac{k_{\rho m}H_{n-1}^{(1)}(k_{\rho m}\rho_1)-\frac{n}{\rho_1}H_n^{(1)}(k_{\rho m}\rho_1)}{k_{\rho m}H_{n-1}^{(2)}(k_{\rho m}\rho_1)-\frac{n}{\rho_1}H_n^{(2)}(k_{\rho m}\rho_1)} H_n^{(2)}(k_{\rho m}\rho)\right) \leftarrow \text{From } E_\rho^I = [r_{nmj}][A_{nm}] + [\boldsymbol{s_{nmj}}][B_{nm}] \tag{6-8}$$

$$[t_{nmj}] = -\frac{\omega\varepsilon}{\rho}\sum_{n=-\infty}^{\infty} n e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{1}{k_{\rho m}^2}\left(\cos k_{z_m}\left(z-\frac{a}{2}\right)\right)\left(H_n^{(1)}(k_{\rho_m}\rho) - \frac{H_n^{(1)}(k_{\rho_m}\rho_1)}{H_n^{(2)}(k_{\rho_m}\rho_1)} H_n^{(2)}(k_{\rho_m}\rho)\right) \leftarrow From\ H_\rho^I = \mathbf{[t_{nmj}]}[A_{nm}] + [u_{nmj}][B_{nm}] \tag{6-9}$$

$$[u_{nmj}] = -j\sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{m=-\infty}^{\infty} \frac{k_{zm}}{k_{\rho m}^2}\left(\sin k_{z_m}\left(z-\frac{a}{2}\right)\right)\left(\left(k_{\rho_m} H_{n-1}^{(1)}(k_{\rho_m}\rho) - \frac{n}{\rho} H_n^{(1)}(k_{\rho_m}\rho)\right) - \frac{k_{\rho m}H_{n-1}^{(1)}(k_{\rho m}\rho_1) - \frac{n}{\rho_1}H_n^{(1)}(k_{\rho m}\rho_1)}{k_{\rho m}H_{n-1}^{(2)}(k_{\rho m}\rho_1) - \frac{n}{\rho_1}H_n^{(2)}(k_{\rho m}\rho_1)}\left(k_{\rho_m} H_{n-1}^{(2)}(k_{\rho_m}\rho) - \frac{n}{\rho} H_n^{(2)}(k_{\rho_m}\rho)\right)\right) \leftarrow From\ H_\rho^I = [t_{nmj}][A_{nm}] + \mathbf{[u_{nmj}]}[B_{nm}] \tag{6-10}$$

The TM_z mode region I field expansion coefficient is represented by $[A_{nm}]$ and the TE_z mode region I field expansion coefficient is represented by $[B_{nm}]$. The remaining are region I field components of the respective mode to the coefficient it's a product with.

6.2.2 Matrix form of Region II Equations

The following are the matrix form of the subcomponents of the region II equations described in Chapter 3 and Chapter 5. This includes the superposition of all the region II fields within the same coordinate axis, including TM_z and TE_z modes for both the scattered and incident field. Hence, the total field for each axis is summarized as

$$E_z^{II} = E_{z^{II}}^i + E_{z\_TM^{II}}^s \tag{6-11}$$

$$E_\rho^{II} = E_{\rho^{II}}^i + E_{\rho\_TM^{II}}^s + E_{\rho\_TE^{II}}^s \tag{6-12}$$

$$E_\varphi^{II} = E_{\varphi^{II}}^i + E_{\varphi\_TM^{II}}^s + E_{\varphi\_TE^{II}}^s \tag{6-13}$$

$$H_z^{II} = H_{z^{II}}^i + H_{z\_TE^{II}}^s \tag{6-14}$$

$$H_\rho^{II} = H_{\rho^{II}}^i + H_{\rho\_TM^{II}}^s + H_{\rho\_TE^{II}}^s \tag{6-15}$$

$$H_\varphi^{II} = H_{\varphi^{II}}^i + H_{\varphi\_TM^{II}}^s + H_{\varphi\_TE^{II}}^s \tag{6-16}$$

Below are the field components of E_z^{II}, H_z^{II}, E_φ^{II}, H_φ^{II}, E_ρ^{II} and H_ρ^{II} in matrix form. Just as in section 6.2.1 , the equation's substituted form is identified and located within the matrix form through the use of highlighting and bolding the font.

$$[b_{nj}] = E_0 \sin\theta^i e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \leftarrow From\ E_z^{II} = [\boldsymbol{b_{nj}}] + [a_{nl}][C_{nl}] \quad (6\text{-}17)$$

$$[a_{nlj}] = \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{l=-\infty}^{\infty} e^{-jk_{z_l}z} H_n^{(2)}(k_{\rho_l}\rho) \leftarrow From\ E_z^{II} = [b_{nj}] + [\boldsymbol{a_{nl}}][C_{nl}] \quad (6\text{-}18)$$

$$[f_{nj}] = H_0 \sin\theta^i e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \leftarrow From\ H_z^{II} = [\boldsymbol{f_{nj}}] + [g_{nl}][D_{nl}] \quad (6\text{-}19)$$

$$[g_{nlj}] = \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{l=-\infty}^{\infty} e^{-jk_{z_l}z} H_n^{(2)}(k_{\rho_l}\rho) \leftarrow From\ H_z^{II} = [f_{nj}] + [\boldsymbol{g_{nlj}}][D_{nl}] \quad (6\text{-}20)$$

$$[d_{nj}] = [-E_0 \sin\varphi \cos\theta^i + ZH_0 \cos\varphi] e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \leftarrow From\ E_\varphi^{II} = [\boldsymbol{d_{nj}}] + ([c_{nlj}][C_{nl}] + [e_{nlj}][D_{nl}]) \quad (6\text{-}21)$$

$$[c_{nlj}] = \frac{1}{\rho} \sum_{n=-\infty}^{\infty} n e^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{k_{z_l}}{k_{\rho_l}^2} e^{-jk_{z_l}z} H_n^{(2)}(k_{\rho_l}\rho) \leftarrow From\ E_\varphi^{II} = [d_{nj}] + ([\boldsymbol{c_{nlj}}][C_{nl}] + [e_{nlj}][D_{nl}]) \quad (6\text{-}22)$$

$$[e_{nlj}] = j\omega\mu \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{1}{k_{\rho_l}^2} e^{-jk_{z_l}z} \left(k_{\rho_l} H_{n-1}^{(2)}(k_{\rho_l}\rho) - \frac{n H_n^{(2)}(k_{\rho_l}\rho)}{\rho} \right) \leftarrow From\ E_\varphi^{II} = [d_{nj}] + ([c_{nlj}][C_{nl}] + [\boldsymbol{e_{nlj}}][D_{nl}]) \quad (6\text{-}23)$$

$$[m_{nj}] = \left[-\frac{E_0}{Z} \cos\varphi - H_0 \sin\varphi \cos\theta^i \right] e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \leftarrow From\ H_\varphi^{II} = [\boldsymbol{m_{nj}}] + ([n_{nlj}][C_{nl}] + [q_{nlj}][D_{nl}]) \quad (6\text{-}24)$$

$$[n_{nlj}] = -j\omega\varepsilon \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{1}{k_{\rho_l}^2} e^{-jk_{z_l}z} \left(k_{\rho_l} H_{n-1}^{(2)}(k_{\rho_l}\rho) - \frac{n H_n^{(2)}(k_{\rho_l}\rho)}{\rho} \right) \leftarrow From\ H_\varphi^{II} = [m_{nj}] + ([\boldsymbol{n_{nlj}}][C_{nl}] + [q_{nlj}][D_{nl}]) \quad (6\text{-}25)$$

$$[q_{nlj}] = \frac{1}{\rho} \sum_{n=-\infty}^{\infty} n e^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{k_{z_l}}{k_{\rho_l}^2} e^{-jk_{z_l}z} H_n^{(2)}(k_{\rho_l}\rho) \leftarrow From\ H_\varphi^{II} = [m_{nj}] + ([n_{nlj}][C_{nl}] + [\boldsymbol{q_{nlj}}][D_{nl}]) \quad (6\text{-}26)$$

$$[\gamma_{nj}] = [E_0 \cos\varphi \cos\theta^i + ZH_0 \sin\varphi] e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \leftarrow From\ E_\rho^{II} = [\boldsymbol{\gamma_{nj}}] + ([v_{nlj}][C_{nl}] + [w_{nlj}][D_{nl}]) \quad (6\text{-}27)$$

$$[v_{nlj}] = -j \sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{k_{z_l}}{k_{\rho_l}^2} e^{-jk_{z_l}z} \left(k_{\rho_l} H_{n-1}^{(2)}(k_{\rho_l}\rho) - \frac{n H_n^{(2)}(k_{\rho_l}\rho)}{\rho} \right) \leftarrow From\ E_\rho^{II} = [\gamma_{nj}] + ([\boldsymbol{v_{nlj}}][C_{nl}] + [w_{nlj}][D_{nl}]) \quad (6\text{-}28)$$

$$[w_{nlj}] = \frac{\omega\mu}{\rho} \sum_{n=-\infty}^{\infty} n e^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{1}{k_{\rho_l}^2} e^{-jk_{z_l}z} H_n^{(2)}(k_{\rho_l}\rho) \leftarrow From\ E_\rho^{II} = [\gamma_{nj}] + ([v_{nlj}][C_{nl}] + [\boldsymbol{w_{nlj}}][D_{nl}]) \quad (6\text{-}29)$$

$$[\delta_{nj}] = \left[-\frac{E_0}{Z} \sin\varphi + H_0 \cos\varphi \cos\theta^i \right] e^{jkz\cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho \sin\theta^i) e^{jn\varphi} \leftarrow From\ H_\rho^{II} = [\boldsymbol{\delta_{nj}}] + ([x_{nlj}][C_{nl}] + [y_{nlj}][D_{nl}]) \quad (6\text{-}30)$$

$$[x_{nlj}] = -\frac{\omega\varepsilon}{\rho} \sum_{n=-\infty}^{\infty} n e^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{1}{k_{\rho_l}^2} e^{-jk_{z_l}z} H_n^{(2)}(k_{\rho_l}\rho) \leftarrow From\ H_\rho^{II} = [\delta_{nj}] + ([\boldsymbol{x_{nlj}}][C_{nl}] + [y_{nlj}][D_{nl}]) \quad (6\text{-}31)$$

$$[y_{nlj}] = -j\sum_{n=-\infty}^{\infty} e^{jn\varphi} \sum_{l=-\infty}^{\infty} \frac{k_{z_l}}{k_{\rho_l}^2} e^{-jk_{z_l}z}\left(k_{\rho_l} H_{n-1}^{(2)}(k_{\rho_l}\rho) - \frac{nH_n^{(2)}(k_{\rho_l}\rho)}{\rho}\right) \leftarrow From\ H_\rho^{II} = [\delta_{nj}] + ([x_{nlj}][C_{nl}] + [\boldsymbol{y_{nlj}}][D_{nl}]) \quad (6\text{-}32)$$

The components $[b_{nj}], [f_{nj}], [d_{nj}], [m_{nj}], [\gamma_{nj}]$ and $[\delta_{nj}]$ represent the incident field of the corresponding cylindrical orientation with both TM_z and TE_z mode combined. The TM_z mode scattered field expansion coefficient is represented by $[C_{nl}]$ and the TE_z mode scattered field expansion coefficient is represented by $[D_{nl}]$. The remaining are scattered field components of the respective mode to the coefficient it's a product with.

6.2.3 Matrix Equation Matches for Point Matching

There are 6 unknown expansion coefficients, each of which expand out to either $n * m$ or $n * l$ quantities of unknowns. This means that there needs to be at least 6 matrix equation matches in order to solve for the unknowns. Based on the boundary conditions discussed in section 6.1 the following 6 equations will provide a system of equations to solve for the unknowns.

For boundary 'a':

$$[E_z^I] = [E_z^{II(a)}] \rightarrow [h_{nmj}][A_{nm}] = [b_{nj}] + [a_{nlj}][C_{nl}^{(a)}] \quad (6\text{-}33)$$

$$[E_\varphi^I] = [E_\varphi^{II(b)}] \rightarrow [k_{nmj}][A_{nm}] + [l_{nmj}][B_{nm}] = [d_{nj}] + ([c_{nlj}][C_{nl}^{(a)}] + [e_{nlj}][D_{nl}^{(a)}]) \quad (6\text{-}34)$$

$$[H_z^I] = [H_z^{II(a)}] \rightarrow [i_{nmj}][B_{nm}] = [f_{nj}] + [g_{nlj}][D_{nl}^{(a)}] \quad (6\text{-}35)$$

$$[H_\varphi^I] = [H_\varphi^{II(a)}] \rightarrow [o_{nmj}][A_{nm}] + [p_{nmj}][B_{nm}] = [m_{nj}] + ([n_{nlj}][C_{nl}^{(a)}] + [q_{nlj}][D_{nl}^{(a)}]) \quad (6\text{-}36)$$

For boundary 'b':

$$0 = [E_z^{II(b)}] \rightarrow 0 = [b_{nj}] + [a_{nlj}][C_{nl}^{(b)}] \quad (6\text{-}37)$$

$$0 = [E_\varphi^{II(b)}] \rightarrow 0 = [d_{nj}] + ([c_{nlj}][C_{nl}^{(b)}] + [e_{nlj}][D_{nl}^{(b)}]) \quad (6\text{-}38)$$

A partially expanded matrix form provides a good aid of the matrix form which is represented here as well. Subscripts within the matrices will be further observable and the matrix shape made obvious. Though, a few notes on the sub-subscripts need to be made. A sub-subscript of 1 on the subscript "n", "m", and "l" signifies the first increment of that index. A sub-subscript of "n_max", m"_max", or "l_max" signifies the last increment of that index, which are on "n", "m", or "l" respectively.

For boundary 'a':

$$[E_z^I] = [E_z^{II(a)}] \rightarrow \begin{bmatrix} h_{n_1 m_1 j_1} & \cdots & h_{n_{n\_max} m_{m\_max} j_1} \\ \vdots & \ddots & \vdots \\ h_{n_1 m_1 j_{j\_max}} & \cdots & h_{n_{n\_max} m_{m\_max} j_{j\_max}} \end{bmatrix} \begin{bmatrix} A_{n_1 m_1} \\ \vdots \\ A_{n_{n\_max} m_{m\_max}} \end{bmatrix} = \begin{bmatrix} b_{n j_1} \\ \vdots \\ b_{n j_{j\_max}} \end{bmatrix} + \begin{bmatrix} a_{j_1 n_1 l_1} & \cdots & a_{j_1 n_{n\_max} l_{l\_max}} \\ \vdots & \ddots & \vdots \\ a_{n_1 l_1 j_{j\_max}} & \cdots & a_{n_{n\_max} l_{l\_max} j_{j\_max}} \end{bmatrix} \begin{bmatrix} C_{n_1 l_1}^{(b)} \\ \vdots \\ C_{n_{n\_max} l_{l\_max}}^{(b)} \end{bmatrix} \quad (6\text{-}39)$$

$$[E_\varphi^I] = [E_\varphi^{II(a)}] \rightarrow \begin{bmatrix} k_{n_1 m_1 j_1} & \cdots & k_{j_1 n_{n\_max} m_{m\_max} j_1} \\ \vdots & \ddots & \vdots \\ k_{n_1 m_1 j_{j\_max}} & \cdots & k_{n_{n\_max} m_{m\_max} j_{j\_max}} \end{bmatrix} \begin{bmatrix} A_{n_1 m_1} \\ \vdots \\ A_{n_{n\_max} m_{m\_max}} \end{bmatrix} + \begin{bmatrix} l_{n_1 m_1 j_1} & \cdots & l_{n_{n\_max} m_{m\_max} j_1} \\ \vdots & \ddots & \vdots \\ l_{n_1 m_1 j_{j\_max}} & \cdots & l_{n_{n\_max} m_{m\_max} j_{j\_max}} \end{bmatrix} \begin{bmatrix} B_{n_1 m_1} \\ \vdots \\ B_{n_{n\_max} m_{m\_max}} \end{bmatrix} = \begin{bmatrix} d_{n j_1} \\ \vdots \\ d_{n j_{j\_max}} \end{bmatrix} + \left(\begin{bmatrix} c_{n_1 l_1 j_1} & \cdots & c_{n_{n\_max} l_{l\_max} j_1} \\ \vdots & \ddots & \vdots \\ c_{n_1 l_1 j_{j\_max}} & \cdots & c_{n_{n\_max} l_{l\_max} j_{j\_max}} \end{bmatrix} \begin{bmatrix} C_{n_1 l_1}^{(a)} \\ \vdots \\ C_{n_{n\_max} l_{l\_max}}^{(a)} \end{bmatrix} + \begin{bmatrix} e_{n_1 l_1 j_1} & \cdots & e_{n_{n\_max} l_{l\_max} j_1} \\ \vdots & \ddots & \vdots \\ e_{n_1 l_1 j_{j\_max}} & \cdots & e_{n_{n\_max} l_{l\_max} j_{j\_max}} \end{bmatrix} \begin{bmatrix} D_{n_1 l_1}^{(a)} \\ \vdots \\ D_{n_{n\_max} l_{l\_max}}^{(a)} \end{bmatrix} \right) \quad (6\text{-}40)$$

$$[H_z^I] = [H_z^{II(a)}] \rightarrow \begin{bmatrix} i_{n_1 m_1 j_1} & \cdots & i_{n_{n\_max} m_{m\_max} j_1} \\ \vdots & \ddots & \vdots \\ i_{n_1 m_1 j_{j\_max}} & \cdots & i_{n_{n\_max} m_{m\_max} j_{j\_max}} \end{bmatrix} \begin{bmatrix} B_{n_1 m_1} \\ \vdots \\ B_{n_{n\_max} m_{m\_max}} \end{bmatrix} = \begin{bmatrix} f_{n j_1} \\ \vdots \\ f_{n j_{j\_max}} \end{bmatrix} + \begin{bmatrix} g_{j_1 n_1 l_1} & \cdots & g_{n_{n\_max} l_{l\_max} j_1} \\ \vdots & \ddots & \vdots \\ g_{n_1 l_1 j_{j\_max}} & \cdots & g_{n_{n\_max} l_{l\_max} j_{j\_max}} \end{bmatrix} \begin{bmatrix} D_{n_1 l_1}^{(a)} \\ \vdots \\ D_{n_{n\_max} l_{l\_max}}^{(a)} \end{bmatrix} \quad (6\text{-}41)$$

$$\left[H_\varphi^a\right] = \left[H_\varphi^{II(a)}\right] \rightarrow \begin{bmatrix} o_{n_1 l_1 j_1} & \cdots & o_{n_{n\_max} l_{l\_max} j_1} \\ \vdots & \ddots & \vdots \\ o_{n_1 l_1 j_{j\_max}} & \cdots & o_{n_{n\_max} l_{l\_max} j_{j\_max}} \end{bmatrix} \begin{bmatrix} A_{n_1 m_1} \\ \vdots \\ A_{n_{n\_max} m_{m\_max}} \end{bmatrix} + \begin{bmatrix} p_{n_1 l_1 j_1} & \cdots & p_{n_{n\_max} l_{l\_max} j_1} \\ \vdots & \ddots & \vdots \\ p_{n_1 l_1 j_{j\_max}} & \cdots & p_{n_{n\_max} l_{l\_max} j_{j\_max}} \end{bmatrix} \begin{bmatrix} B_{n_1 m_1} \\ \vdots \\ B_{n_{n\_max} m_{m\_max}} \end{bmatrix} = \begin{bmatrix} m_{n j_1} \\ \vdots \\ m_{n j_{j\_max}} \end{bmatrix} + \left(\begin{bmatrix} n_{n_1 l_1 j_1} & \cdots & n_{n_{n\_max} l_{l\_max} j_1} \\ \vdots & \ddots & \vdots \\ n_{n_1 l_1 j_{j\_max}} & \cdots & n_{n_{n\_max} l_{l\_max} j_{j\_max}} \end{bmatrix} \begin{bmatrix} C_{n_1 l_1}^{(a)} \\ \vdots \\ C_{n_{n\_max} l_{l\_max}}^{(a)} \end{bmatrix} + \begin{bmatrix} q_{n_1 l_1 j_1} & \cdots & q_{n_{n\_max} l_{l\_max} j_1} \\ \vdots & \ddots & \vdots \\ q_{n_1 l_1 j_{j\_max}} & \cdots & q_{n_{n\_max} l_{l\_max} j_{j\_max}} \end{bmatrix} \begin{bmatrix} D_{n_1 l_1}^{(a)} \\ \vdots \\ D_{n_{n\_max} l_{l\_max}}^{(a)} \end{bmatrix} \right) \quad (6\text{-}42)$$

For boundary ‘b’:

$$0 = \left[E_z^{II(b)}\right] \rightarrow 0 = \begin{bmatrix} b_{n j_1} \\ \vdots \\ b_{n j_{j\_max}} \end{bmatrix} + \begin{bmatrix} a_{n_1 l_1 j_1} & \cdots & a_{n_{n\_max} l_{l\_max} j_1} \\ \vdots & \ddots & \vdots \\ a_{n_1 l_1 j_{j\_max}} & \cdots & a_{n_{n\_max} l_{l\_max} j_{j\_max}} \end{bmatrix} \begin{bmatrix} C_{n_1 l_1}^{(b)} \\ \vdots \\ C_{n_{n\_max} l_{l\_max}}^{(b)} \end{bmatrix} \quad (6\text{-}43)$$

$$0 = \left[E_\varphi^{II(b)}\right] \rightarrow 0 = \begin{bmatrix} d_{n j_1} \\ \vdots \\ d_{n j_{j\_max}} \end{bmatrix} + \left(\begin{bmatrix} c_{n_1 l_1 j_1} & \cdots & c_{n_{n\_max} l_{l\_max} j_1} \\ \vdots & \ddots & \vdots \\ c_{n_1 l_1 j_{j\_max}} & \cdots & c_{n_{n\_max} l_{l\_max} j_{j\_max}} \end{bmatrix} \begin{bmatrix} C_{n_1 l_1}^{(b)} \\ \vdots \\ C_{n_{n\_max} l_{l\_max}}^{(b)} \end{bmatrix} + \begin{bmatrix} e_{n_1 l_1 j_1} & \cdots & e_{n_{n\_max} l_{l\_max} j_1} \\ \vdots & \ddots & \vdots \\ e_{n_1 l_1 j_{j\_max}} & \cdots & e_{n_{n\_max} l_{l\_max} j_{j\_max}} \end{bmatrix} \begin{bmatrix} D_{n_1 l_1}^{(b)} \\ \vdots \\ D_{n_{n\_max} l_{l\_max}}^{(b)} \end{bmatrix} \right) \quad (6\text{-}44)$$

6.3 Summation Truncation

It’s also important to note that the summations for all the equations, and therefore the matrices, need to be truncated from an infinite to a finite length indices. This is required prior to numerical computation. Truncation can reduce accuracy, so it follows that smaller truncated values can be less accurate than when truncated to higher values. However, it is a fair trade off between computer computational power and available processing time versus accuracy. That’s due to the fact that the higher “n”, “m”, and “l” indices represent higher orders and wave numbers of waves that either don't propagate because they're evanescent or contribute very little overall amplitude and can be neglected. Also, computing solutions that contribute little to the final solution can be computationally burdensome. A result of this trade ultimately leads to the truncated size selection for the summation ranges and matrix sizes.

An example for finding evanescent modes of higher order "m" can be seen when examining equation (4-32) with $k_{\rho_m} = \sqrt{k^2 - k_{z_m}^2}$. At larger values of "m", it can be seen that $k^2 < k_{z_m}^2$ which makes k_{ρ_m} imaginary, representing evanescent radial wave modes. When $k = \omega\sqrt{\mu_I \varepsilon_I}$ and (4-32) are substituted into $k^2 > k_{z_m}^2$, where k_{ρ_m} is still real, the relationship

$$\omega^2 \mu_I \varepsilon_I > \frac{m^2 \pi^2}{a^2} \quad \rightarrow \quad \frac{a^2 \omega^2 \mu_I \varepsilon_I}{\pi^2} > m^2 \quad \rightarrow \quad \frac{a * \omega\sqrt{\mu_I \varepsilon_I}}{\pi} > \pm m \qquad (6\text{-}45)$$

can be established, setting the upper limit for "m". The same can be done for k_{ρ_l} and k_{z_l}. Following the same approach but using equations $k = \omega\sqrt{\mu_{II} \varepsilon_{II}}$ and (5-9), the relationship

$$\omega^2 \mu_{II} \varepsilon_{II} > \left(k \cos\theta^i + \frac{2\pi l}{(a+b)}\right)^2 \quad \rightarrow \quad \omega\sqrt{\mu_{II} \varepsilon_{II}} > k \cos\theta^i + \frac{2\pi l}{(a+b)} \quad \rightarrow \qquad (6\text{-}46)$$

$$\frac{(a+b)(\omega\sqrt{\mu_{II} \varepsilon_{II}} - k \cos\theta^i)}{2\pi} > l \quad \rightarrow \quad \frac{(a+b)k(1 - \cos\theta^i)}{2\pi} > l$$

will provide an upper limit for "l". These equations are implemented in the code, which will be discussed in the following sections, as checks for the user to compare his or her inputs. Violating these limits can prove challenging for the code to find solutions to the field equations, producing nonsensical, erroneous results with values of orders of magnitudes 10's to 100's of times greater than expected.

In summary, the “m” and “l” max values are selected in order to restrict the values of both k_{ρ_l} and k_{ρ_m} to the real domain, which also drives the size of the matrices, for the following reasons:

- Imaginary of k_{ρ_l} and k_{ρ_m} produce rapidly decaying evanescent fields that contribute little to the overall amplitude
- Computationally it can be burdensome to compute and in some cases challenging to find a solution.

It’s also important to note here that this restriction is acceptable to the author. In any approximation technique, there will be areas where computations are truncated and simplification assumptions are made. The results and comparisons in Chapter 7 will provide a guide for when the approach presented by this paper is most valid, with respect to the dimensions of the corrugated cylinder structure when compared to the wavelength of the incident plane-wave. It will be left for a future endeavor to stress the model approach presented in this research, to determine the exact limitations of specific assumptions (e.g. under what conditions can the restricting of k_{ρ_l} and k_{ρ_m} being real be considered unacceptable).

6.4 Numeric Computation Techniques and Tool Methodology

6.4.1 Tool Selection and Use

This next stage requires that of all the equations get implemented into a computational tool. There are many computational tools to choose from, however this author chose to use Mathematica® [17] based on its ability to handle math symbolically. Also, another significant driver for the selection of Mathematica® is the availability of a free toolbox add-on to Mathematica®, available online from UC San Diego known as NCAlgebra [18], which can handle noncommutative algebra.

NCAlgebra makes it easy in manipulating matrices and vectors algebraically and symbolically. Its key use in this paper was to solve for each of the unknown expansion coefficients in terms of known terms, using the matching equations from (6-33) to (6-38). Essentially, a solve function native to the tool is used, which, once that is done, numeric values for each of terms are computed, plugged in, computed again and the coefficients would be solved for.

However, there were some convergence issues in the results. The issue was not fully identified but at this point, it is important to mention that when NCAlgebra produces solutions, it checks and validates it. This validation check produced a warning describing that the solution may be prone to error for certain circumstances, not described, since it could not guarantee the solution's accuracy. Based on the results and the NCAlgebra warning, the fully derived NCAlgebra method was commented out of the code. It was replaced by a loop solve algorithm developed for this research. However, NCAlgebra was still used to derive some of the unknown expansion coefficient equations where the solution check was validated, and were also used to spot check solutions from the loop solve algorithm used to replace it.

The loop solve algorithm was used to solve partially solved unknown coefficients. The solve function in Mathematica® was used within a loop to solve for the large matrix system of equations. As the loop solve algorithm increments, it solves an equation in terms of an unknown, then plugs that unknown into the equation of the next loop solve iteration. This process eliminates an unknown with each step. At the last step, the unknown is found to have a value. At this point, another loop is implemented working backwards and plugging in values until all the unknowns are known.

During the numerical computational effort, there were issues that arose in solving for the unknown coefficients. They were attributed to ill-conditioned matrices which will be described in the next section.

6.4.2 Ill-Conditioned Matrices

It turned out that during the computation iterations in the process of this work, the computational tool would produce errors indicating ill-conditioned matrices. This was found during the inversion process of many of the matrices.

There has been much work behind ill-conditioned matrices, particularly in such a case as this one, where an ill-conditioned system of linear equations is involved in an engineering problem. The driving source is from an ill-posed problem, which does not necessarily stem from an ill-conceived design, but rather a fundamental physical limitation to the data at hand [19]. Described in such work, are suggestions for the use of the Moore–Penrose pseudoinverse, which relates to the least squares regression method in finding the shortest length solution to a problem [20]. Both the pseudoinverse and least squares method are made use of in the code, in order to eliminate the ill-conditioned errors and solve for the unknown expansion coefficients.

6.4.3 Tool Methodology

The following section will describe the high-level workflow on how the computational tools are used. It will show how the tools mentioned in section 6.4.1 are used in conjunction with the solving methods mentioned in section 6.4.2 . The workflow below will describe the code in an already existing state and therefore the workflow steps describe the sequential flow of data, and not the creation of code.

Step 1: The boundary ‘b’ equations, (6-37) and (6-38), are used to solve for $C_{nl}^{(b)}$ and $D_{nl}^{(b)}$ using the least squares method with tool’s built-in least squares solver function. Equations were rearranged to solve for the specified expansion coefficient per the least squares solver function required arrangement [21]. See Figure 6-3 for pictorial form.

Step 2: The boundary ‘a’ equations (6-33) and (6-35) are used to solve for A_{nm} and B_{nm}. They are rearranged using manual matrix manipulation method with noncommutative algebra and solve for A_{nm} and B_{nm} in terms of known of field components, $C_{nl}^{(a)}$ and $D_{nl}^{(a)}$

Step 3: The boundary ‘a’ equation (6-34) is now used to solve for $C_{nl}^{(a)}$ in terms of $D_{nl}^{(a)}$ using the same method as in step 2.

Step 4: The new form of $C_{nl}^{(a)}$ (in terms of $D_{nl}^{(a)}$) is plugged into the new form of A_{nm} (in terms of $C_{nl}^{(a)}$) in order to put A_{nm} in terms of $D_{nl}^{(a)}$

Step 5: The boundary ‘a’ equation (6-36) is now used to solve for $D_{nl}^{(a)}$ in terms of known of field components, A_{nm}, and B_{nm}, where A_{nm}, and B_{nm} are substituted by the solutions in the previous step.

Step 6: $D_{nl}^{(a)}$ is now expanded and solved for all the expanded coefficients within the loop algorithm referenced earlier as the loop solve algorithm. Once complete, $D_{nl}^{(a)}$ is fully solved for numerically.

Step 7: Dnl(a) is plugged into $C_{nl}^{(a)}$ and solved for. Both of which are now plugged into A_{nm} and B_{nm}, which are solved for as well, completing the numerical solving of the expansion coefficients. Steps 2 – 7 can be seen in Figure 6-4.

A pictorial overview of this workflow can be seen in Figure 6-2. It includes additional steps required for configuration and result generation.

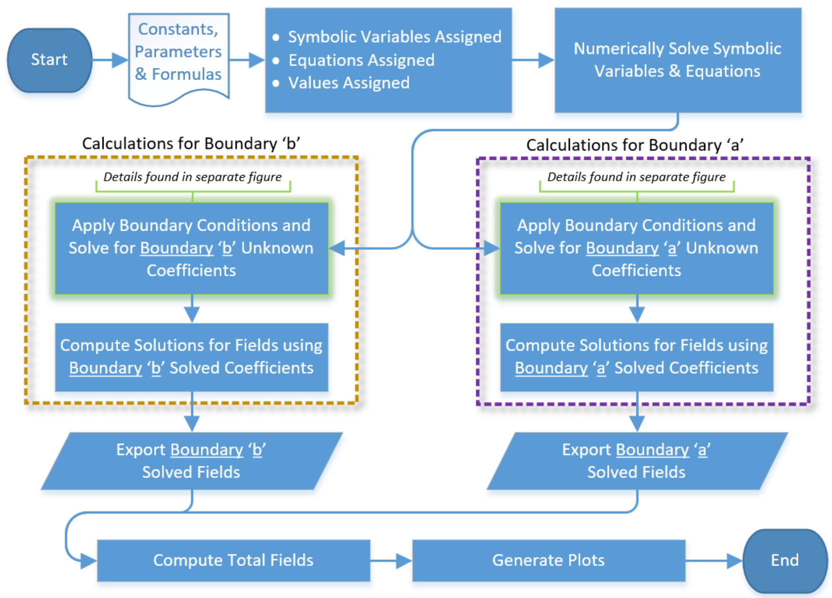


Figure 6-2 Software Process Workflow

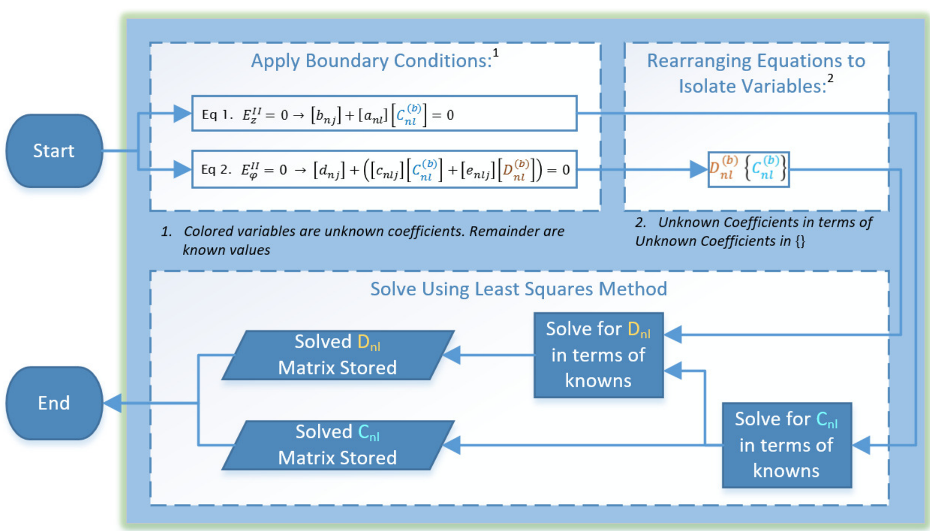


Figure 6-3 Solving for Boundary 'b' Unknown Coefficients Flow Chart

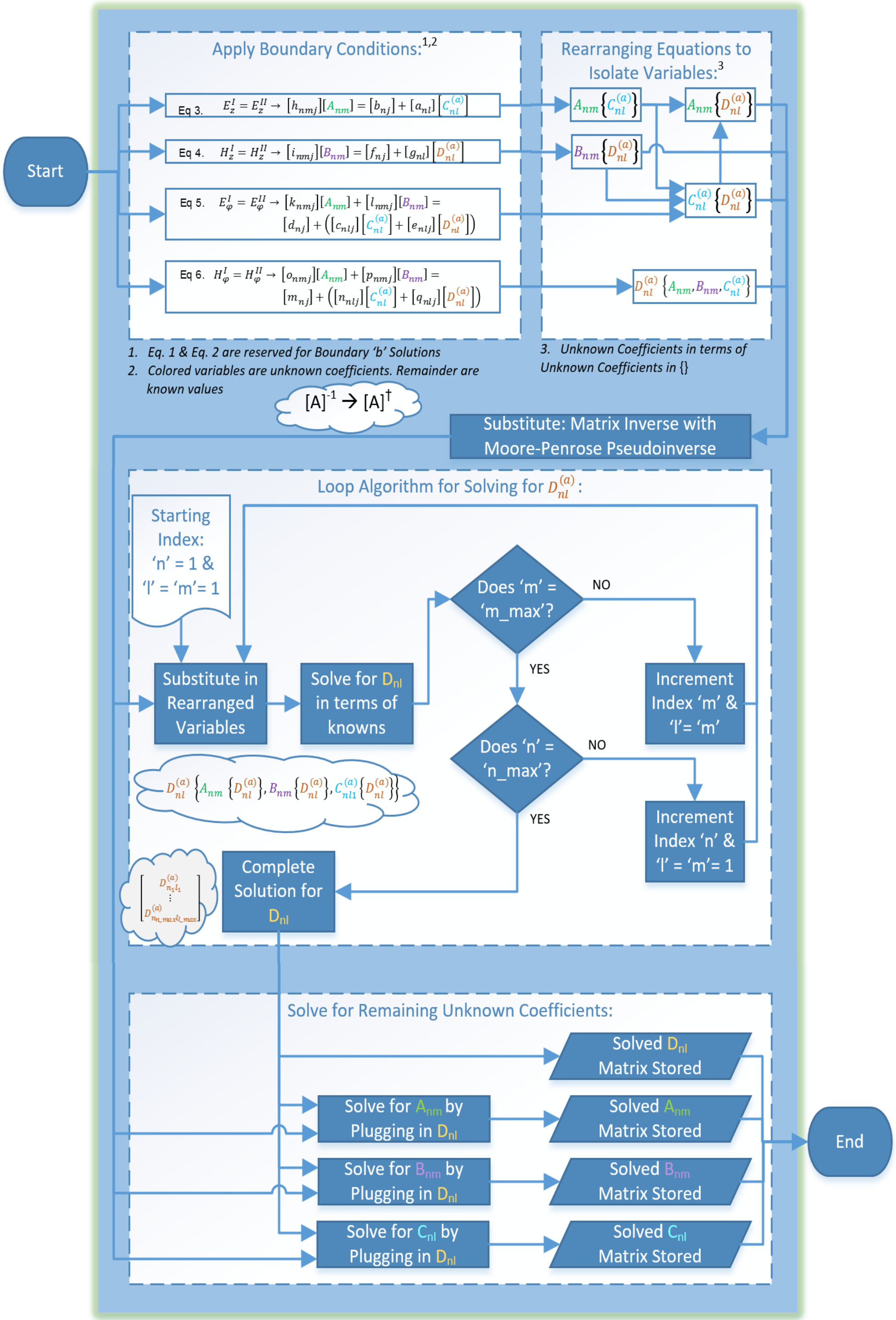


Figure 6-4 Solving for Boundary 'a' Unknown Coefficients Flow Chart

CHAPTER 7 RESULTS, COMPARISONS AND FUTURE RESEARCH

7.1 Model Configuration and Parameters

Many configurations and parameters were set up to support the research in this paper. This section will attempt to describe the key configurations and parameters in order to facilitate understanding of the results. Note, this model is composed of multiple smaller models, most centered on the periodic corrugated cylinder. However, there's also a smooth cylinder model incorporated, which will be discussed in section 7.3 .

7.1.1 Parameters Relative to Lambda

The model computes and displays many of the results in terms of the wavelength λ, where $\lambda = \frac{c}{f}$, where 'f' is the frequency of the incident field and 'c' is the speed of light. Also, the dimensions of the corrugated cylinder are described in units of λ, which include the corrugation opening denoted by 'a', thickness of the metallic corrugation portion denoted by 'b, the inner corrugation radius ρ_1 and the exterior corrugation radius ρ_2. Note that the period of the corrugation is equal to 'a + b'. This includes the plots displayed later in this chapter, which vary in ρ and have an axis in units of λ, displayed as ρ/λ. This allows the model to display results that are independent of a specified frequency/wavelength. However, parameters do have values thus a λ is chosen and can be varied while fixing the geometry, if so desired. For the purpose of this paper, λ is fixed and the geometry and other parameters are varied.

In the results section, multiple scenarios were run with different parameter changes. One such parameter was labeled 'd' which signifies a relative 'dimension' to λ, which the physical dimension parameters of the corrugated and smooth cylinders are linked to. For example, the dimension 'b', the thickness of the protruding part of the corrugation, is always set equal to 'd' in all the scenarios, unless explicitly stated otherwise. The parameter 'd' allows for the description of 3 different groupings of relative physical dimensions which are $d \gg \lambda$ by setting $d = 20\lambda$, $d \approx \lambda$ by setting $d = 2\lambda$, and $d \ll \lambda$ by setting $d = 0.1\lambda$. This was done in order to more clearly gauge the performance of the model relative to the different scattering regimes laid out by the figure, as per described in [22]. Also, according to Fuhs [23, p. 18], when in the Rayleigh scattering region ($d \gg \lambda$), polarization is not important for the magnitude of RCS.

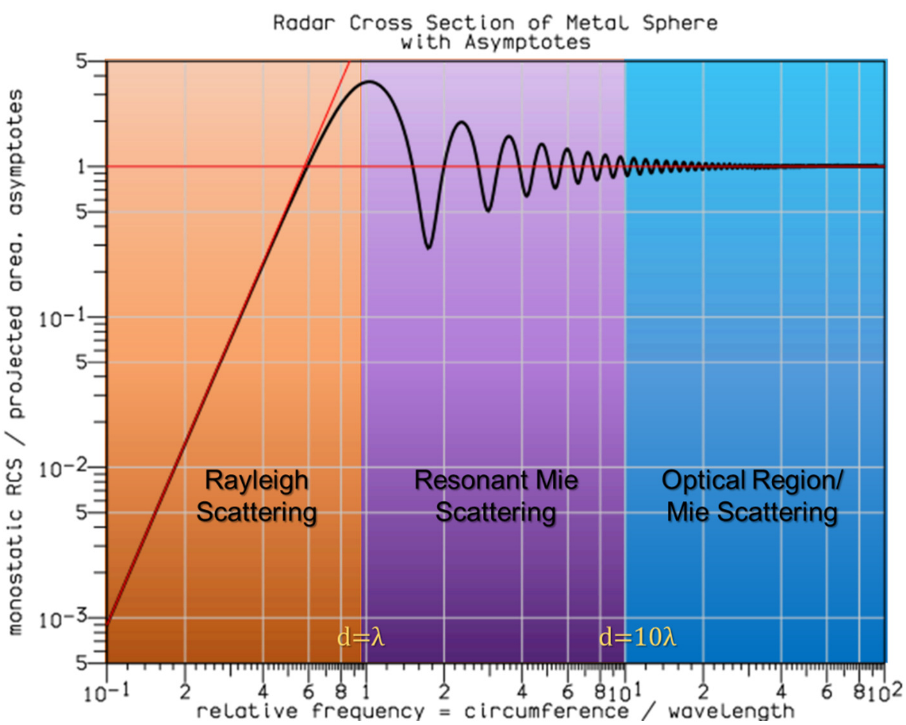


Figure 7-1 Relative relationship of target size to illuminated wavelength with associated regions of scattering approximation, courtesy of Wikipedia [24]

7.1.2 Incident Field Parameters

The incident field amplitudes are driven by the E_0 value for the TM_z mode and the H_0 value for the TE_z mode. Since these modes are independent of each other, E_0 and H_0 values can be selected independent of each other. It is important to note at this point, that the H-fields amplitudes in the TM_z mode are driven by the E_0, where the H-field amplitude is equal to $\frac{E_0}{Z}$, where $Z = \sqrt{\frac{\mu_{II}}{\varepsilon_{II}}}$ and is the impedance of the medium in which the incident field is in. The same holds true for the E-fields in the TE_z mode, in that it's dependent on the H_0 value, where the E-field amplitude is equal to ZH_0.

7.1.3 Point Matching Selection

As discussed in Chapter 6, point matching is required for the numerical solution of the unknown expansion coefficients. There are various points that could be selected with the prescribed range for that boundary condition, with ρ always being equal to ρ_2. The type and quantity of matching points can worsen or improve the numerical results, though a minimum of 'j' quantity, expansion number of the coefficients, is required.

Different matching point techniques were attempted during the development of the model. When the NCAlgebra method was first utilized, it allowed for an overdetermined solution where there were more equations (or 'j' points), than unknowns, (or expansion count of coefficients). That's because it was fully defined in the pseudoinverse and least squares method. However, due to its limitations as described in section 6.4.1 , the NCAlgebra method was abandoned and the new technique did not allow for an overdetermined solution in its current form.

Two main matching point techniques were maintained in the model. One was fixing φ to an arbitrary φ value from 0 to 2π while varying z 'j' times, evenly spaced spanning the distance of the boundary, being ‘a’ or ‘b’ depending on that boundary. The other was to fix z to an arbitrary z value within the boundary, while varying φ ‘j’ times evenly from 0 to 2π.

The results computed were done fixing z and varying φ. This produces more accurate results, which is due to the fact that the computed comparison results did not vary z values, but did vary φ in the changing φ results. Therefore, the selection in matching points that is best suited is the matching points that would reinforce the intended computational analysis, which was to vary φ.

7.1.4 Total E-Field Calculation

The total E-field calculation finds the magnitude of the combined fields. The method is fairly straight forward. The fields are converted from cylindrical coordinates to rectangular coordinates, which can be found in [12, p. 923] to form

$$E_x = E_\rho \cos\phi + E_\phi \sin\phi \tag{7-1}$$

$$E_Y = E_\rho \sin\phi + E_\phi \cos\phi \tag{7-2}$$

$$E_Z = E_Z \tag{7-3}$$

Then, each of the rectangular coordinate forms of fields are squared, summed together and then the square root of that number is attained.

$$\sqrt{{E_x}^2 + {E_Y}^2 + {E_Z}^2} = E_{Total} \tag{7-4}$$

This is done for every data point to generate the total E-field data.

7.1.5 Reconciliation of Boundary ‘a’ and Boundary ‘b’

Up to this point, the equations for boundary ‘a’ and boundary ‘b’ have been given separate treatment. However, results were computed for the separate field equations and results plotted and incorporated into the results section. At the near field, where $\rho \leq \lambda$, the boundaries can be treated separately.

For the far field, $\rho >> \lambda$, boundary ‘a’ and boundary ‘b’ solutions are examined for their accuracy. Either technique alone should produce a gross representation of the scatter, though within certain regions the approximations of the field amplitudes can be poor. However, the hybrid nature of the problem formulation allows for the depolarization representation of the periodic corrugated cylinder.

A technique described by Kishk et al [25], the asymptotic boundary condition method, utilizes coefficients that vary with the z axis, and are weighted by a ratio factor w/p, where ‘w’ is the dimension of the corrugation opening and ‘p’ is the corrugation period. This is done at the boundary condition to develop a complete field solution, where more detail can be found in [26]. This technique and variations thereof were investigated but not implemented.

In the present work, the field solutions for boundary ‘a’ is added with that of boundary ‘b’. This superposition of fields, produces the results shown for the runs labeled “Run a_plus_b…”. The results are compared to that of a smooth cylinder as well as to alternate methods for computing the scattered field of a corrugated cylinder [27]. This treatment of superposition of fields is done in [28], where the field of a cylinder without accounting for the corrugated perturbations, represented as metallic rings, and then added to the scattering due to the metallic rings.

7.2 RCS Computation

The model produces various Radar Cross Section (RCS) plots, as this is a typical convention in the electromagnetics scattering community, to display and compare scattered fields. The RCS symbol, σ, is the designated symbol to represent the RCS value, which is in units of m^2. The formula representing σ is derived from the free-space loss factor cause by the spherical spreading of a propagating planewave, which a full description of its derivation can be found in [29, p. 96]. This model uses σ in the form

$$\sigma = \lim_{R\to\infty}\left[4\pi R^2 \frac{|E^s|^2}{|E^i|^2}\right] \tag{7-5}$$

where 'R' is the distance of observation from the target, meters. The value 'R' needs to be large enough where R>> than the largest physical dimension of the target and wavelength so that the propagating planewave representation holds and the formula remains valid. The RCS, σ, can also be calculated in decibels, dB, in order to display a large range of data. The units are in decibels per square meter or dBsm. This is calculated as

$$\sigma_{dBsm} = \lim_{R\to\infty}\left[10log_{10}\, 4\pi R^2 \frac{|E^s|^2}{|E^i|^2}\right]. \tag{7-6}$$

It is important to take note of the aspect angles, φ, that are used in the σ calculations, as the computed results can vary. There is the φ of the incidence field, φ_i, and the φ of the observed or of the scattered field, φ_s. In each of the variations, φ_s values are typically swept through from 0 to 2π. The following are the different methods for varying the φ values in order to get the different σ results:

- *RCS φ Sweep Method 1:* φ_s is swept through 0 to 2π and φ_i is kept at a fixed constant value between 0 to 2π.
- *RCS φ Sweep Method 2:* φ_s is swept through 0 to 2π and $\varphi_i = \varphi_s$. This is typically referred to as a monostatic RCS since it assumes the transmitter and receiver, for typical radar applications, are at the same location.

- *RCS φ Sweep Method 3:* φ_s is swept through 0 to 2π and $\varphi_i = \varphi_s + \varphi_{offset}$, where φ_{offset} is an offset value of φ. This is typically referred to as a bistatic RCS since it assumes the transmitter and receiver, for typical radar applications, are at separate locations.

The technique used in this paper is RCS φ sweep method 1.

7.3 Smooth Cylinder Comparison Model

In order to have a comparison model for the periodic corrugated cylinder, a smooth cylinder was modeled. The dimensions of the corrugated cylinder can be adjusted to approximate a smooth cylinder and then compared to the smooth cylinder model.

A smooth cylinder model of oblique incidence planewave is found in [12, pp. 614-624] . This textbook model provided the basis for the TM_z mode scattered equations

$$E_\rho^{s\,(TM)} = jE_0 \cos\theta^i \, e^{+jk\cdot z\cdot \cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} \frac{J_n(k\rho_2 \sin\theta^i)}{H_n^{(2)}(k\rho_2 \sin\theta^i)} \left(\frac{H_{n-1}^{(2)}(k\rho \sin\theta^i) - H_{n+1}^{(2)}(k\rho \sin\theta^i)}{2} \right) e^{jn\varphi} \quad (7\text{-}7)$$

$$E_\varphi^{s\,(TM)} = jE_0 \frac{\cot\theta^i}{k\rho} e^{+jk\cdot z\cdot \cos\theta^i} \sum_{n=-\infty}^{\infty} n j^{-n+1} \frac{J_n(k\rho_2 \sin\theta^i)}{H_n^{(2)}(k\rho_2 \sin\theta^i)} H_n^{(2)}(k\rho \sin\theta^i) e^{jn\varphi} \quad (7\text{-}8)$$

$$E_z^{s\,(TM)} = E_0 \sin\theta^i \, e^{+jk\cdot z\cdot \cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} \frac{J_n(k\rho_2 \sin\theta^i)}{H_n^{(2)}(k\rho_2 \sin\theta^i)} H_n^{(2)}(k\rho \sin\theta^i) e^{jn\varphi} \quad (7\text{-}9)$$

and the TE_z mode scattered equations

$$E_\rho^{s\,(TE)} = -j \frac{H_0}{\omega\varepsilon\rho} \frac{e^{+jk\cdot z\cdot \cos\theta^i}}{\sin\theta^i} \sum_{n=-\infty}^{\infty} n j^{-n+1} \frac{J_{n-1}(k\rho_2 \sin\theta^i) - J_{n+1}(k\rho_2 \sin\theta^i)}{H_{n-1}^{(2)}(k\rho_2 \sin\theta^i) - H_{n+1}^{(2)}(k\rho_2 \sin\theta^i)} H_n^{(2)}(k\rho \sin\theta^i) e^{jn\varphi} \quad (7\text{-}10)$$

$$E_\varphi^{s\,(TE)} = jH_0 \sqrt{\frac{\mu}{\varepsilon}} e^{+jk\cdot z\cdot \cos\theta^i} \sum_{n=-\infty}^{\infty} j^{-n} \frac{J_{n-1}(k\rho_2 \sin\theta^i) - J_{n+1}(k\rho_2 \sin\theta^i)}{H_{n-1}^{(2)}(k\rho_2 \sin\theta^i) - H_{n+1}^{(2)}(k\rho_2 \sin\theta^i)} \left(\frac{H_{n-1}^{(2)}(k\rho \sin\theta^i) - H_{n+1}^{(2)}(k\rho \sin\theta^i)}{2} \right) e^{jn\varphi} \quad (7\text{-}11)$$

$$E_z^{s\,(TE)} = 0\,. \quad (7\text{-}12)$$

The Hankel and Bessel order ‘n’ is set to the same ‘n’ as used in the corrugated cylinder model. The radius of the smooth cylinder is ρ_2, the same dimension as the larger radius of the corrugated cylinder.

Note that only the electric field components were used. Using only the electric fields provides an adequate comparison, as there is no polarization and therefore no need to fully represent a system of each TM_z and TE_z mode as in the corrugated case. Each mode will have the fields calculated separately and then, through the principle of superposition, combined with each other on certain plots in the following section. Also, the incident field is the same as that incident onto the corrugated cylinder in the model described in Chapter 3.

The smooth cylinder results are computed and displayed alongside the periodic corrugated cylinder results. Since the periodic corrugated cylinder is of a hybrid mode, both TM_z and TE_z modes exist within the periodic corrugated cylinder scattered field. However, the data displayed for the smooth cylinder is made available to show the separate TM_z and TE_z modes, as well as the superposition combined modes. The data sets were created for the different cylindrical coordinate axes.

A subset of the data is collected and analytically compared between periodic corrugated cylinder and the smooth cylinder, in order to calculate the mean of percent error between them. This data set is composed of the superposition combined modes, in RCS dBsm, for each of the cylindrical coordinate axes and the total field. The data points at each φ, for the smooth and corrugated cylinders, are subtracted from each other and the absolute value of that is divided by the smooth cylinder dBsm value, and finally multiplied by 100% to give the percent error, as in equation (7-13).

$$\frac{|\sigma_{dBsm\,[smooth]} - \sigma_{dBsm\,[corr]}|}{\sigma_{dBsm\,[smooth]}} \cdot 100\% = \%\ error\ of\ \sigma_{dBsm}\ . \qquad (7\text{-}13)$$

Then, the mean of the percent error is found for each of the data subsets.

7.4 Alternate Corrugated Cylinder Method Comparison

In order to fully validate the results of the periodic corrugated cylinder scattered field, a comparison to an alternate method for the same geometry is merited. The chosen comparison method was from A. Freni et al [27] where results for a Finite Element Method (FEM) and the Method of Moments (MoM) are captured in figure 3 of that paper. The geometry referenced in figure 3 of [27] was utilized and the cross-polar plot was recreated in this paper, by extracting the FEM and MoM data curves, shown in Figure 7-150. Also captured in Figure 7-150 are the results using this research's technique. Data was extracted from figure 3 of [27] using a curve mapping and data extraction tool called WebPlotDigitizer [30]. The data plotted is, $\sigma_{\phi\theta}/\lambda_0$ (dB), which is the cross-polar scatter width derived from

$$\sigma_{\phi\theta} = \lim_{\rho\to\infty}\left[2\pi\rho\frac{|E_\phi^s|^2}{|E_\theta^i|^2}\right] \qquad (7\text{-}14)$$

which the first subscript is the polarization of the scattered field and the second subscript is the polarization of the incident field [28]. A comparison of this technique can also be found in [31] by the author of this research.

7.5 Results

A variety of different runs were executed, as was discussed earlier. Table 1 captures a summary of the boundary 'a' simulation runs with the adjusted parameters identified, on runs comparing the corrugated cylinder to that of the smooth cylinder. Also, boundary 'b' runs were generated (see Table 2), using the same parameters as the subset runs of boundary 'a' shown in this chapter, except with a z position in the boundary ‘b’ region. Table 3 captures a summary of the boundary ‘a’ plus boundary ‘b’ runs, which represent the final solution approach. The different runs captured in the subsequent sections use a title scheme described by the example in Figure 7-1, in order to help identify the configuration for each run and the plots captured in that section.

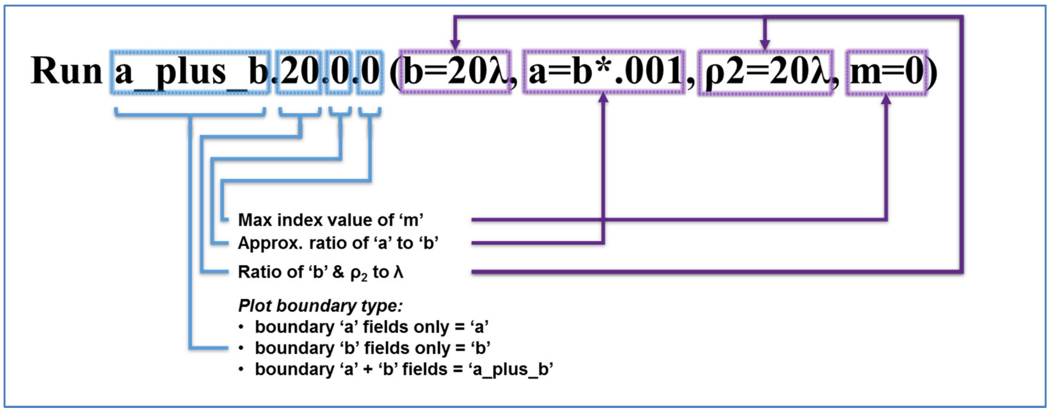


Figure 7-2 A run title example describing each component of the title and how it relates to the run’s configuration

From Table 1, it can be seen that some of the runs have a status of ‘Bad Data/Ill-Condition’. These were mostly runs that had ‘m’ or ‘l’ that were exceeding their max allowable value, allowing k_ρ to become imaginary. As mentioned in section 6.3 , these modes in which k_ρ are imaginary, produce evanescent waves which decay rapidly when moving away from the field source at the cylinder edge, and thus can be ignored.

Another item to note, is that all the runs were kept at n=3. This an acceptable mode or Hankel function order as much of the work with similar scattering geometries have been conducted at n=1 or n= 2, such as [32]. This upper limit was also a computational limitation. Orders above n=3 would not produce a solution with the capability and the time allotted for the computer systems used.

The following subsections provide a subset of the simulation runs collected as part of this research effort. They are pertinent comparative material between the periodic corrugated cylinder model and the comparison model which are used to draw conclusions from. The mean error, as described in section 7.3 are shown for boundary ‘a’(Figure 7-3), boundary ‘b’(Figure 7-4) and boundary ‘a+b’(Figure 7-5).

Each of the presented runs are in their own section with multiple plots. Each section consists of the following:

- 1 detailed summary table of all the φ changing plot parameters,
- 4 Polar Plots of RCS dBsm (Ez, E ρ, E φ and E_{Total} of TM_z +TE_z modes)
- 4 XY Plots of RCS dBsm (Ez, E ρ, E φ and E_{Total} of TM_z +TE_z modes)
- 1 detailed summary table of all the ρ changing plot parameters
- 8 XY field amplitude plots (Absolute value of the fields : Ez, E ρ, E φ and E_{Total} of TM_z +TE_z modes, for scattered field and for scattered + incident field)..

Additional results are also provided here. As mentioned in section 7.4 , an alternate method in representing scattering of a corrugated cylinder is discussed with results in section 7.5.10 . Also, the relative dielectric constant in region I is varied and compared, showing the effects of dielectric loading and lossy dielectric loading, with results in section 7.5.11 .

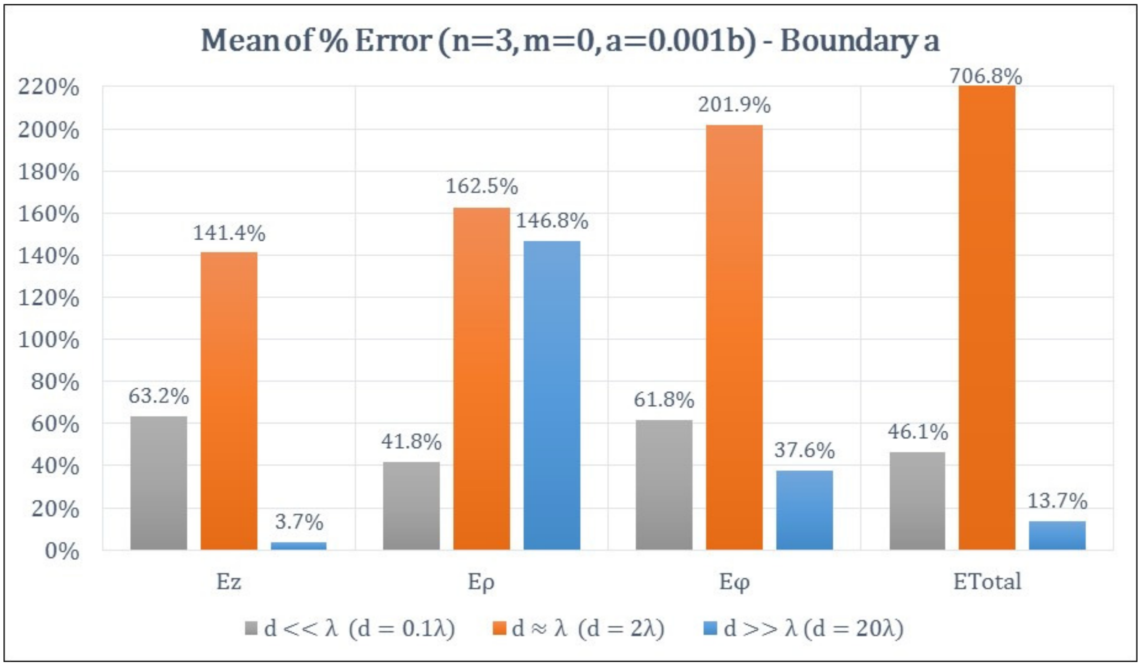


Figure 7-3 Mean of %error between corrugated cylinder and smooth cylinder model, for boundary 'a'

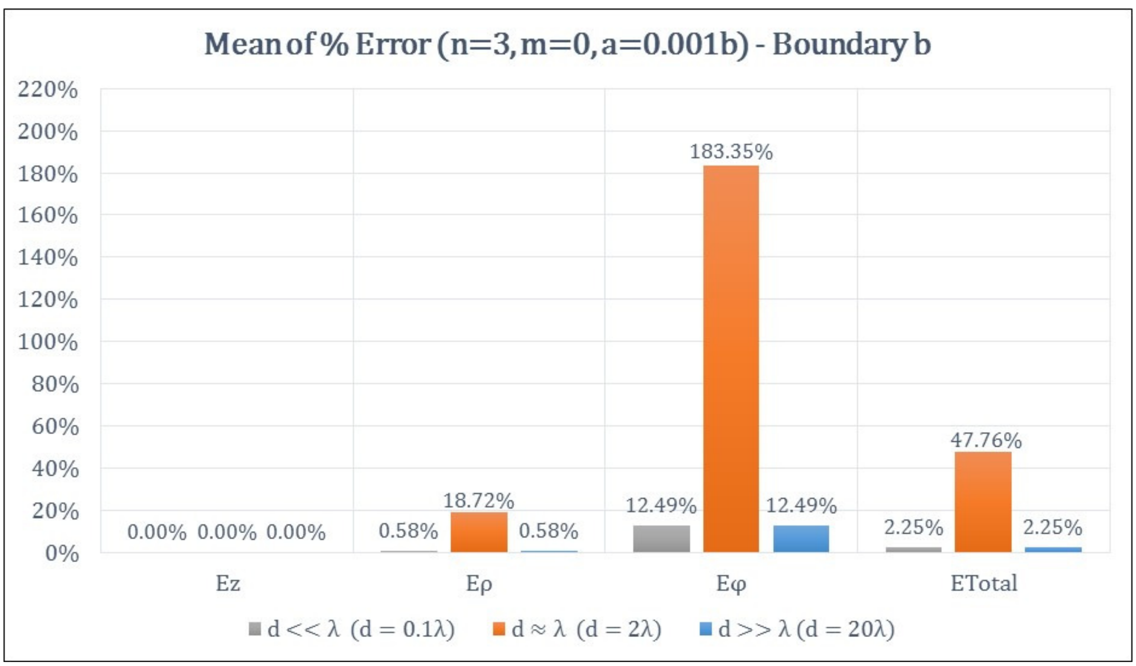


Figure 7-4 Mean of %error between corrugated cylinder and smooth cylinder model, for boundary 'b'

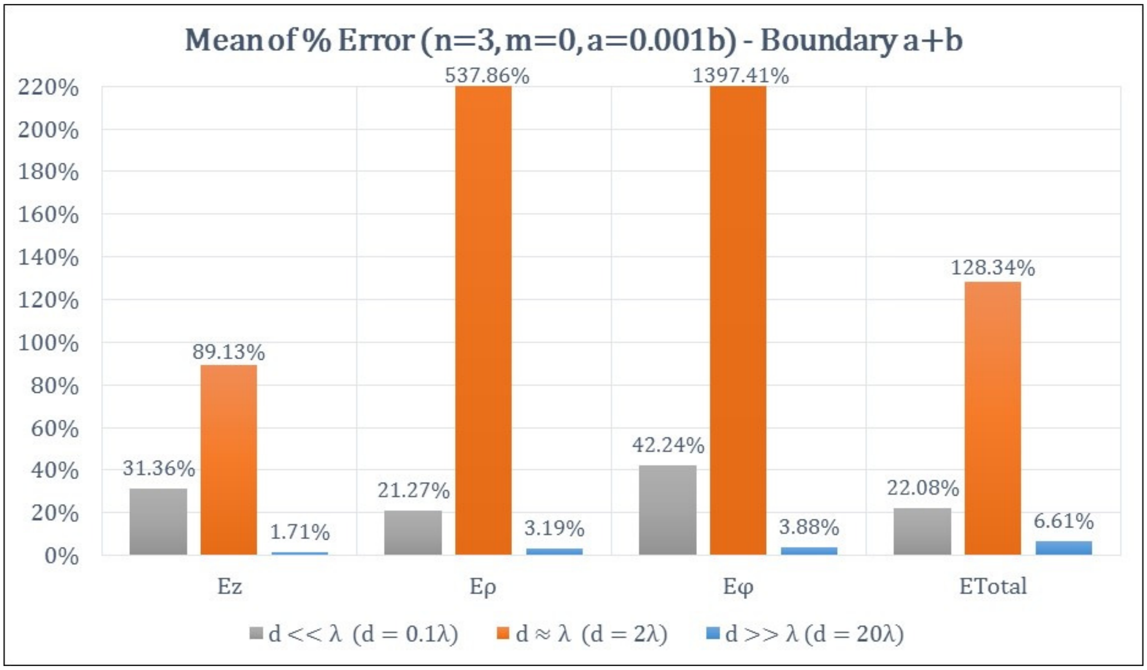


Figure 7-5 Mean of %error between corrugated cylinder and smooth cylinder model, for boundary 'a+b'

Table 1 Summary of boundary 'a' simulation runs conducted and adjusted parameters for comparing the corrugated cylinder and smooth cylinder models

	Run Title	b	a	ρ2	ρ1	pfar	n	m	Status	Plot Location	Max Allowable "m" or "l"
d=20*λ	Run a.20.100.0	20*λ	b*1	20*λ	ρ2*0.99	ρ2*10	3	0	Complete	Appendix	17
	Run a.20.100.1	20*λ	b*1	20*λ	ρ2*0.99	ρ2*10	3	1	Complete	--	17
	Run a.20.75.0	20*λ	b*.75	20*λ	ρ2*0.99	ρ2*10	3	0	Complete	--	14
	Run a.20.75.1	20*λ	b*.75	20*λ	ρ2*0.99	ρ2*10	3	1	Complete	--	14
	Run a.20.50.0	20*λ	b*.5	20*λ	ρ2*0.99	ρ2*10	3	0	Complete	--	12
	Run a.20.50.1	20*λ	b*.5	20*λ	ρ2*0.99	ρ2*10	3	1	Complete	--	12
	Run a.20.25.0	20*λ	b*.25	20*λ	ρ2*0.99	ρ2*10	3	0	Complete	--	10
	Run a.20.25.1	20*λ	b*.25	20*λ	ρ2*0.99	ρ2*10	3	1	Complete	--	10
	Run a.20.0.0	***20*λ***	***b*.001***	***20*λ***	***ρ2*0.99***	***ρ2*10***	***3***	***0***	***Complete***	***Chapter 7***	***0***
	Run a.20.0.0	20*λ	b*.001	20*λ	ρ2*0.99	ρ2*10	3	1	Bad Data/Ill-Conditioned	N/A	0
d=2*λ	Run a.2.100.0	2*λ	b*1	2*λ	p2*0.99	p2*10	3	0	Complete	Appendix	4
	Run a.2.100.1	2*λ	b*1	2*λ	p2*0.99	p2*10	3	1	Complete	--	4
	Run a.2.75.0	2*λ	b*.75	2*λ	p2*0.99	p2*10	3	0	Complete	--	3
	Run a.2.75.1	2*λ	b*.75	2*λ	p2*0.99	p2*10	3	1	Complete	--	3
	Run a.2.50.0	2*λ	b*.5	2*λ	p2*0.99	p2*10	3	0	Complete	--	2
	Run a.2.50.1	2*λ	b*.5	2*λ	p2*0.99	p2*10	3	1	Complete	--	2
	Run a.2.25.0	2*λ	b*.25	2*λ	p2*0.99	p2*10	3	0	Complete	--	1
	Run a.2.25.1	2*λ	b*.25	2*λ	p2*0.99	p2*10	3	1	Could not solve for 'a' coefficients "Input	N/A	1
	Run a.2.0.0	***2*λ***	***b*.001***	***2*λ***	***ρ2*0.99***	***ρ2*10***	***3***	***0***	***Complete***	***Chapter 7***	***0***
	Run a.2.0.1	2*λ	b*.001	2*λ	p2*0.99	p2*10	3	1	Bad Data/Ill-Conditioned	N/A	0
d=0.1*λ	Run a.0.1.100.0	.1*λ	b*1	.1*λ	p2*0.99	p2*10	3	0	Complete	Appendix	0
	Run a.0.1.100.1	.1*λ	b*1	.1*λ	p2*0.99	p2*10	3	1	Bad Data/Ill-Conditioned	N/A	0
	Run a.0.1.75.0	.1*λ	b*.75	.1*λ	p2*0.99	p2*10	3	0	Complete	--	0
	Run a.0.1.75.1	.1*λ	b*.75	.1*λ	p2*0.99	p2*10	3	1	Bad Data/Ill-Conditioned	N/A	0
	Run a.0.1.50.0	.1*λ	b*.5	.1*λ	p2*0.99	p2*10	3	0	Complete	--	0
	Run a.0.1.50.1	.1*λ	b*.5	.1*λ	p2*0.99	p2*10	3	1	Bad Data/Ill-Conditioned	N/A	0
	Run a.0.1.25.0	.1*λ	b*.25	.1*λ	p2*0.99	p2*10	3	0	Complete	--	0
	Run a.0.1.25.1	.1*λ	b*.25	.1*λ	p2*0.99	p2*10	3	1	Bad Data/Ill-Conditioned	N/A	0
	Run a.0.1.0.0	***.1*λ***	***b*.001***	***.1*λ***	***ρ2*0.99***	***ρ2*10***	***3***	***0***	***Complete***	***Chapter 7***	***0***
	Run a.0.1.0.1	.1*λ	b*.001	.1*λ	p2*0.99	p2*10	3	1	Bad Data/Ill-Conditioned	N/A	0

Table 2 Summary of boundary 'b' simulation runs conducted and adjusted parameters for comparing the corrugated cylinder and smooth cylinder models

Run Title	b	a	ρ2	ρ1	pfar	n	m	Status	Plot Location	Max Allowable "m" or "l"
Run b.20.0.0	20*λ	b*.001	20*λ	ρ2*0.99	ρ2*10	3	0	Complete	Chapter 7	0
Run b.2.0.0	2*λ	b*.001	2*λ	ρ2*0.99	ρ2*10	3	0	Complete	Chapter 7	0
Run b.0.1.0.0	.1*λ	b*.001	.1*λ	ρ2*0.99	ρ2*10	3	0	Complete	Chapter 7	0

Table 3 Summary of boundary 'a+b' simulation runs conducted and adjusted parameters for comparing the corrugated cylinder and smooth cylinder models

Run Title	b	a	ρ2	ρ1	pfar	n	m	Status	Plot Location	Max Allowable "m" or "l"
Run a_plus_b.20.0.0	20*λ	b*.001	20*λ	ρ2*0.99	ρ2*10	3	0	Complete	Chapter 7	0
Run a_plus_b..2.0.0	2*λ	b*.001	2*λ	ρ2*0.99	ρ2*10	3	0	Complete	Chapter 7	0
Run a_plus_b..0.1.0.0	.1*λ	b*.001	.1*λ	ρ2*0.99	ρ2*10	3	0	Complete	Chapter 7	0

For sections 7.5.1 through 7.5.9 , refer to Figure 7-2 to interpret the title of each section in order to understand the overall configuration of the model for the plots captured in that section. Also, the first table in each section will provide a more detailed set of configuration parameters specifically for the polar plots, preceding the polar plots. The second table in each section, after the polar plots but prior to the XY phi plots, will provide a detailed set of configuration parameters specifically for the XY phi plots. The third table in each section (if present, as not all results had these sets of plots), after the XY phi plots but prior to the XY rho plots, will provide a detailed set of configuration parameters specifically for the XY rho plots.

7.5.1 Run a.20.0.0 (b=20λ, a=b*.001, ρ2=20λ, m=0)

Table 4 Detailed parameters summary for changing φ plots of Run a.20.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	1.50105 x 10^10 Hz	-	-	-	-
Frequency	0.019986 m	-	-	-	-
a	0.02 λ	-	-	-	-
b	20. λ	-	-	-	-
ρ1	19.98 λ	-	-	-	-
ρ2	20. λ	-	-	-	-
ϕ range	-	0.327273 Deg	359.673 Deg	0.654545 Deg	550
ρ (observed)	200. λ	-	-	-	-
z (observed)	0.25 a && 0.005 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
ϕi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.04	-	-	-	-
max allowable l	8.537	-	-	-	-
Boundary	Boundary a	-	-	-	-

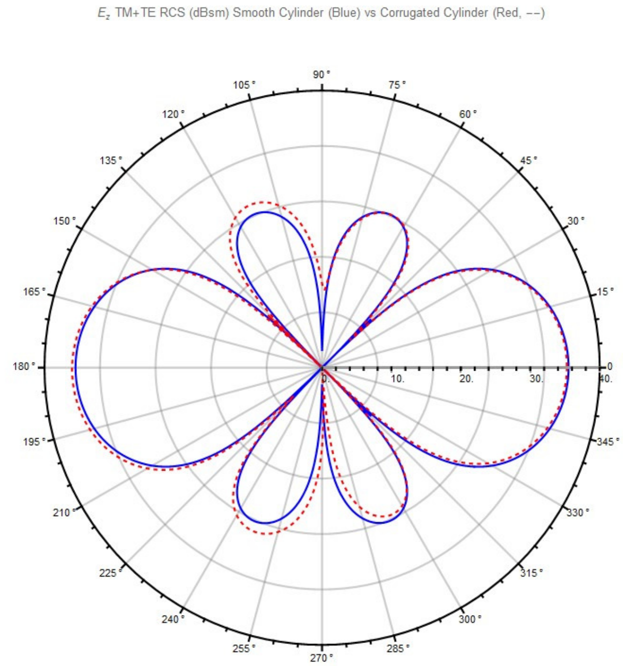


Figure 7-6 Polar Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a.20.0.0

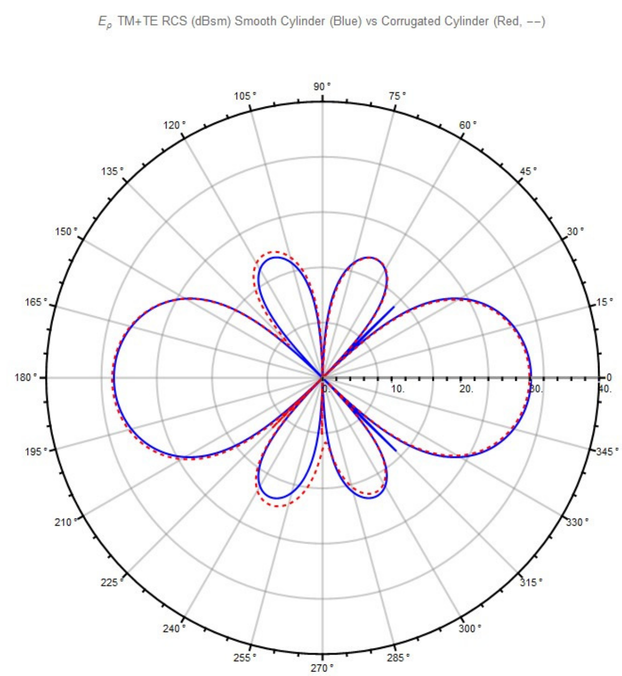


Figure 7-7 Polar Plot form of RCS dBsm for Eρ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a.20.0.0

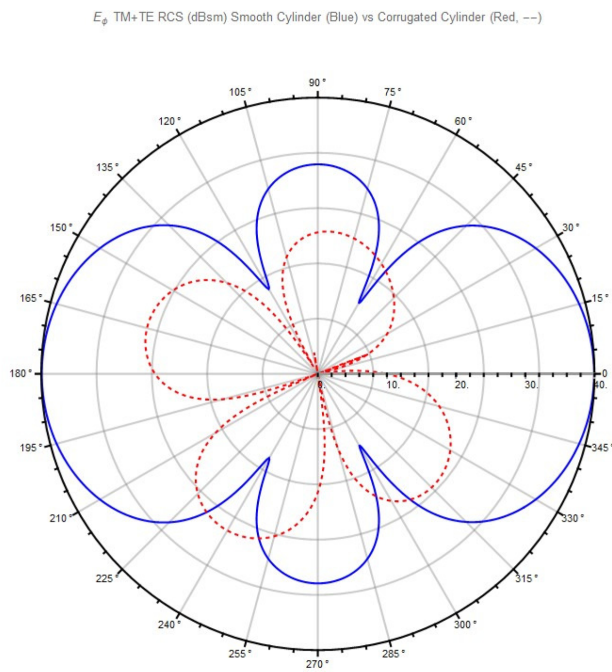


Figure 7-8 Polar Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a.20.0.0

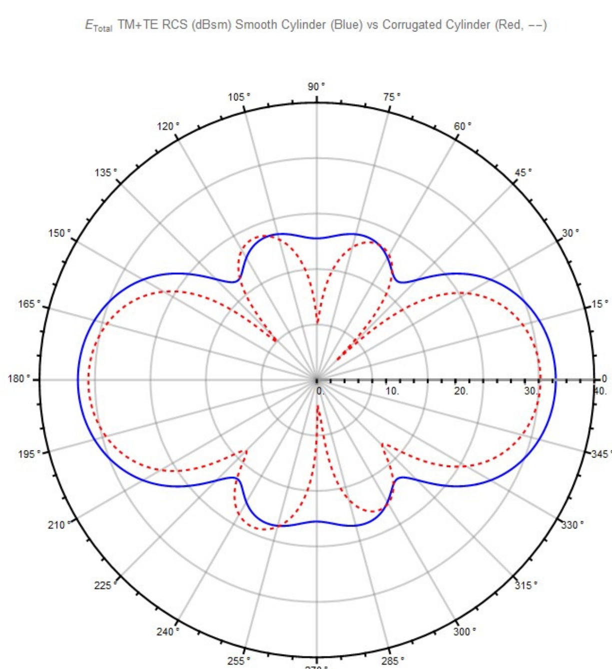


Figure 7-9 Polar Plot form of RCS dBsm for E_{Total} of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.20.0.0

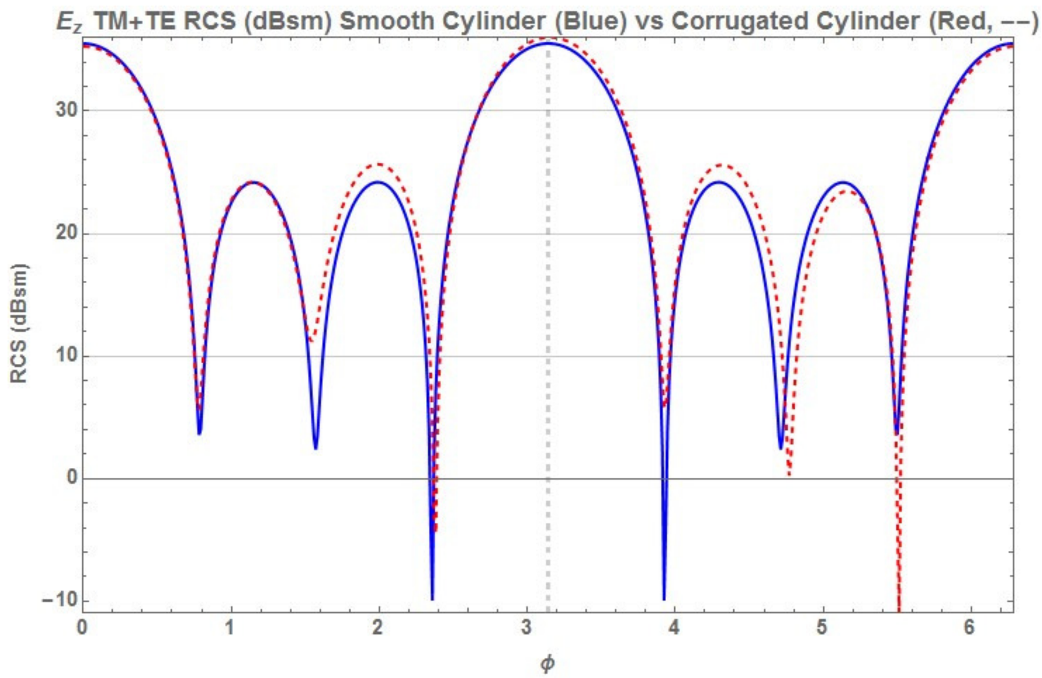


Figure 7-10 XY Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.20.0.0

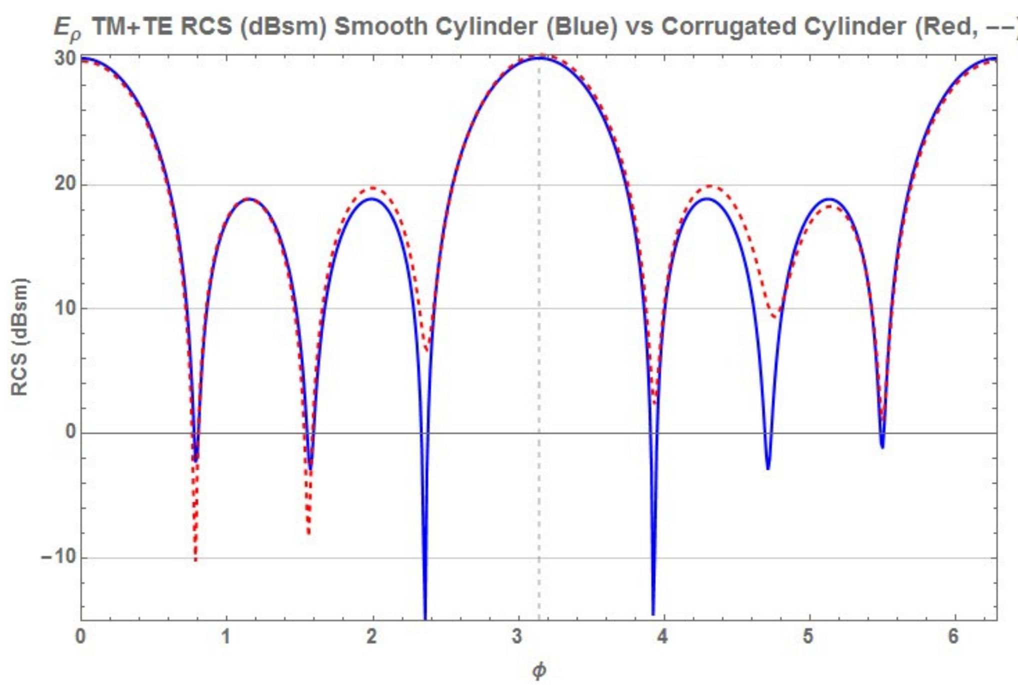


Figure 7-11 XY Plot form of RCS dBsm for E_ρ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.20.0.0

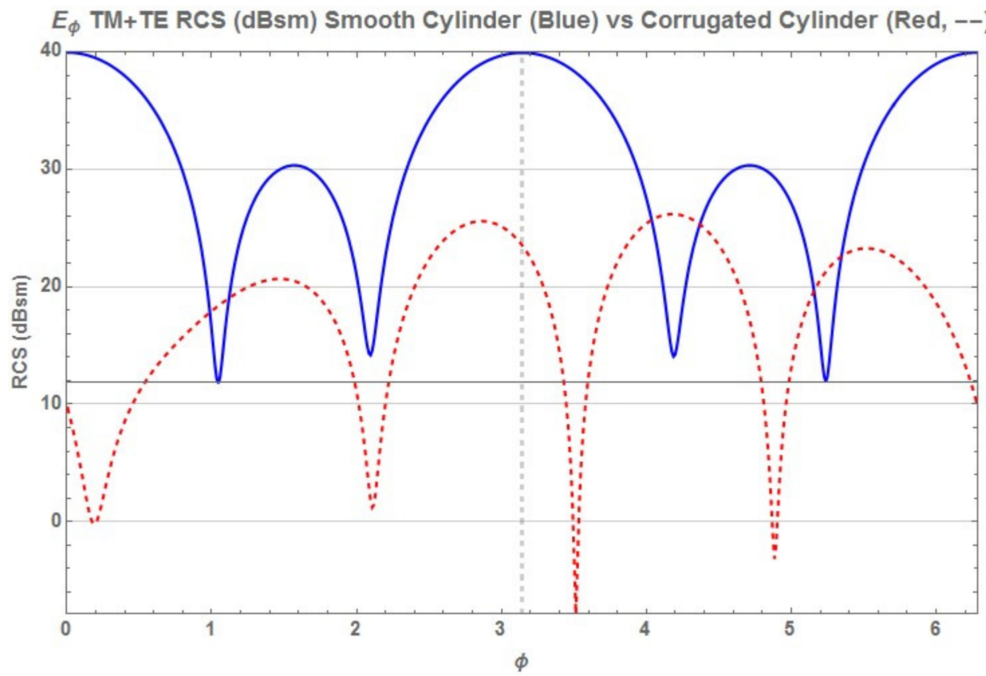


Figure 7-12 XY Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.20.0.0

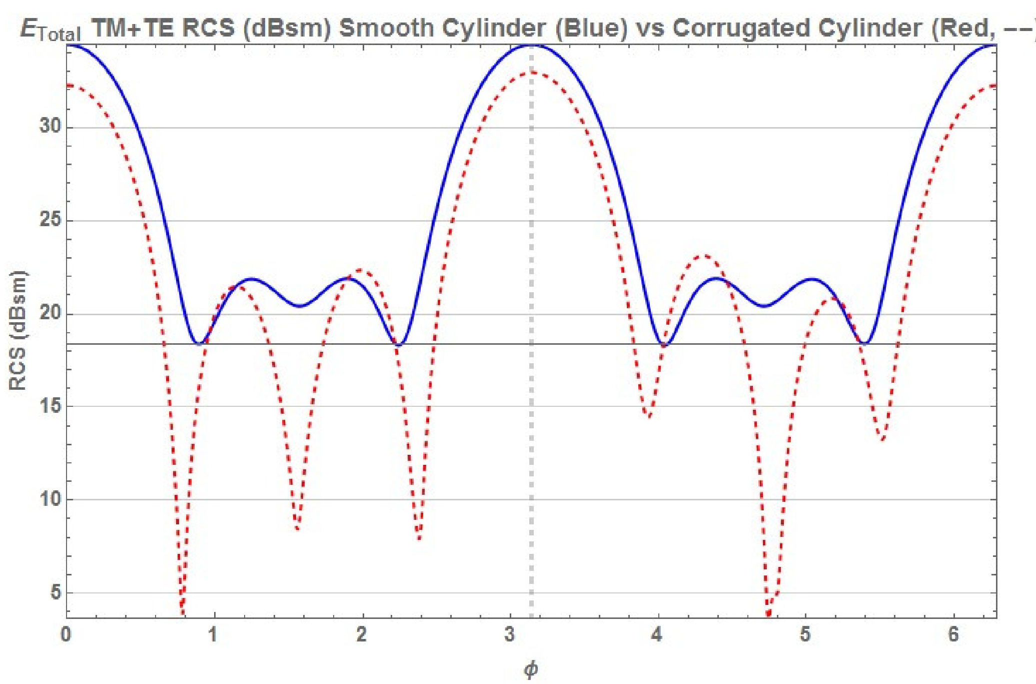


Figure 7-13 XY Plot form of RCS dBsm for E_{Total} of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.20.0.0

Table 5 Detailed parameters summary for changing ρ plots of Run a.20.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	0.019986 m	-	-	-	-
Frequency	1.50105×10^{10} Hz	-	-	-	-
a	0.02 λ	-	-	-	-
b	20. λ	-	-	-	-
ρ1	19.98 λ	-	-	-	-
ρ2	20. λ	-	-	-	-
ρ range	-	17.982 λ	30. λ	0.0218907 λ	550
ϕ (observed)	37. Deg	-	-	-	-
z (observed)	0.25 a && 0.005 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
ϕi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.04	-	-	-	-
max allowable l	8.537	-	-	-	-
Boundary	Boundary a	-	-	-	-

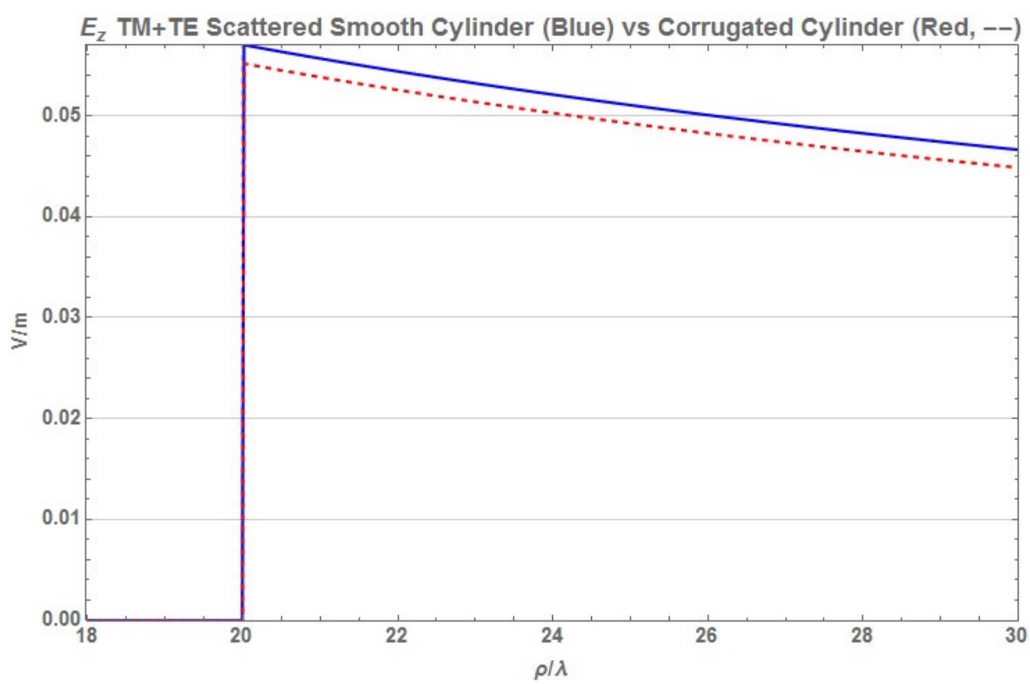


Figure 7-14 XY Plot of Scattered Field Amplitude Only, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.20.0.0

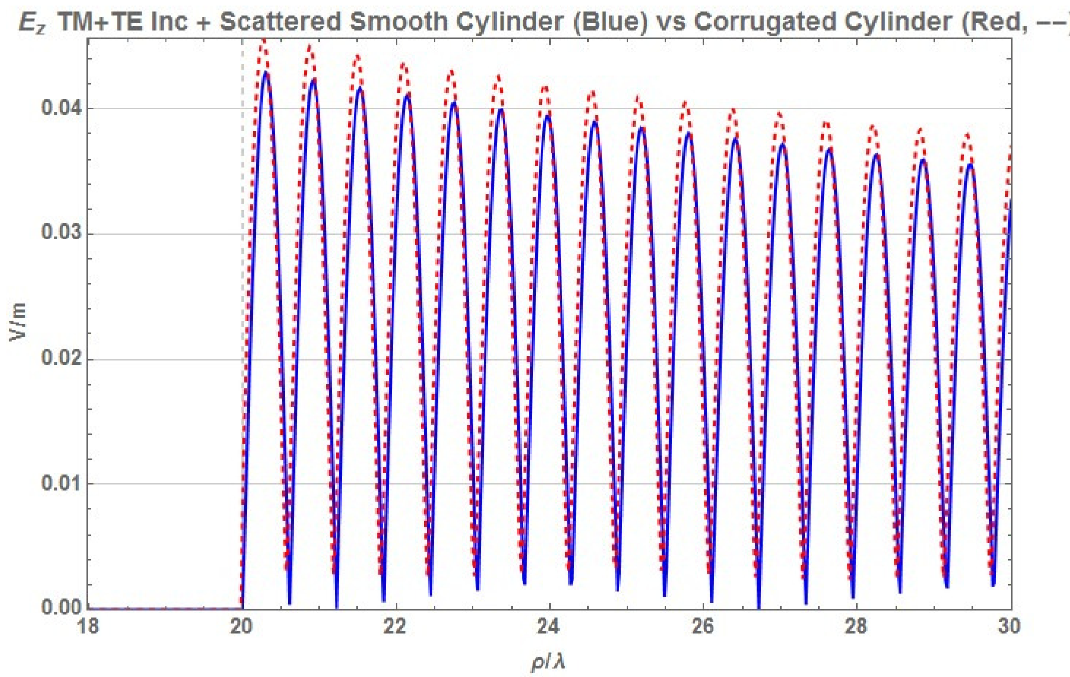


Figure 7-15 XY Plot of Scattered + Incident Field Amplitude, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.20.0.0

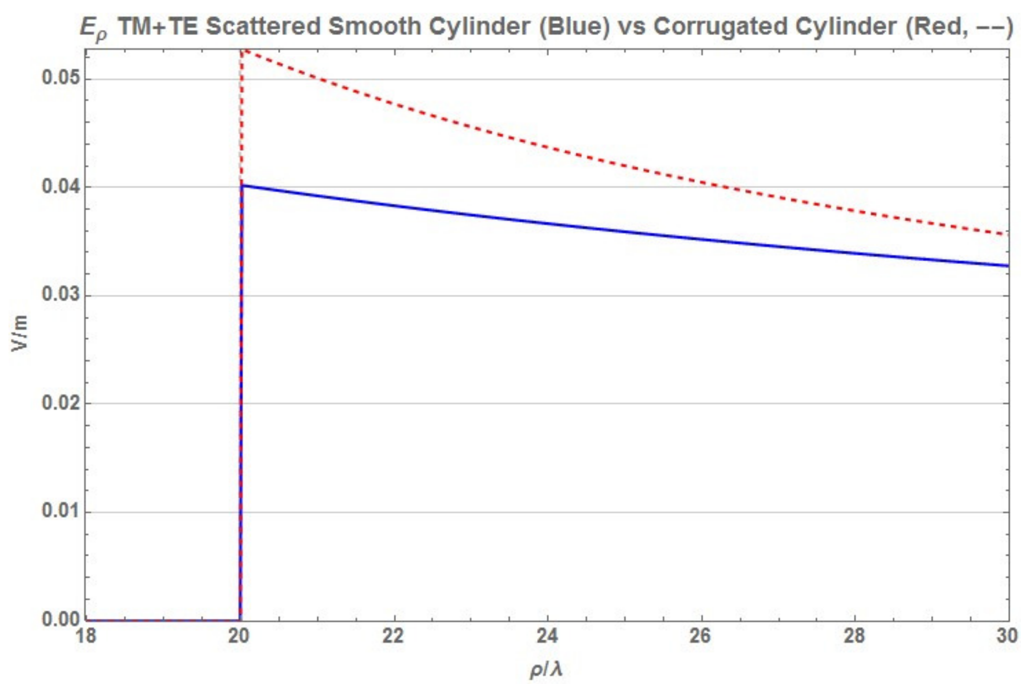


Figure 7-16 XY Plot of Scattered Field Amplitude Only, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.20.0.0

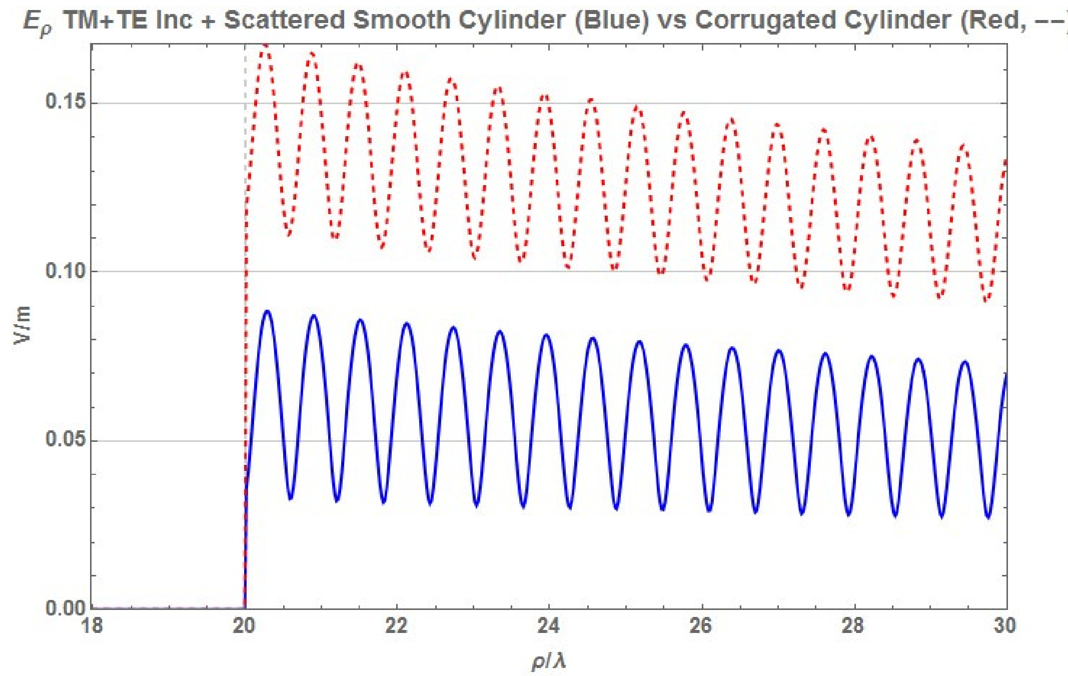


Figure 7-17 XY Plot of Scattered + Incident Field Amplitude, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.20.0.0

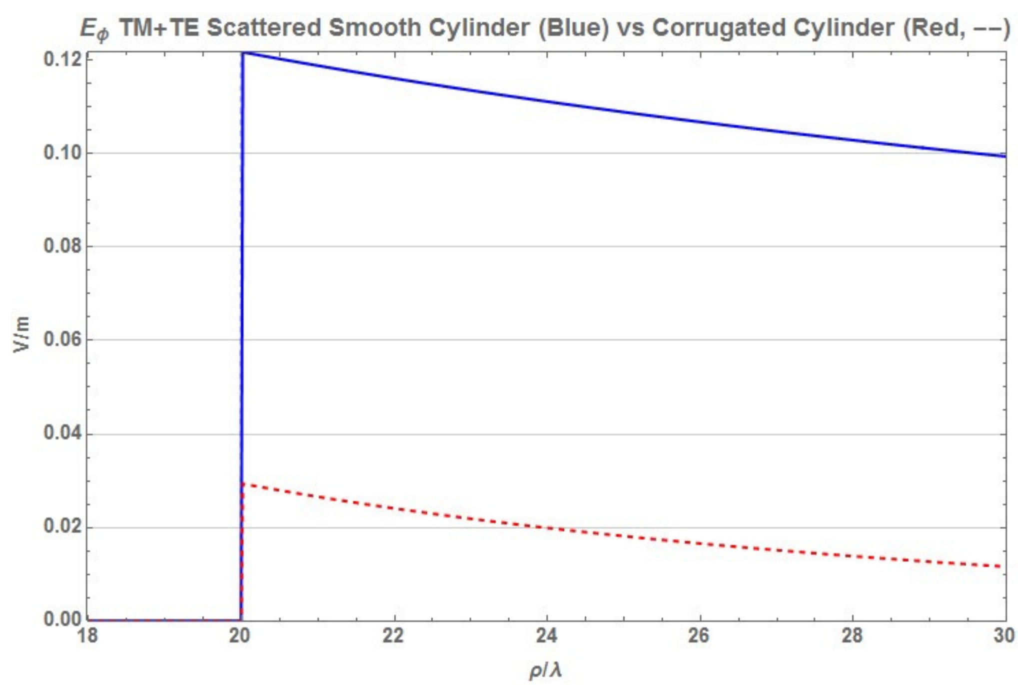


Figure 7-18 XY Plot of Scattered Field Amplitude Only, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.20.0.0

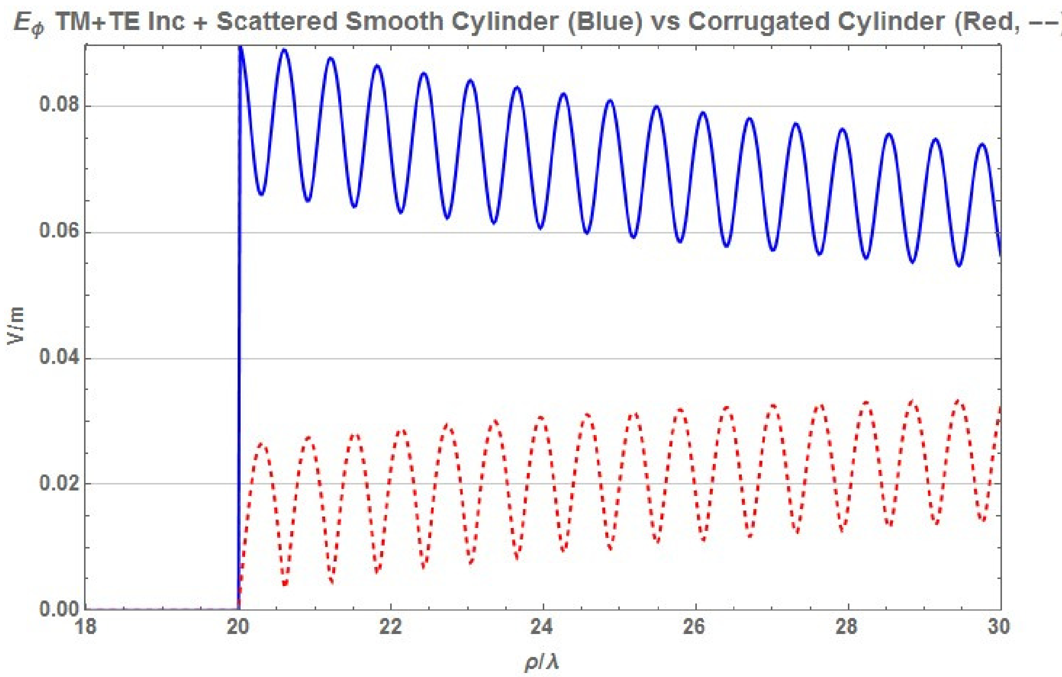


Figure 7-19 XY Plot of Scattered + Incident Field Amplitude, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.20.0.0

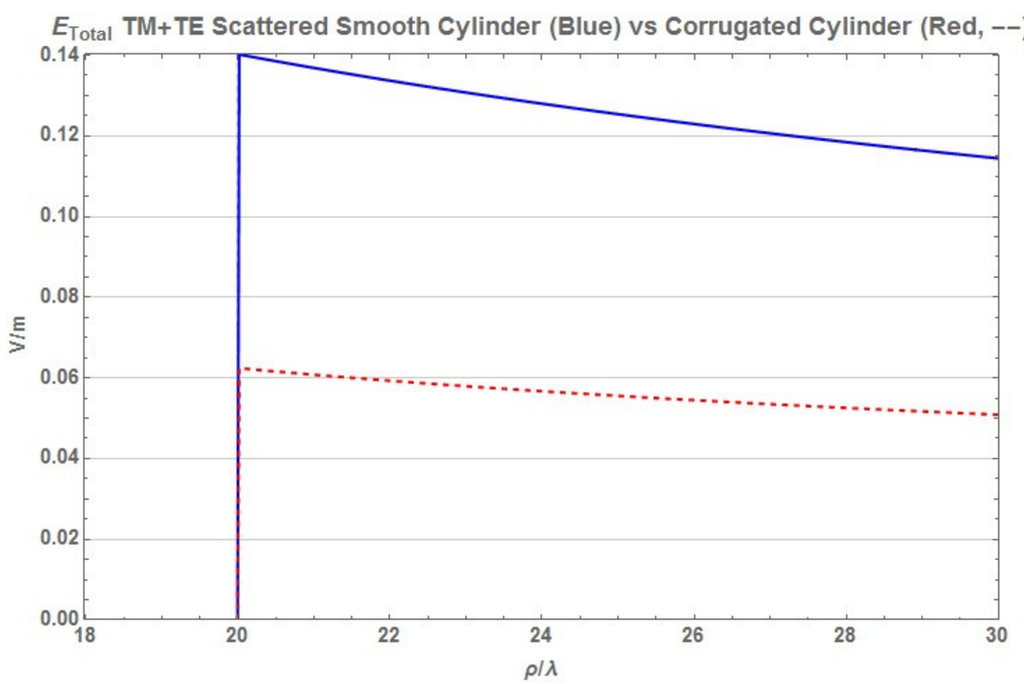


Figure 7-20 XY Plot of Scattered Field Amplitude Only, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.20.0.0

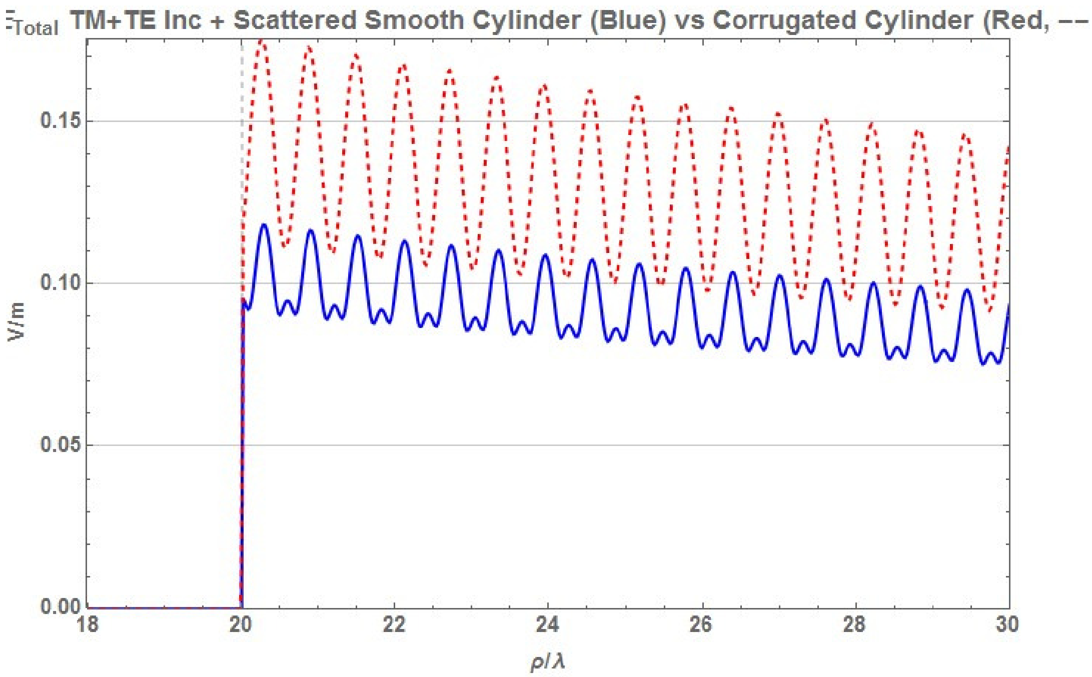


Figure 7-21 XY Plot of Scattered + Incident Field Amplitude, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.20.0.0

7.5.2 Run a.2.0.0 (b=2λ, a=b*.001, ρ2=2λ, m=0)

Table 6 Detailed parameters summary for changing φ plots of Run a.2.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	1.50105×10^{10} Hz	-	-	-	-
Frequency	0.019986 m	-	-	-	-
a	0.002 λ	-	-	-	-
b	2. λ	-	-	-	-
ρ1	1.998 λ	-	-	-	-
ρ2	2. λ	-	-	-	-
ϕ range	-	0.327273 Deg	359.673 Deg	0.654545 Deg	550
ρ (observed)	20. λ	-	-	-	-
z (observed)	0.25 a && 0.0005 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
ϕi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.004	-	-	-	-
max allowable l	0.8537	-	-	-	-
Boundary	Boundary a	-	-	-	-

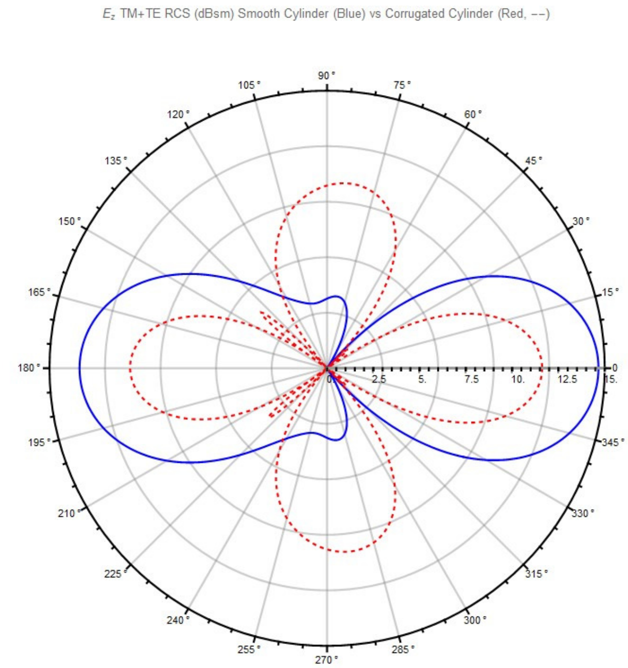


Figure 7-22 Polar Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a.2.0.0

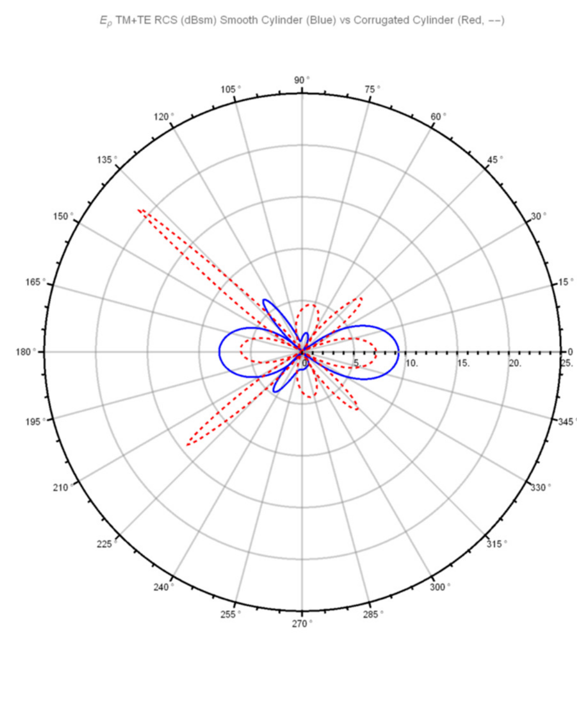


Figure 7-23 Polar Plot form of RCS dBsm for Eρ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a.2.0.0

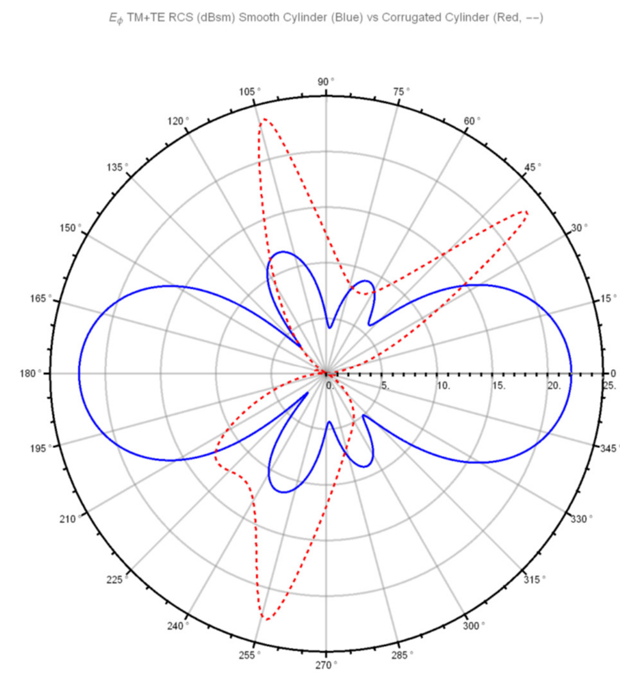


Figure 7-24 Polar Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a.2.0.0

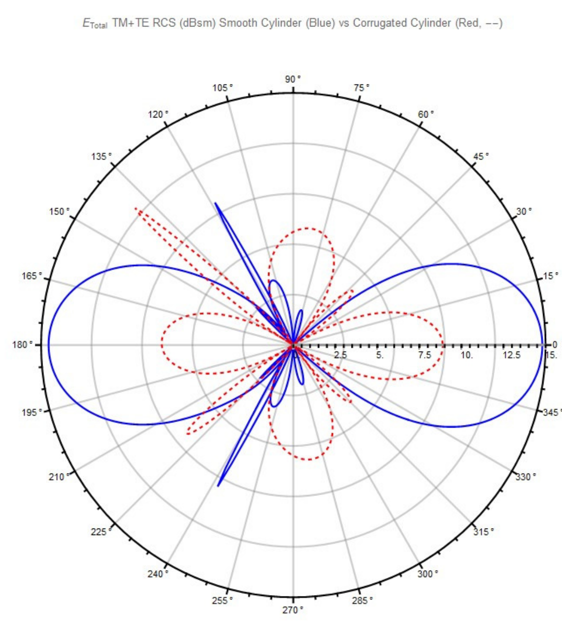


Figure 7-25 Polar Plot form of RCS dBsm for ETotal of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.2.0.0

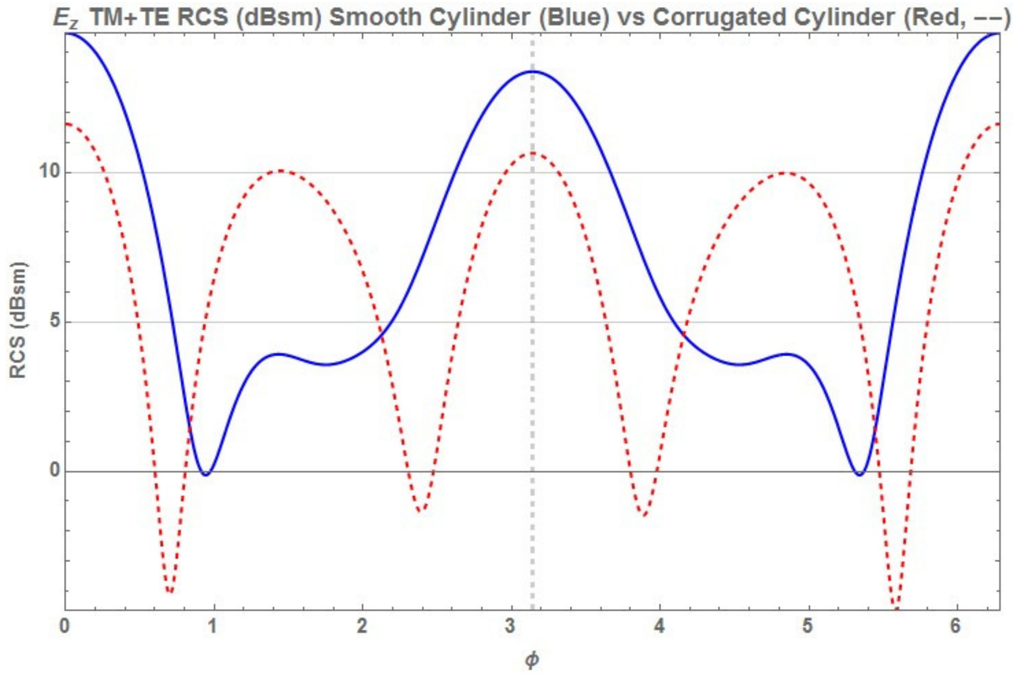


Figure 7-26 XY Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.2.0.0

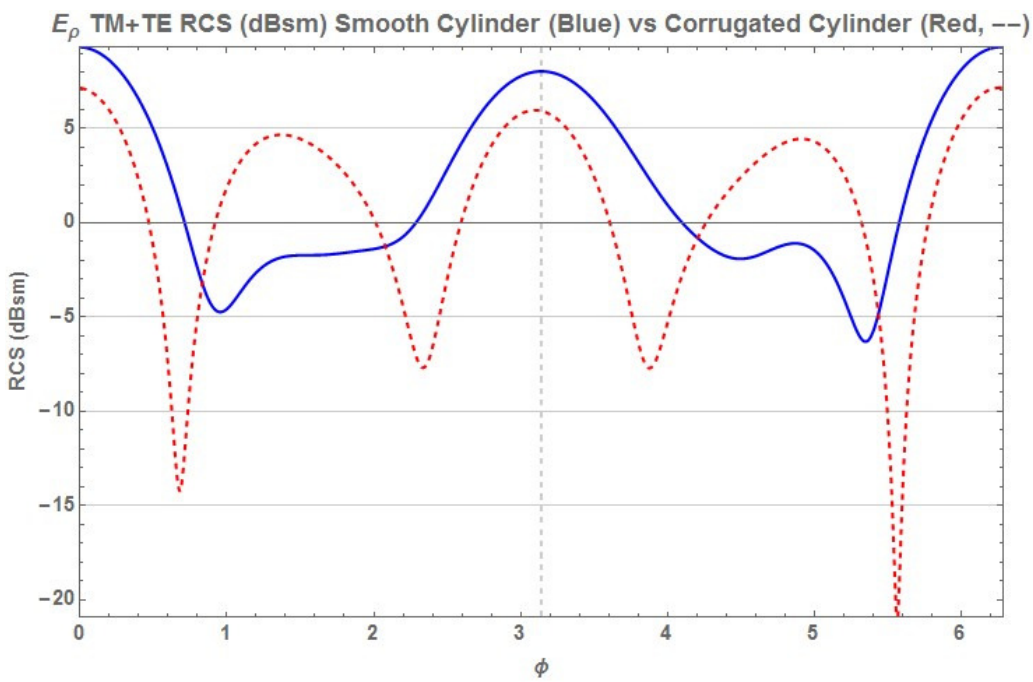


Figure 7-27 XY Plot form of RCS dBsm for Eρ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.2.0.0

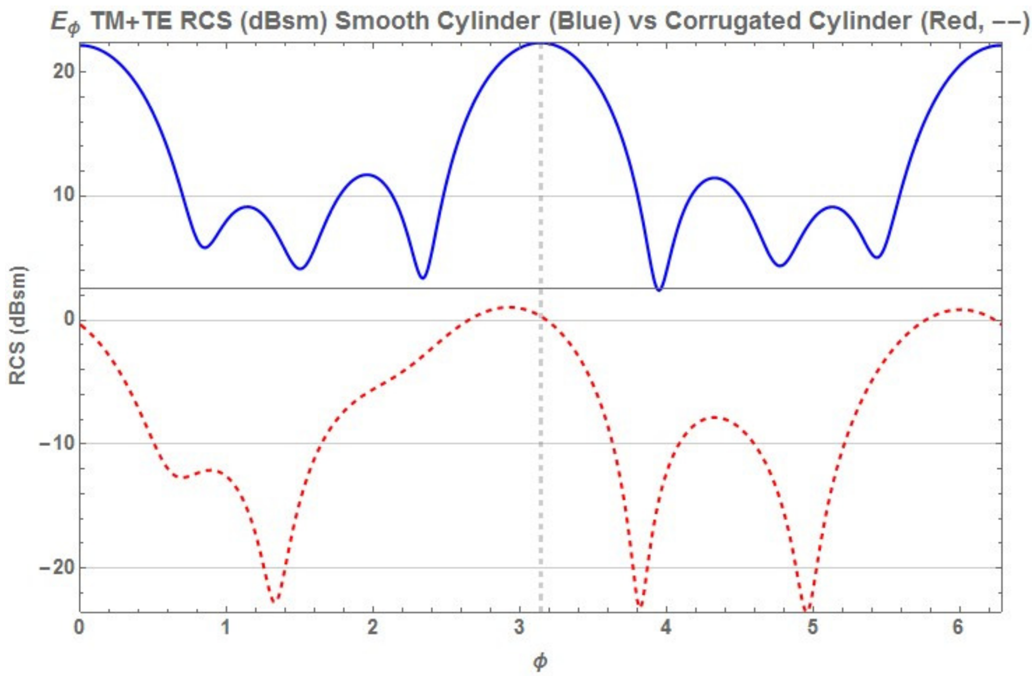


Figure 7-28 XY Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.2.0.0

.

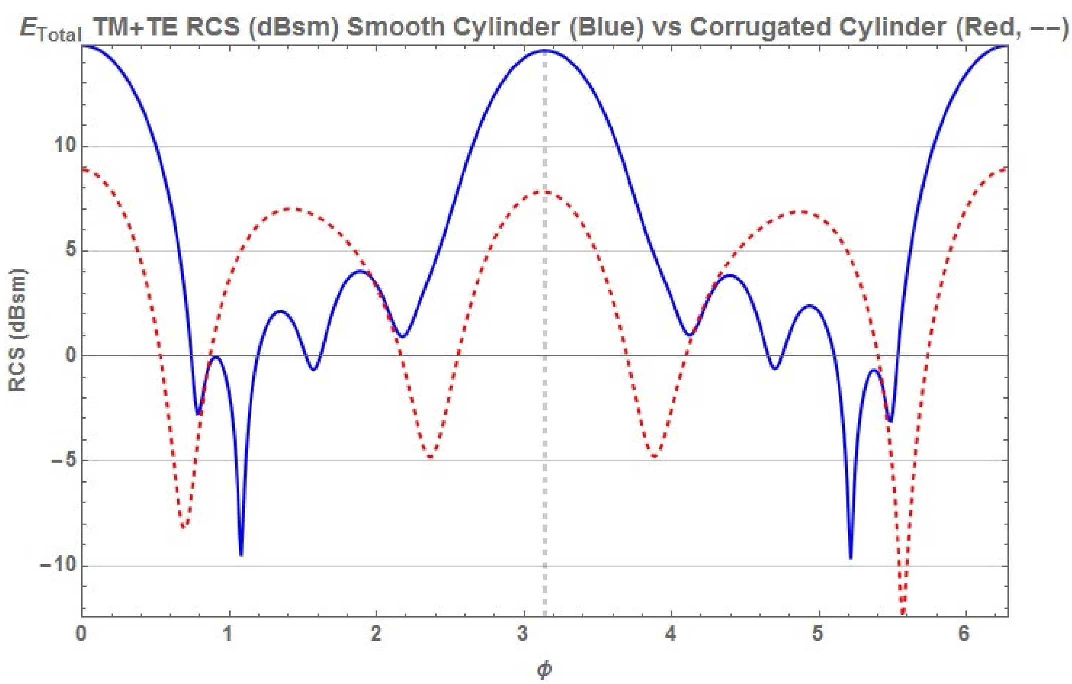


Figure 7-29 XY Plot form of RCS dBsm for ETotal of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.2.0.0

Table 7 Detailed parameters summary for changing ρ plots of Run a.2.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	0.019986 m	-	-	-	-
Frequency	1.50105×10^{10} Hz	-	-	-	-
a	0.002 λ	-	-	-	-
b	2. λ	-	-	-	-
ρ1	1.998 λ	-	-	-	-
ρ2	2. λ	-	-	-	-
ρ range	-	1.7982 λ	12. λ	0.0185825 λ	550
ϕ (observed)	37. Deg	-	-	-	-
z (observed)	0.25 a && 0.0005 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
ϕi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.004	-	-	-	-
max allowable l	0.8537	-	-	-	-
Boundary	Boundary a	-	-	-	-

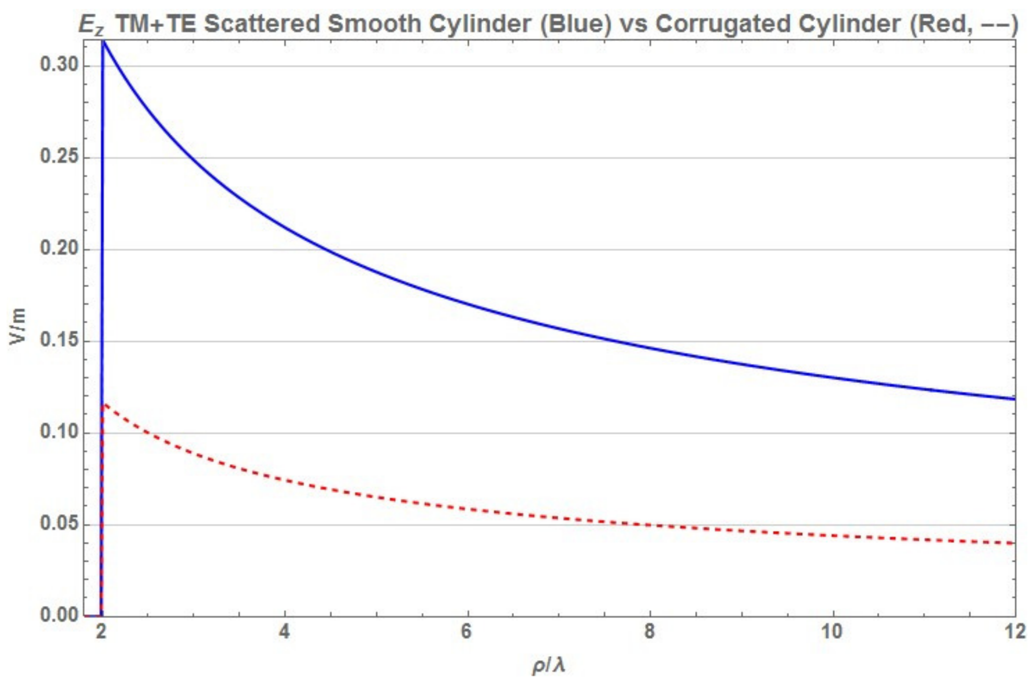


Figure 7-30 XY Plot of Scattered Field Amplitude Only, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.2.0.0

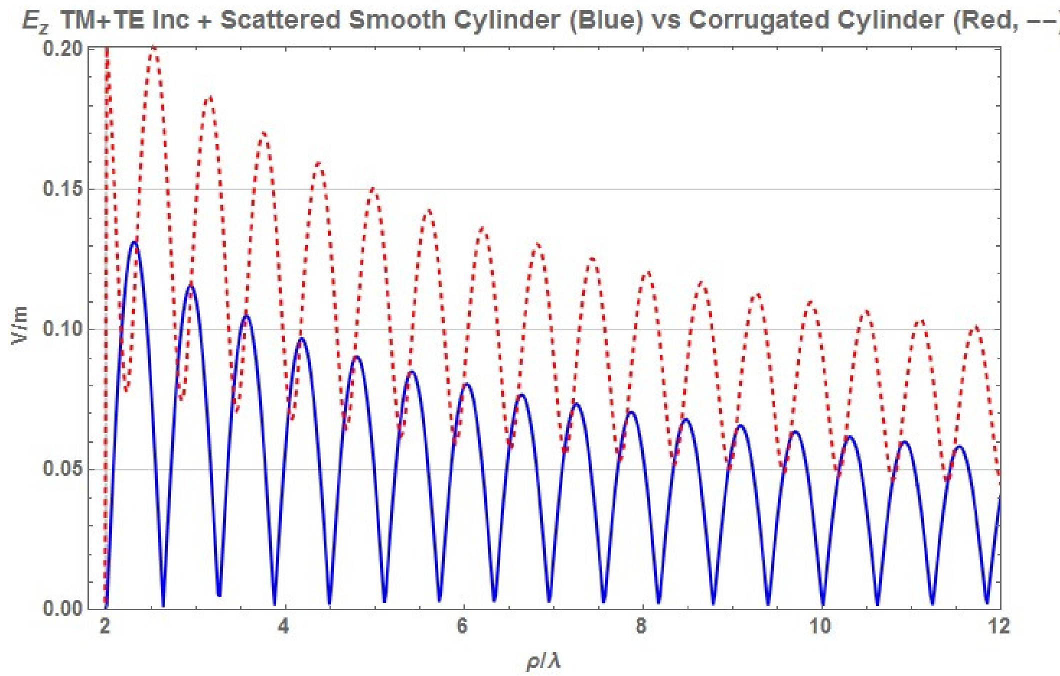


Figure 7-31 XY Plot of Scattered + Incident Field Amplitude, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.2.0.0

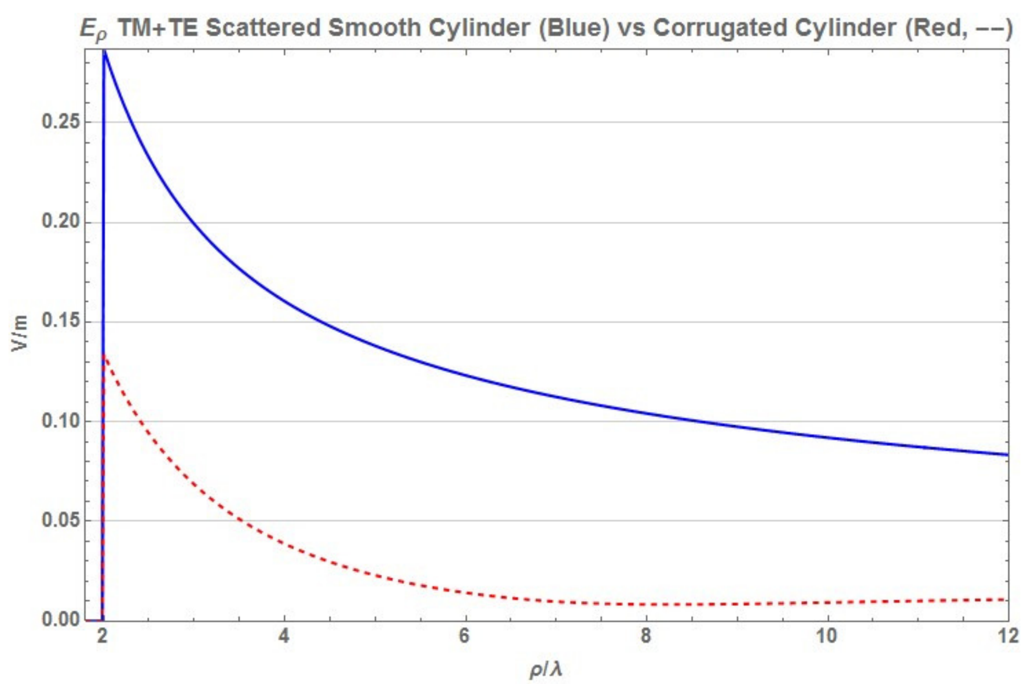


Figure 7-32 XY Plot of Scattered Field Amplitude Only, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.2.0.0

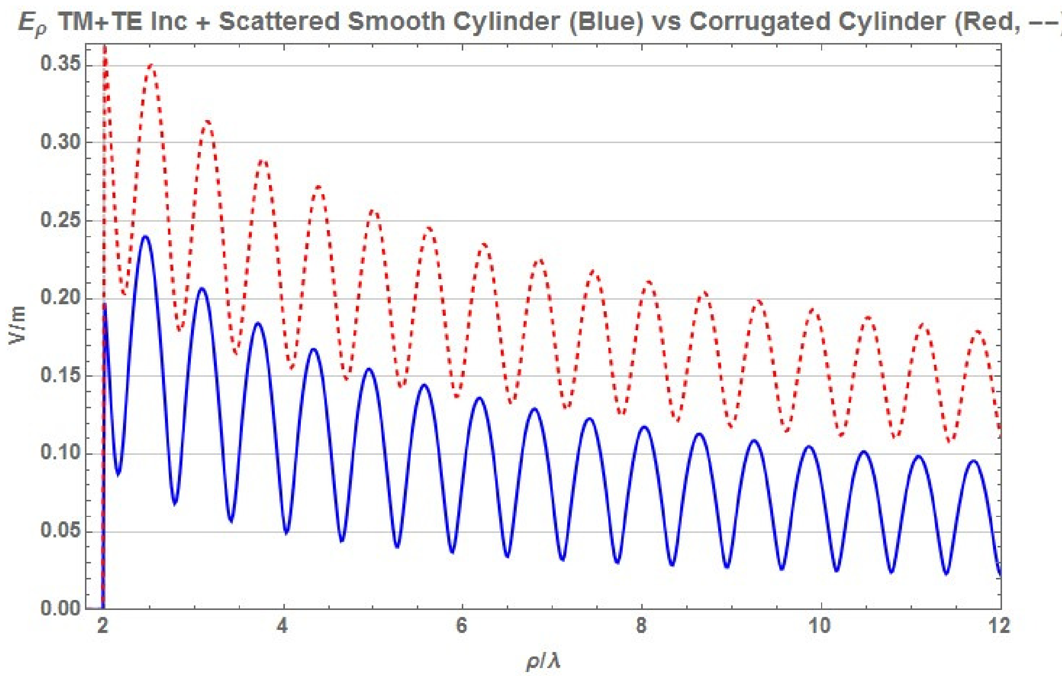


Figure 7-33 XY Plot of Scattered + Incident Field Amplitude, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.2.0.0

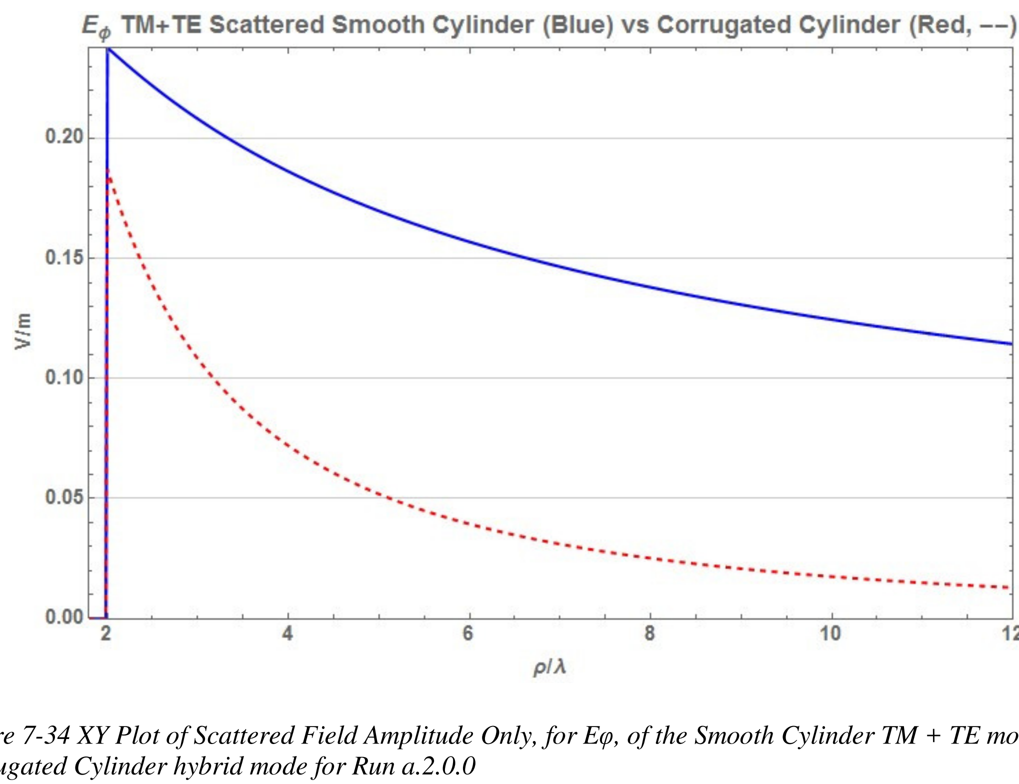


Figure 7-34 XY Plot of Scattered Field Amplitude Only, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.2.0.0

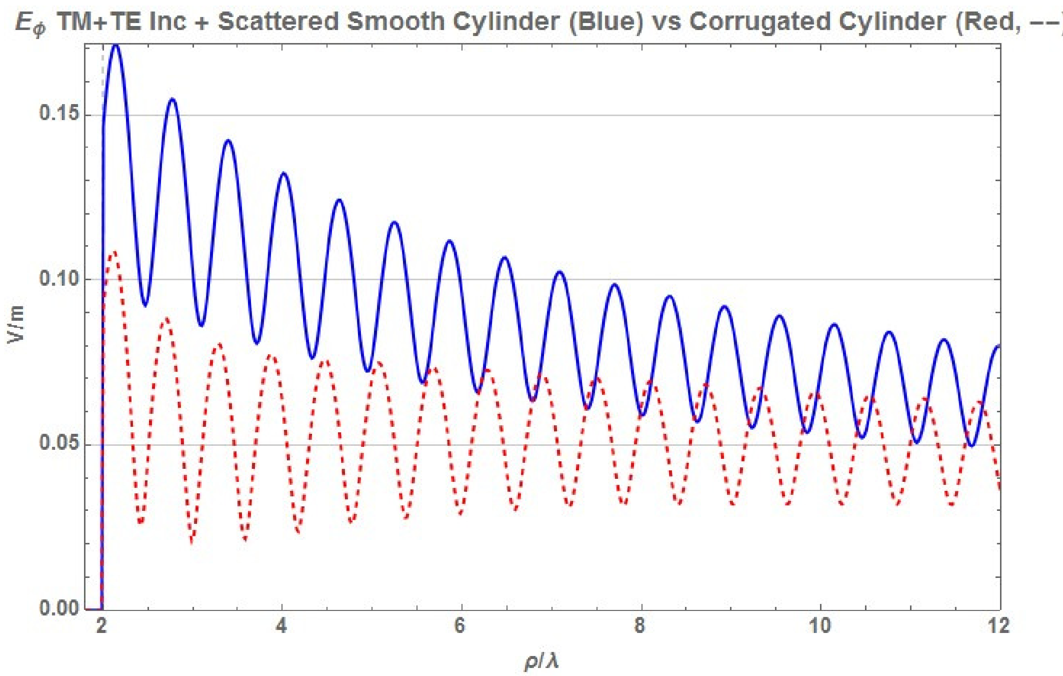


Figure 7-35 XY Plot of Scattered + Incident Field Amplitude, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.2.0.0

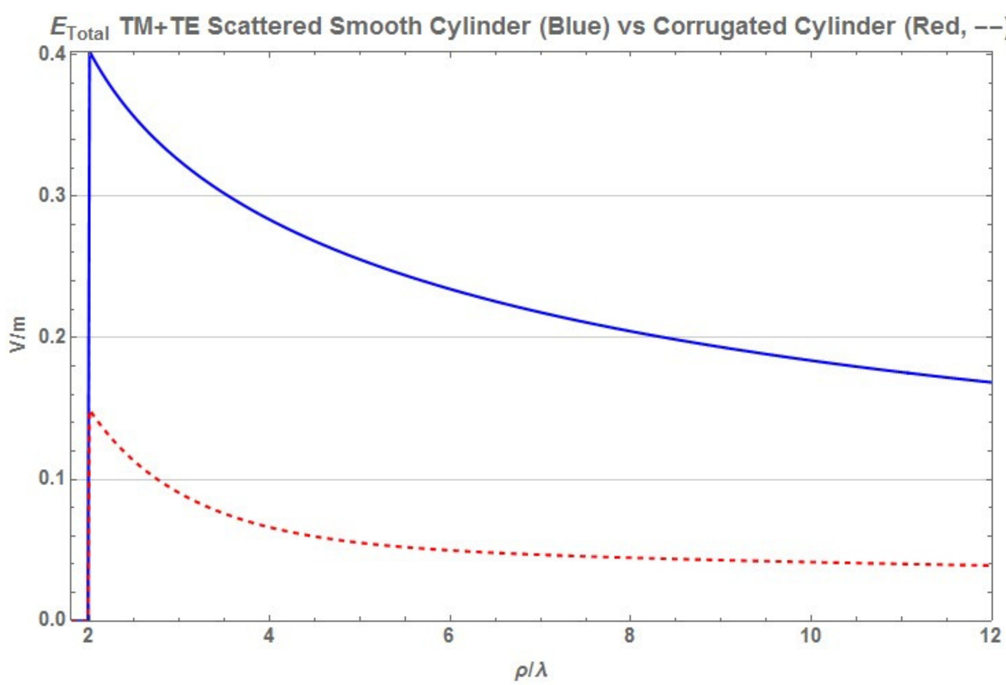


Figure 7-36 XY Plot of Scattered Field Amplitude Only, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.2.0.0

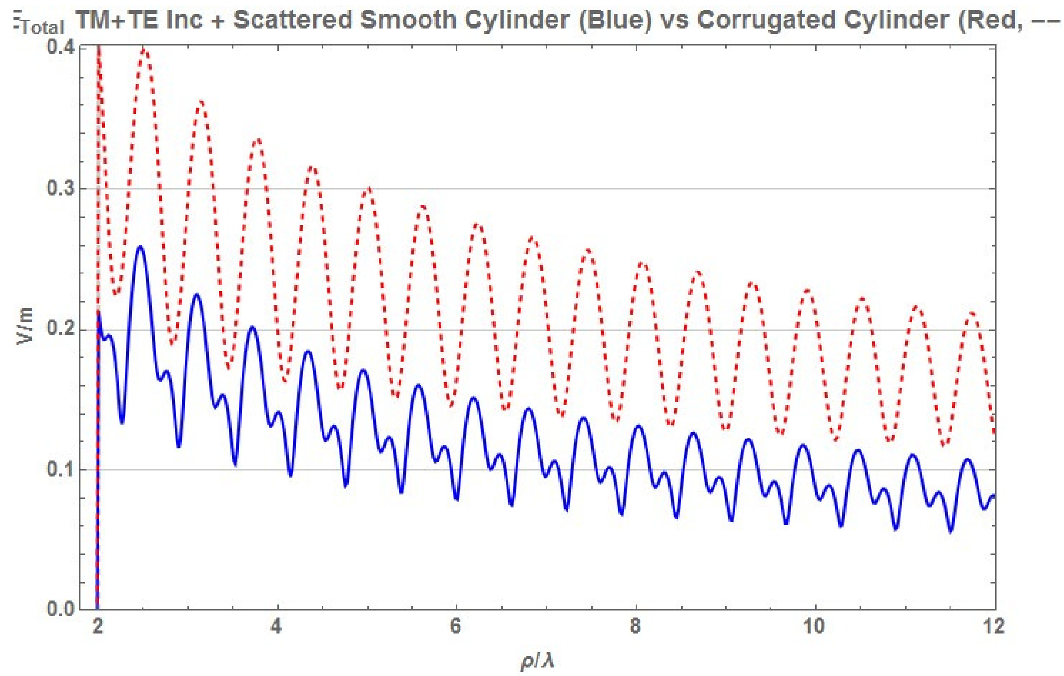


Figure 7-37 XY Plot of Scattered + Incident Field Amplitude, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.2.0.0

7.5.3 Run a.0.1.0.0 (b=0.1λ, a=b*.001, ρ2=0.1λ, m=0)

Table 8 Detailed parameters summary for changing φ plots of Run a.0.1.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	1.50105×10^{10} Hz	-	-	-	-
Frequency	0.019986 m	-	-	-	-
a	0.0001 λ	-	-	-	-
b	0.1 λ	-	-	-	-
ρ1	0.0999 λ	-	-	-	-
ρ2	0.1 λ	-	-	-	-
φ range	-	0.327273 Deg	359.673 Deg	0.654545 Deg	550
ρ (observed)	1. λ	-	-	-	-
z (observed)	0.25 a && 0.000025 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
φi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.0002	-	-	-	-
max allowable l	0.042685	-	-	-	-
Boundary	Boundary a	-	-	-	-

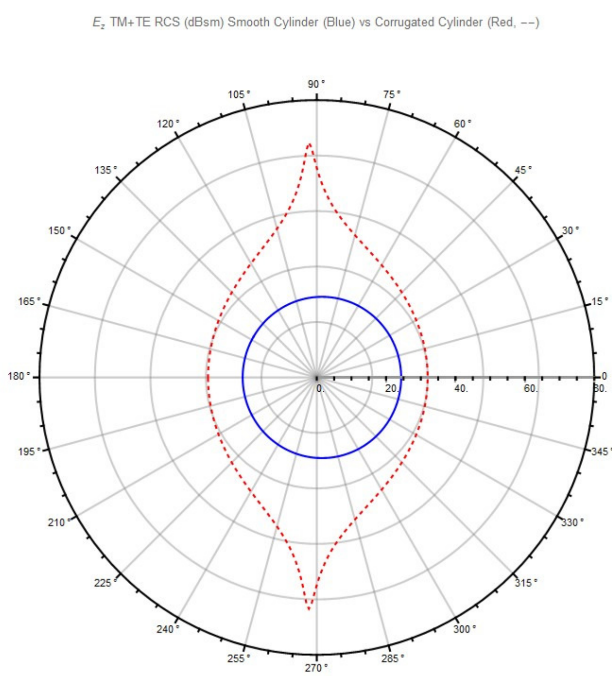


Figure 7-38 Polar Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a.0.1.0.0

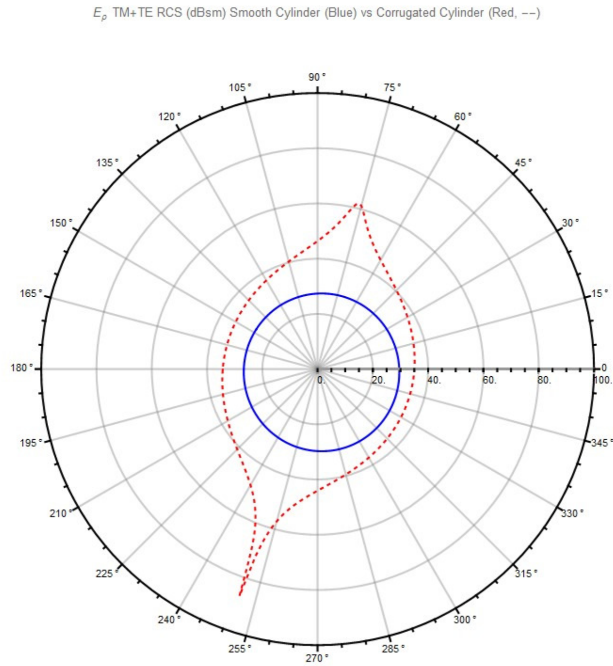


Figure 7-39 Polar Plot form of RCS dBsm for Eρ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a.0.1.0.0

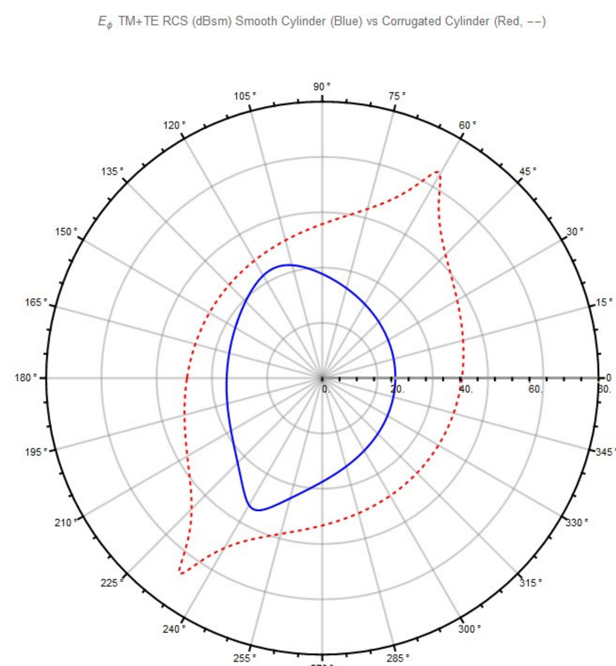


Figure 7-40 Polar Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a.0.1.0.0

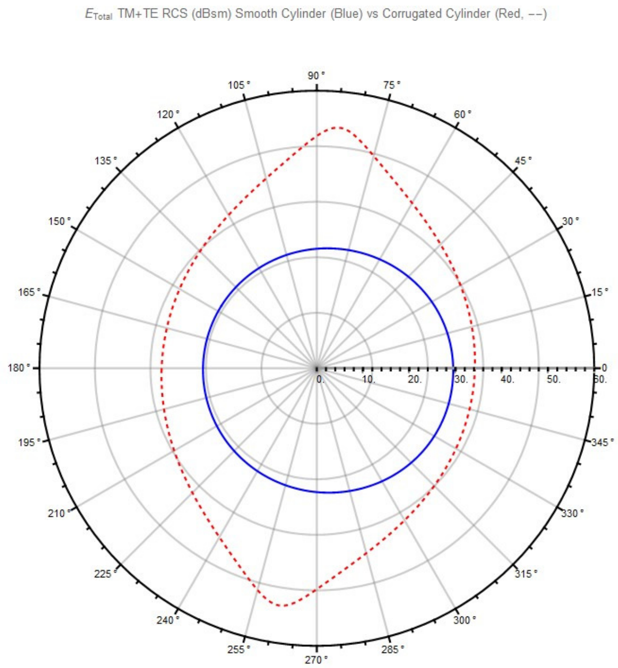


Figure 7-41 Polar Plot form of RCS dBsm for ETotal of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.0.1.0.0

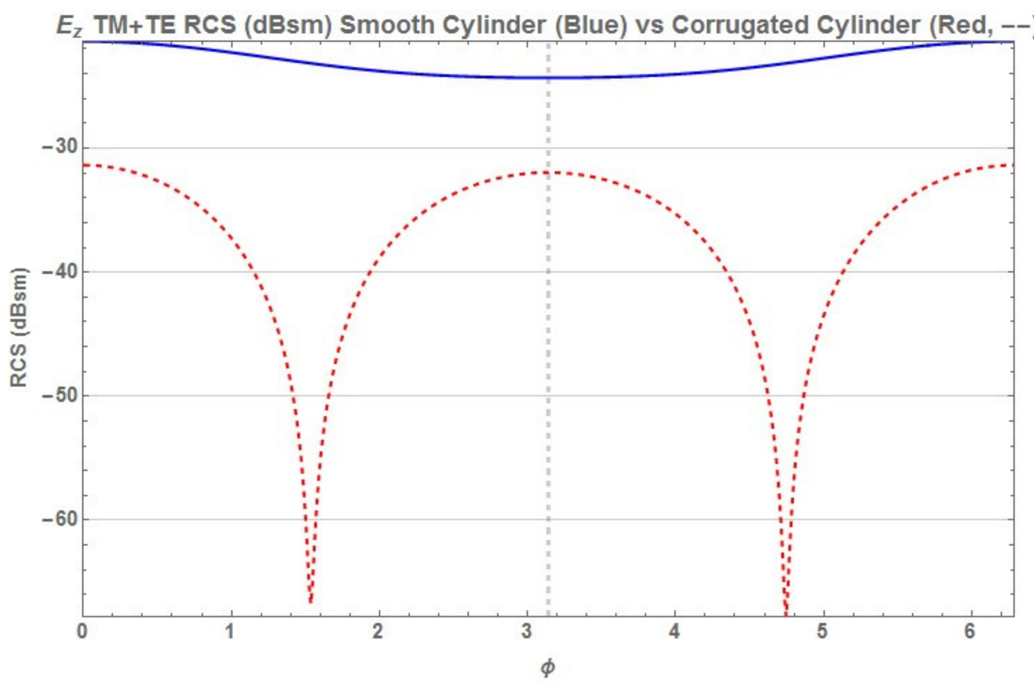


Figure 7-42 XY Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.0.1.0.0

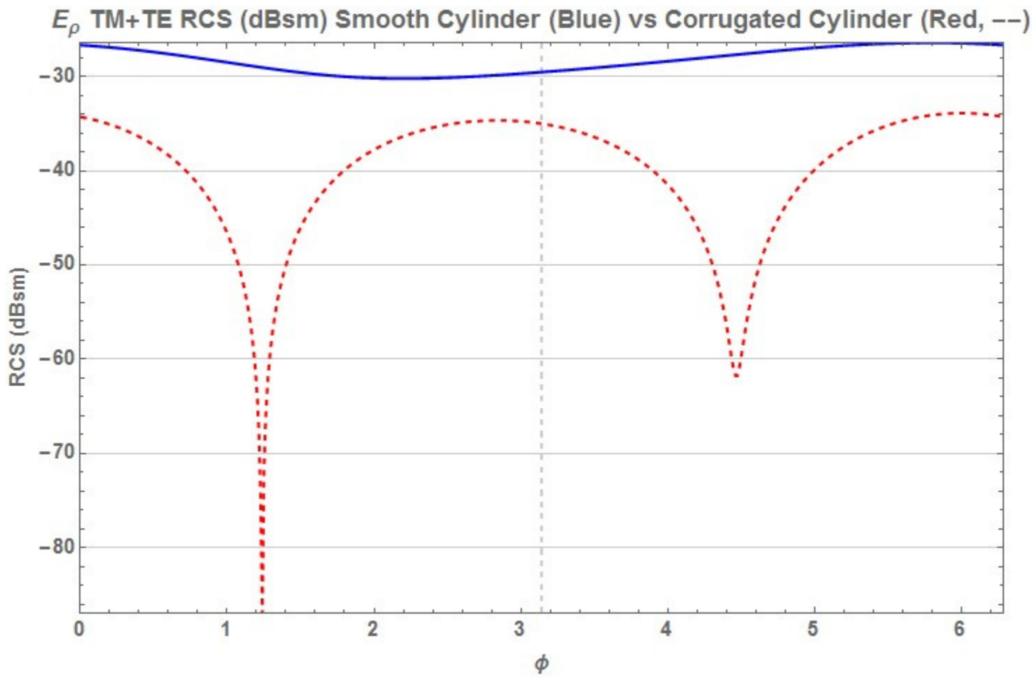


Figure 7-43 XY Plot form of RCS dBsm for Eρ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.0.1.0.0

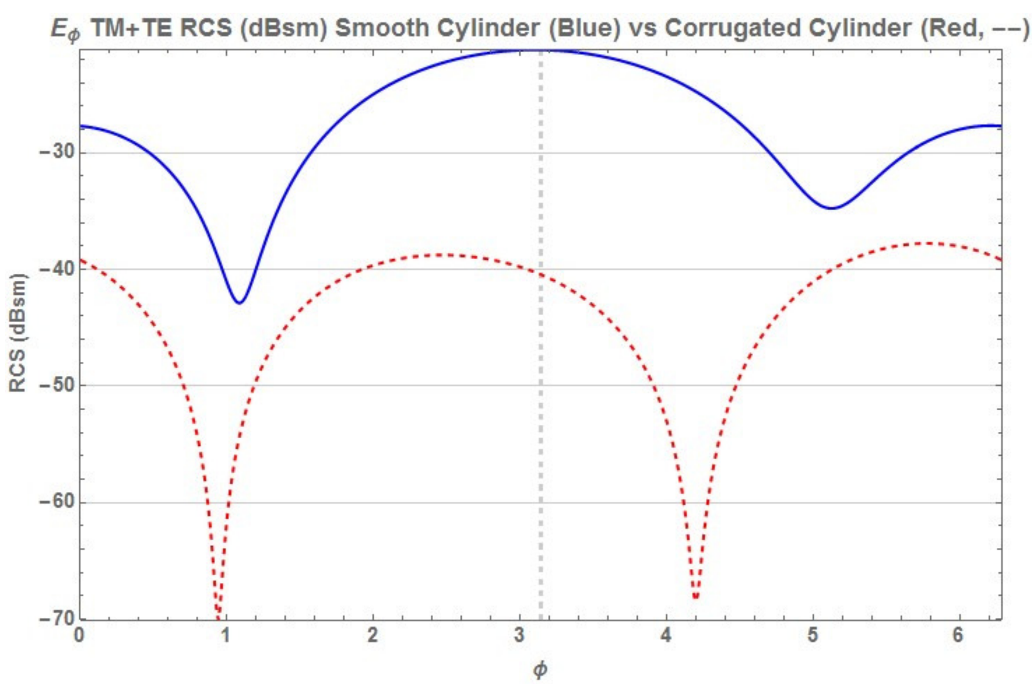


Figure 7-44 XY Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.0.1.0.0

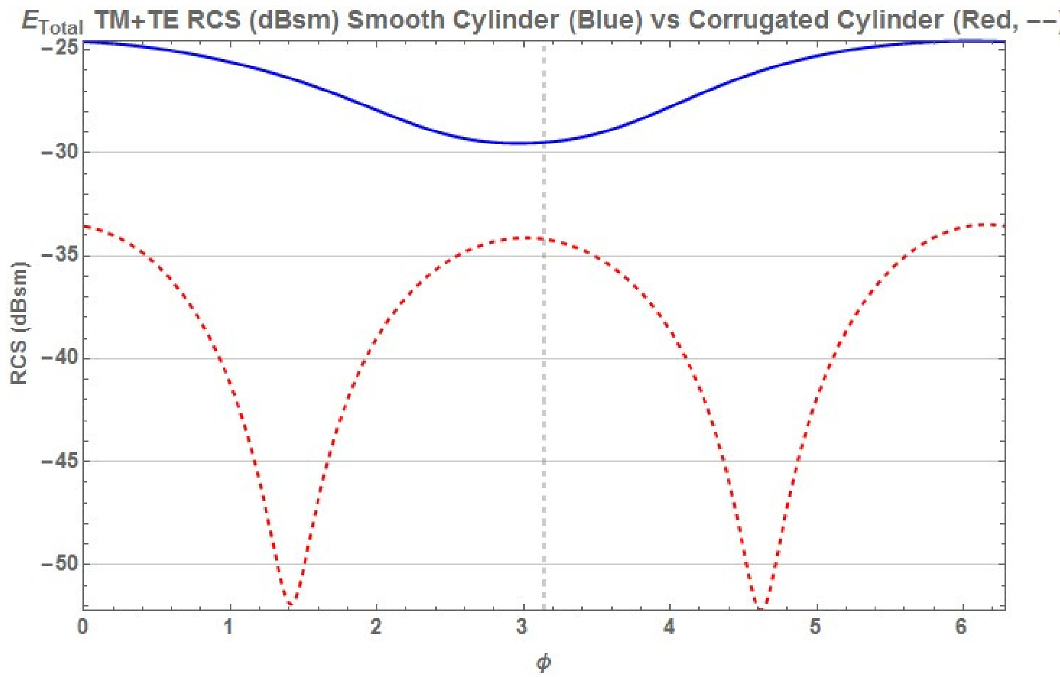


Figure 7-45 XY Plot form of RCS dBsm for E_{Total} of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.0.1.0.0

Table 9 Detailed parameters summary for changing ρ plots of Run a.0.1.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	0.019986 m	-	-	-	-
Frequency	1.50105×10^{10} Hz	-	-	-	-
a	0.0001 λ	-	-	-	-
b	0.1 λ	-	-	-	-
ρ1	0.0999 λ	-	-	-	-
ρ2	0.1 λ	-	-	-	-
ρ range	-	0.08991 λ	10.1 λ	0.0182333 λ	550
ϕ (observed)	37. Deg	-	-	-	-
z (observed)	0.25 a && 0.000025 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
ϕi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.0002	-	-	-	-
max allowable l	0.042685	-	-	-	-
Boundary	Boundary a	-	-	-	-

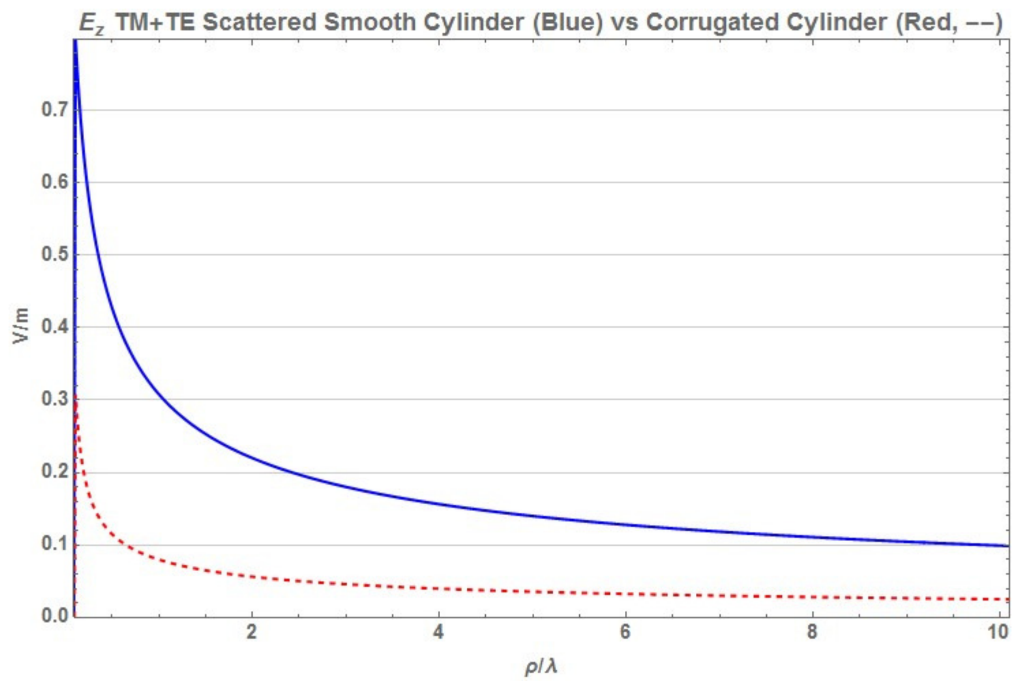


Figure 7-46 XY Plot of Scattered Field Amplitude Only, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.0.10.0.0

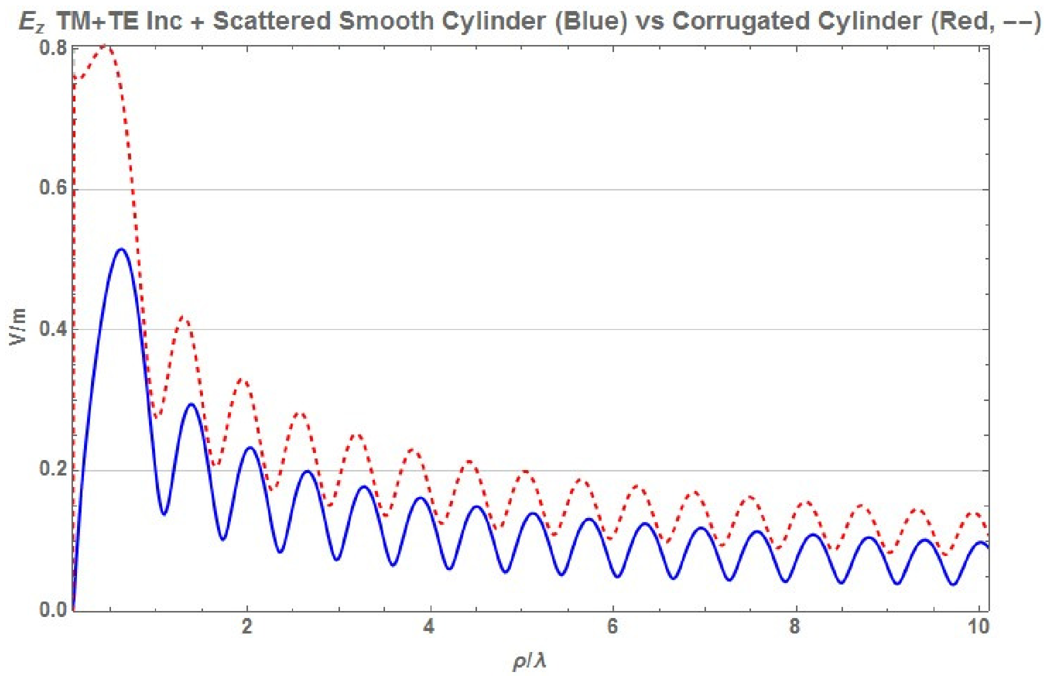
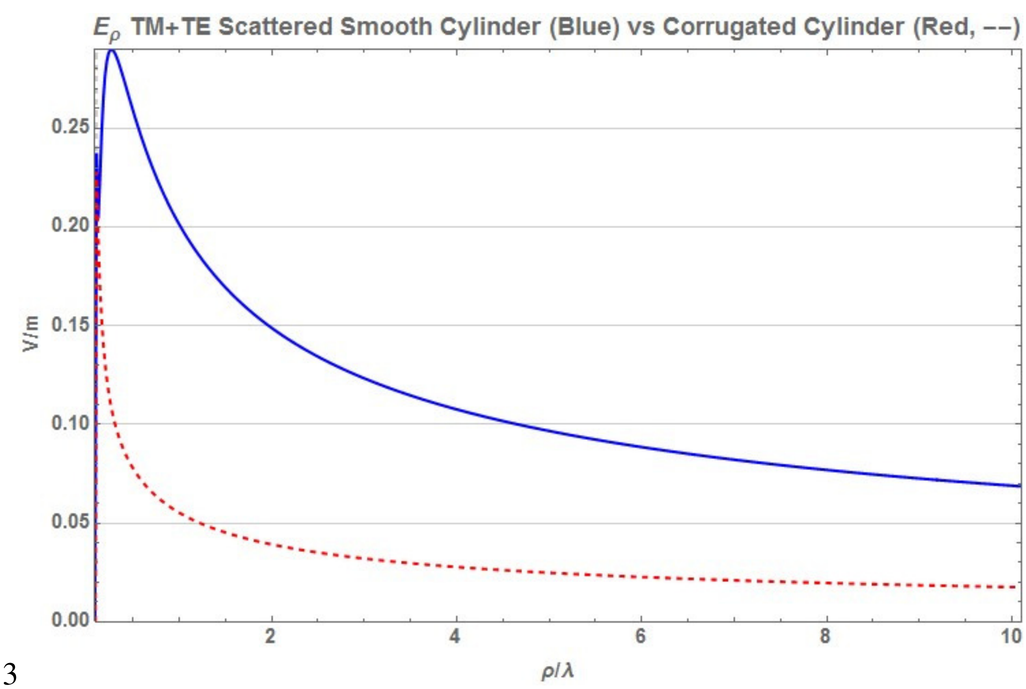


Figure 7-47 XY Plot of Scattered + Incident Field Amplitude Only, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.0.10.0.0



3

Figure 7-48 XY Plot of Scattered Field Amplitude Only, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.0.10.0.0

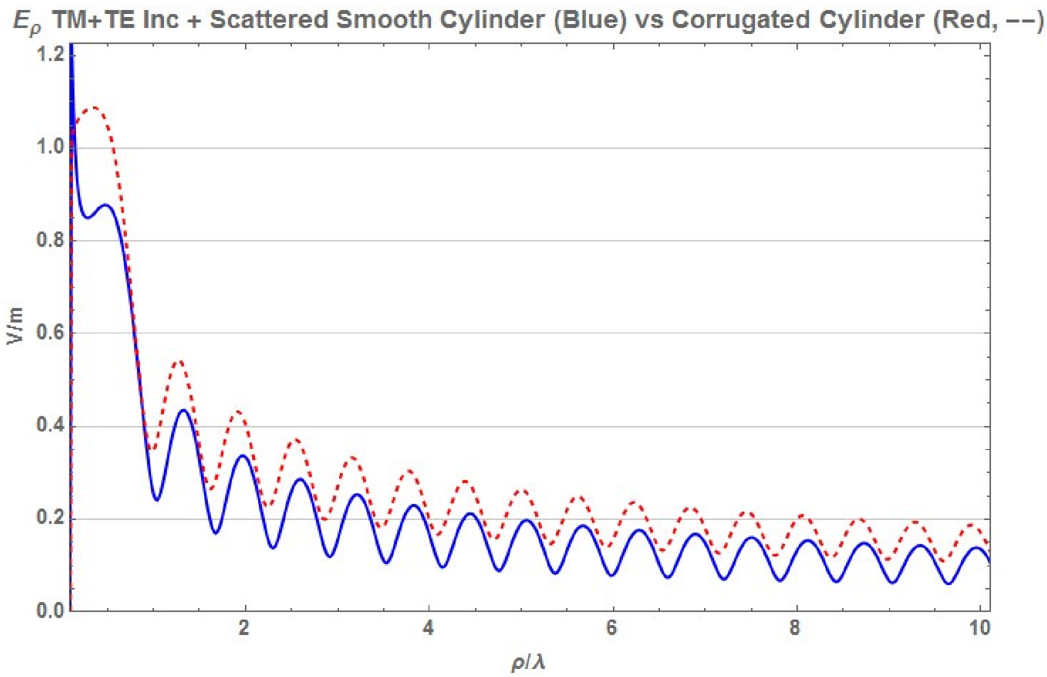


Figure 7-49 XY Plot of Scattered + Incident Field Amplitude Only, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.0.10.0.0

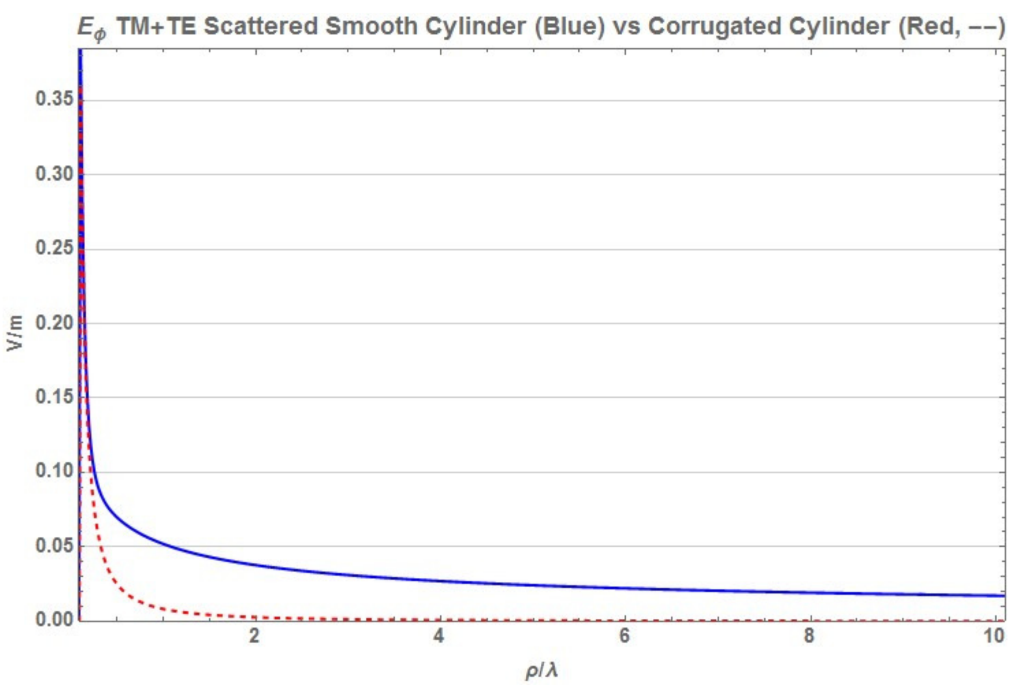


Figure 7-50 XY Plot of Scattered Field Amplitude Only, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.0.10.0.0

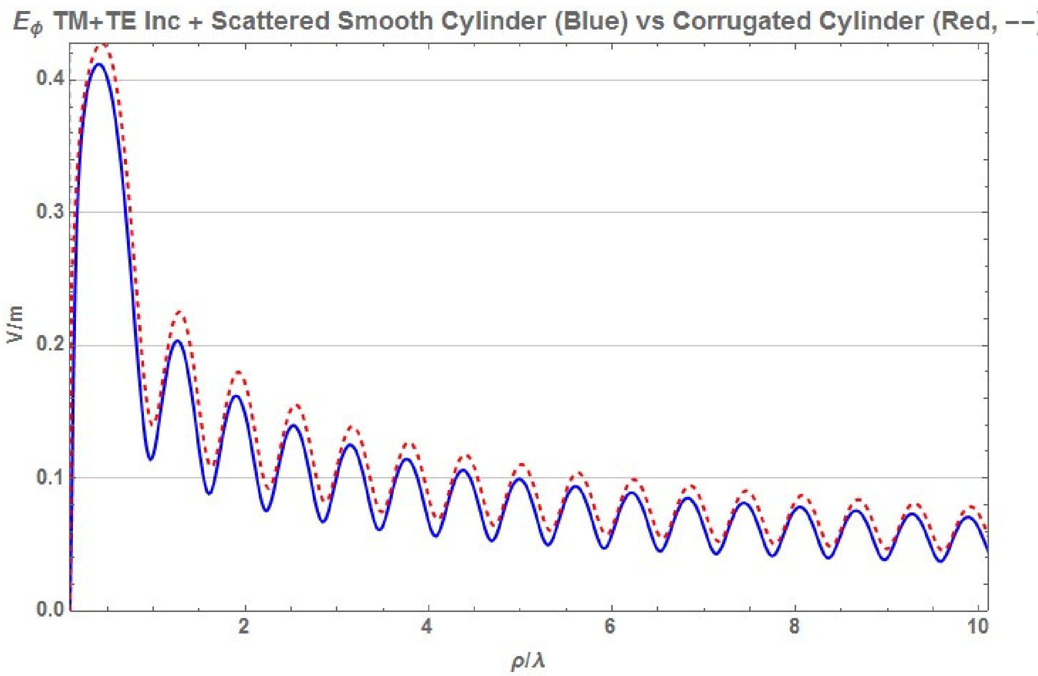


Figure 7-51 XY Plot of Scattered + Incident Field Amplitude Only, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.0.10.0.0

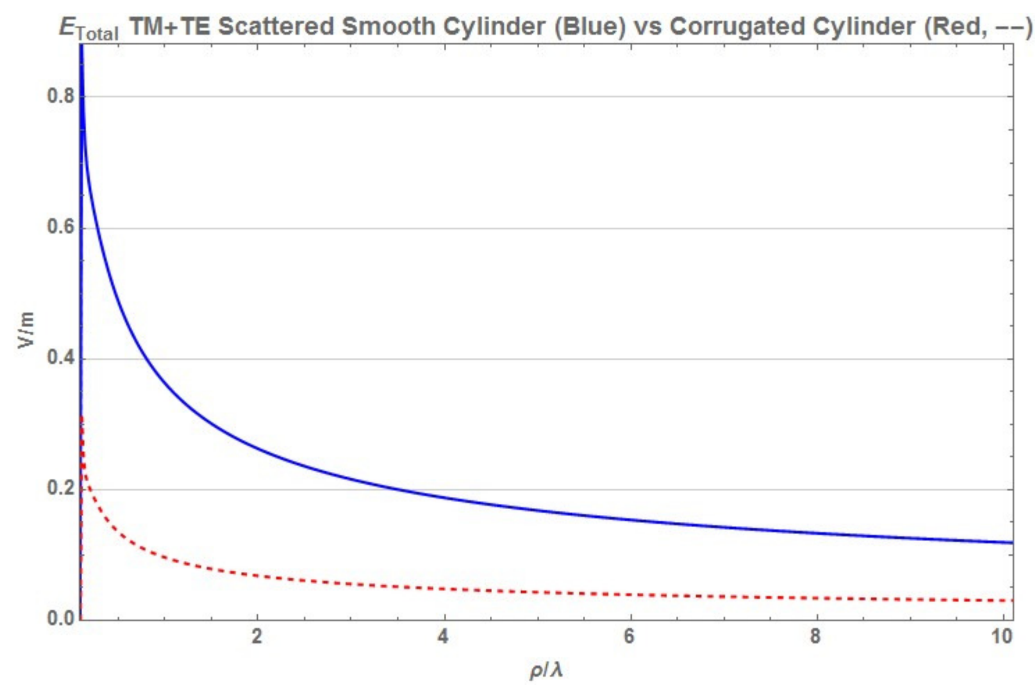


Figure 7-52 XY Plot of Scattered Field Amplitude Only, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.0.10.0.0

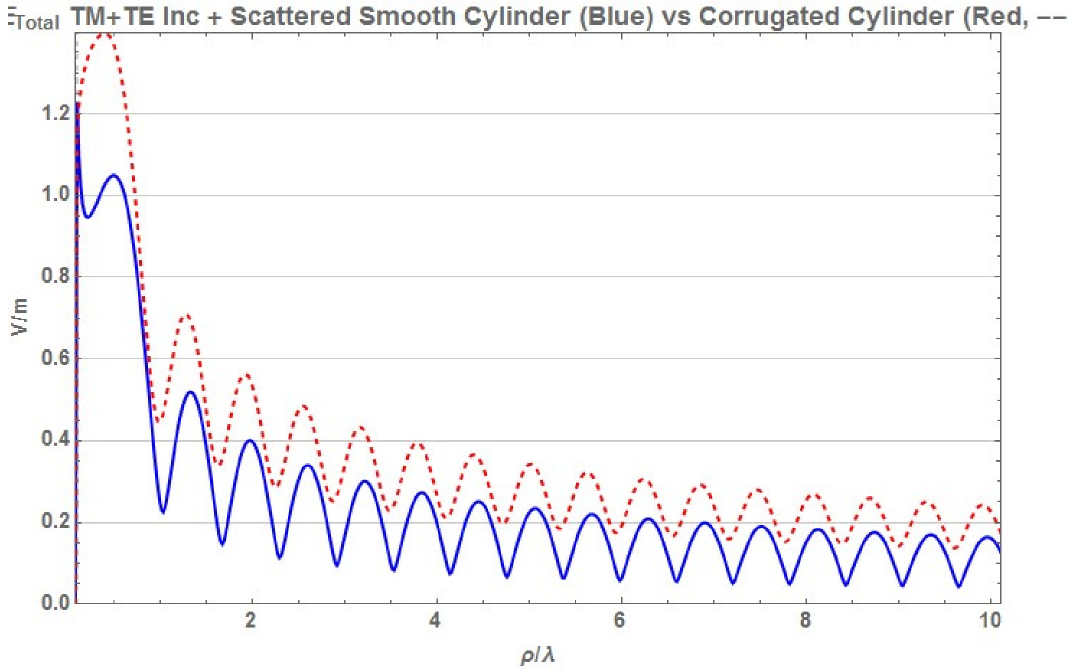


Figure 7-53 XY Plot of Scattered + Incident Field Amplitude Only, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a.0.10.0.0

7.5.4 Run b.20.0.0 (b=20λ, a=b*.001, ρ2=20λ, m=0)

Table 10 Detailed parameters summary for changing φ plots of Run b.20.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	1.50105×10^{10} Hz	-	-	-	-
Frequency	0.019986 m	-	-	-	-
a	0.02 λ	-	-	-	-
b	20. λ	-	-	-	-
ρ1	19.98 λ	-	-	-	-
ρ2	20. λ	-	-	-	-
ϕ range	-	0.327273 Deg	359.673 Deg	0.654545 Deg	550
ρ (observed)	200. λ	-	-	-	-
z (observed)	500.5 a && 10.01 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
ϕi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.04	-	-	-	-
max allowable l	8.537	-	-	-	-
Boundary	Boundary b	-	-	-	-

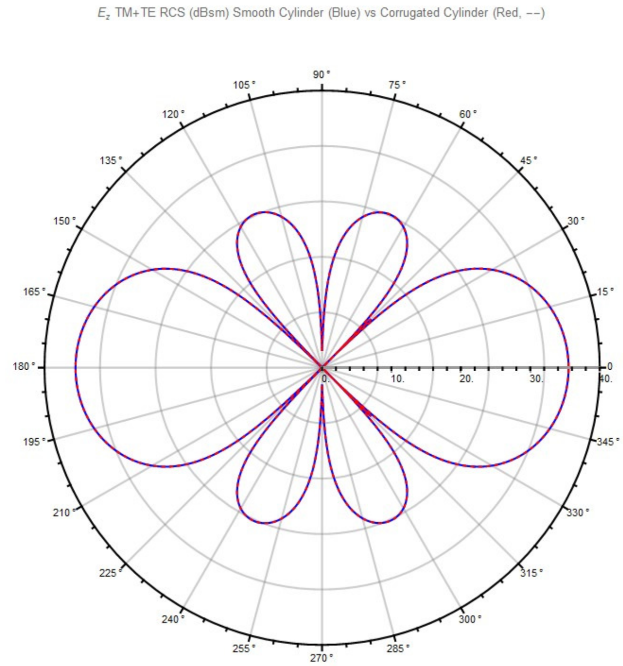


Figure 7-54 Polar Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run b.20.0.0

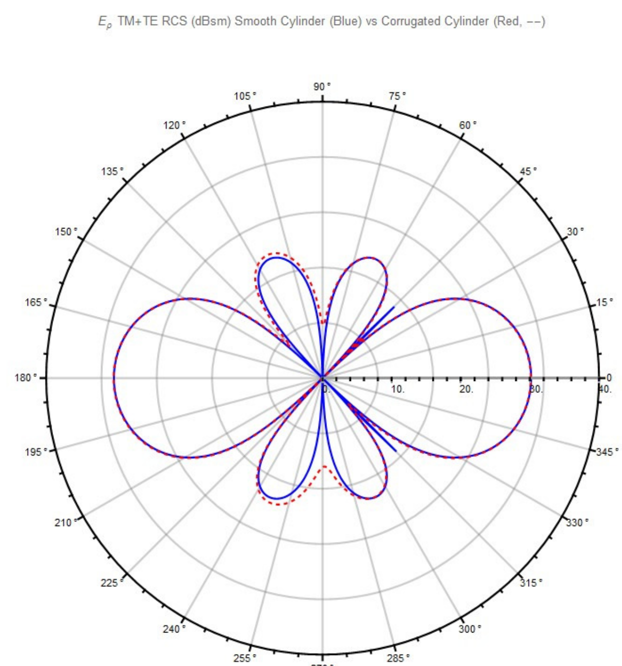


Figure 7-55 Polar Plot form of RCS dBsm for Eρ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run b.20.0.0

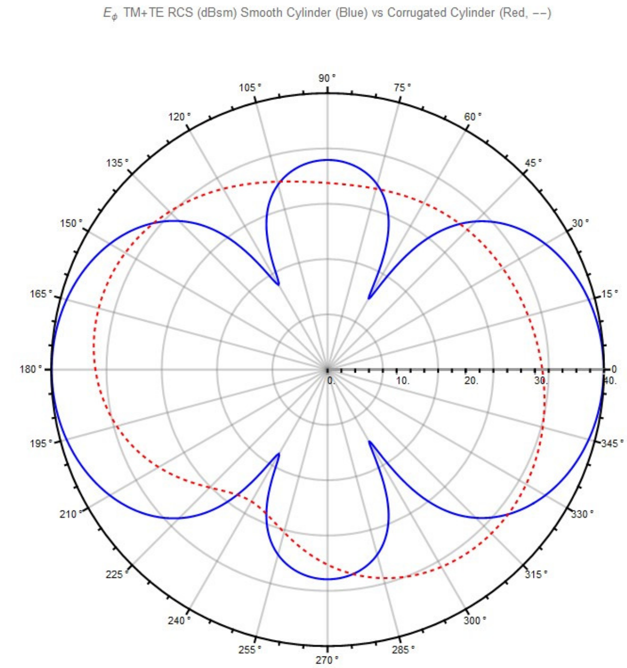


Figure 7-56 Polar Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run b.20.0.0

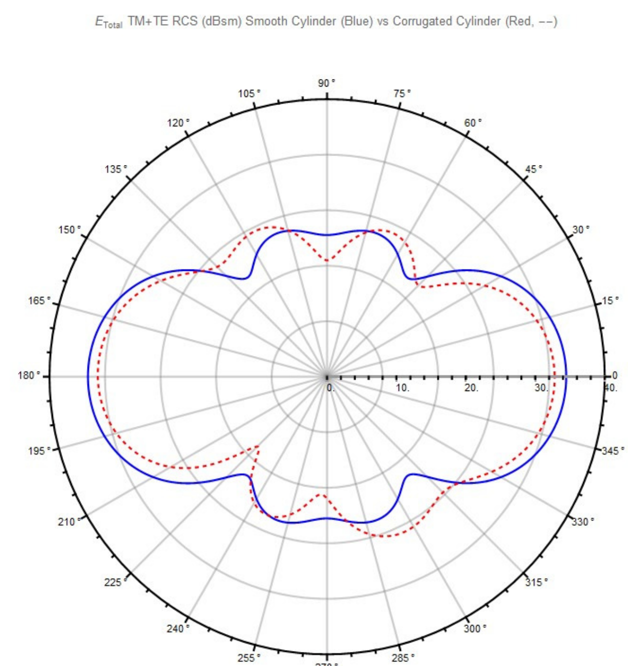


Figure 7-57 Polar Plot form of RCS dBsm for E_{Total} of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.20.0.0

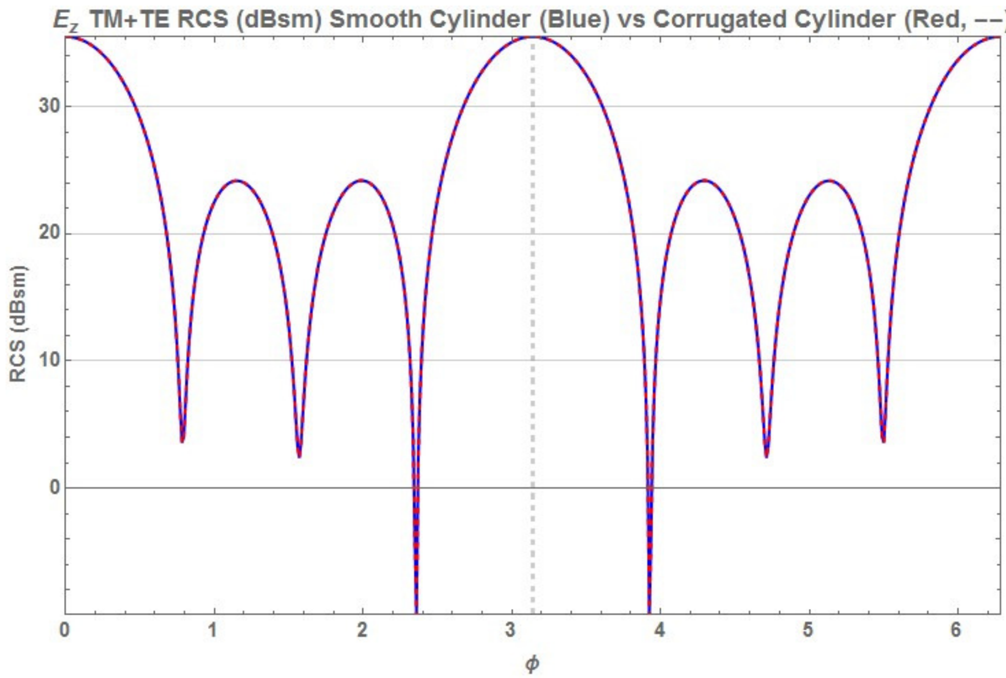


Figure 7-58 XY Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.20.0.0

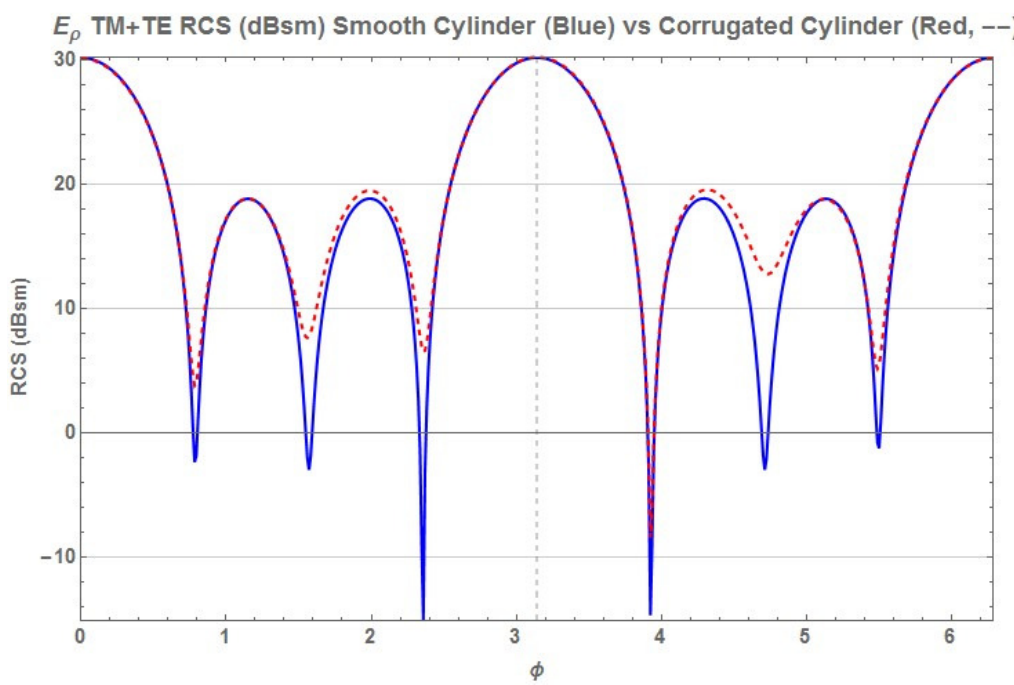


Figure 7-59 XY Plot form of RCS dBsm for E_ρ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.20.0.0

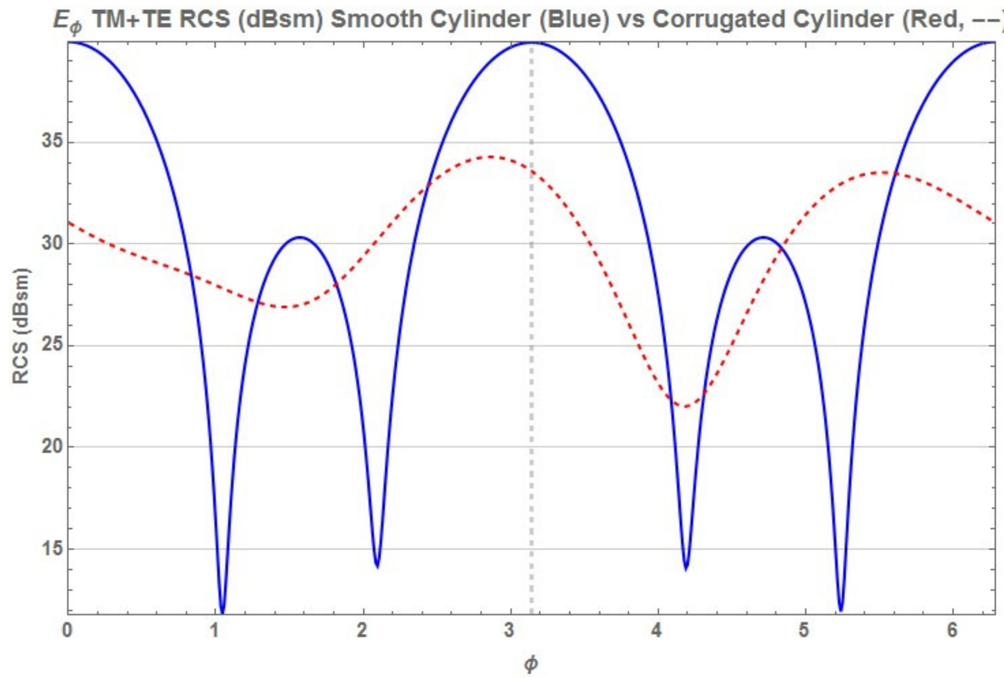


Figure 7-60 XY Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.20.0.0

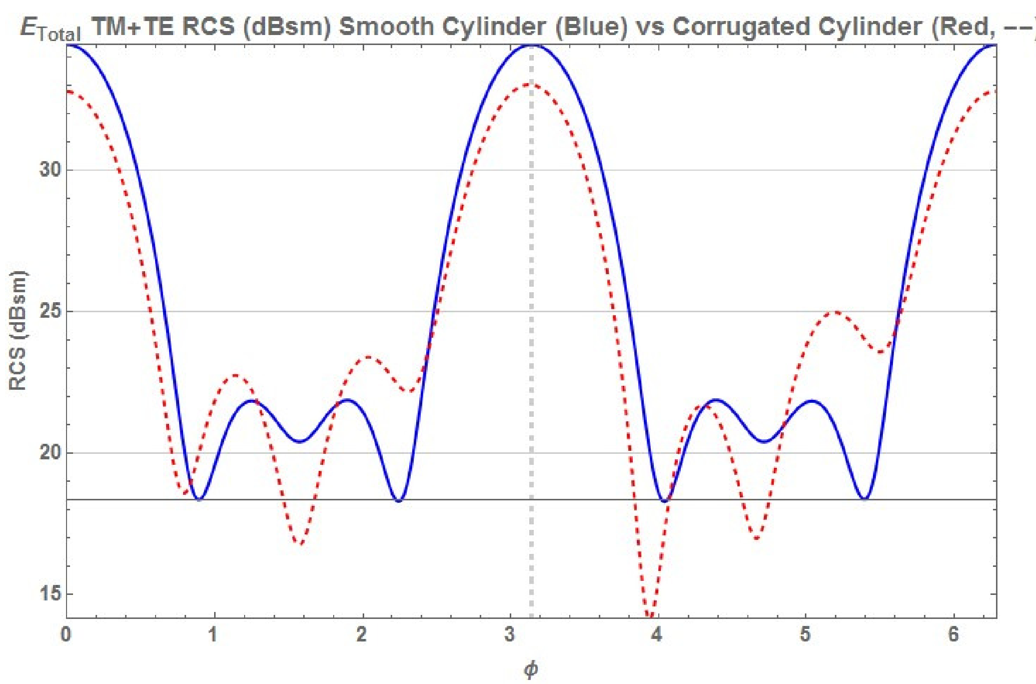


Figure 7-61 XY Plot form of RCS dBsm for E_{Total} of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.20.0.0

Table 11 Detailed parameters summary for changing ρ plots of Run b.20.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	0.019986 m	-	-	-	-
Frequency	1.50105×10^{10} Hz	-	-	-	-
a	0.02 λ	-	-	-	-
b	20. λ	-	-	-	-
ρ1	19.98 λ	-	-	-	-
ρ2	20. λ	-	-	-	-
ρ range	-	17.982 λ	30. λ	0.0218907 λ	550
ϕ (observed)	37. Deg	-	-	-	-
z (observed)	500.5 a && 10.01 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
ϕi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.04	-	-	-	-
max allowable l	8.537	-	-	-	-
Boundary	Boundary b	-	-	-	-

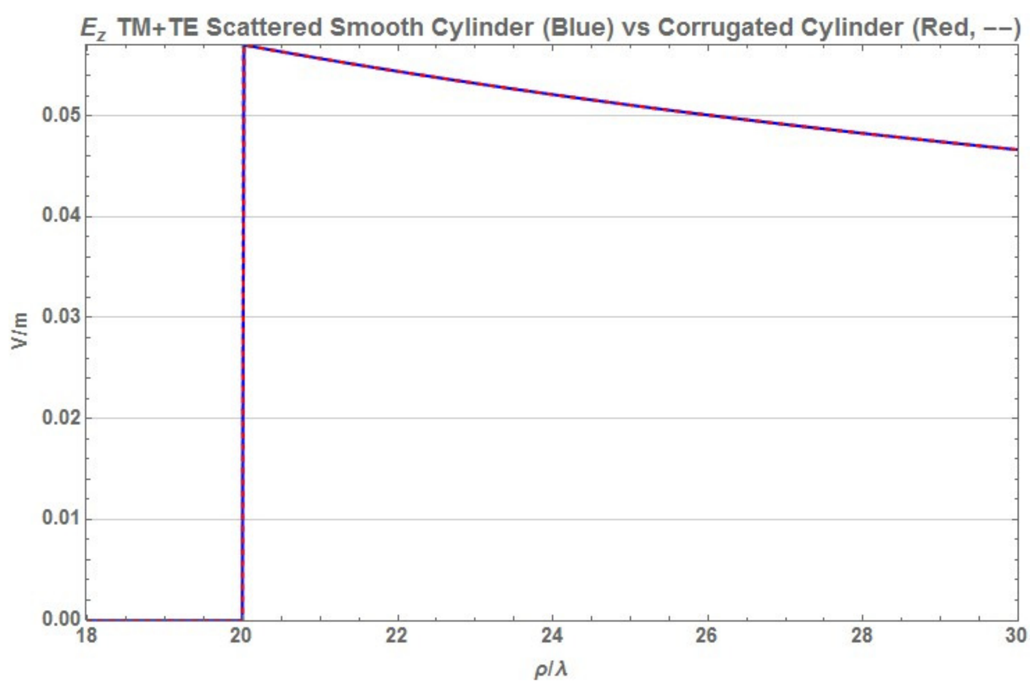


Figure 7-62 XY Plot of Scattered Field Amplitude Only, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.20.0.0

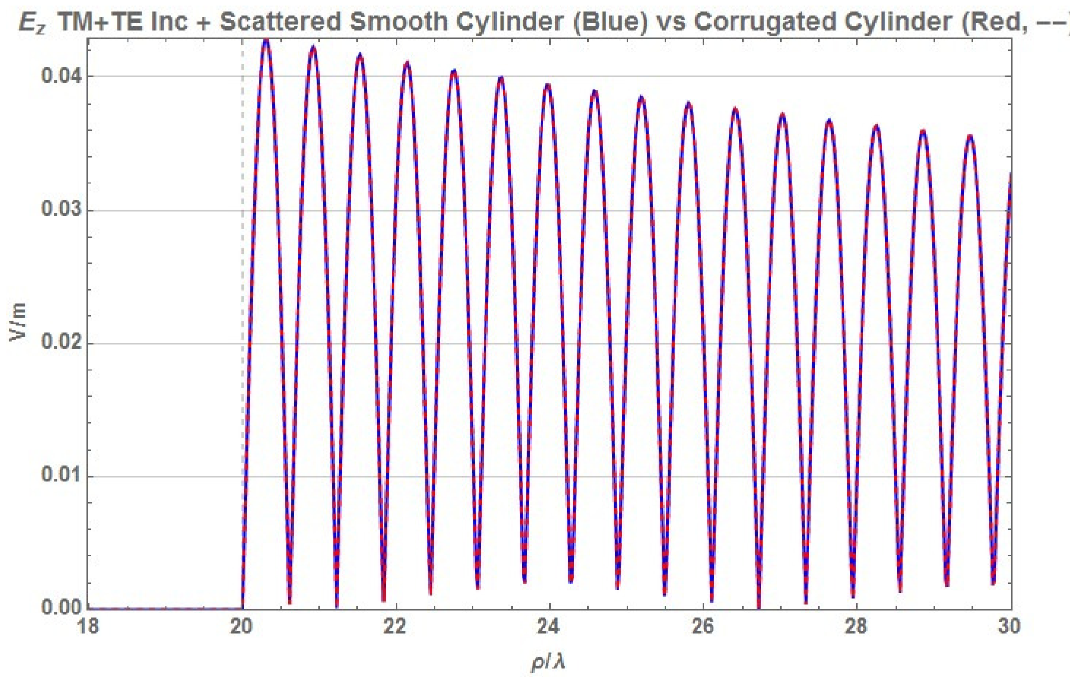


Figure 7-63 XY Plot of Scattered + Incident Field Amplitude, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.20.0.0

Figure 7-64 XY Plot of Scattered Field Amplitude Only, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.20.0.0

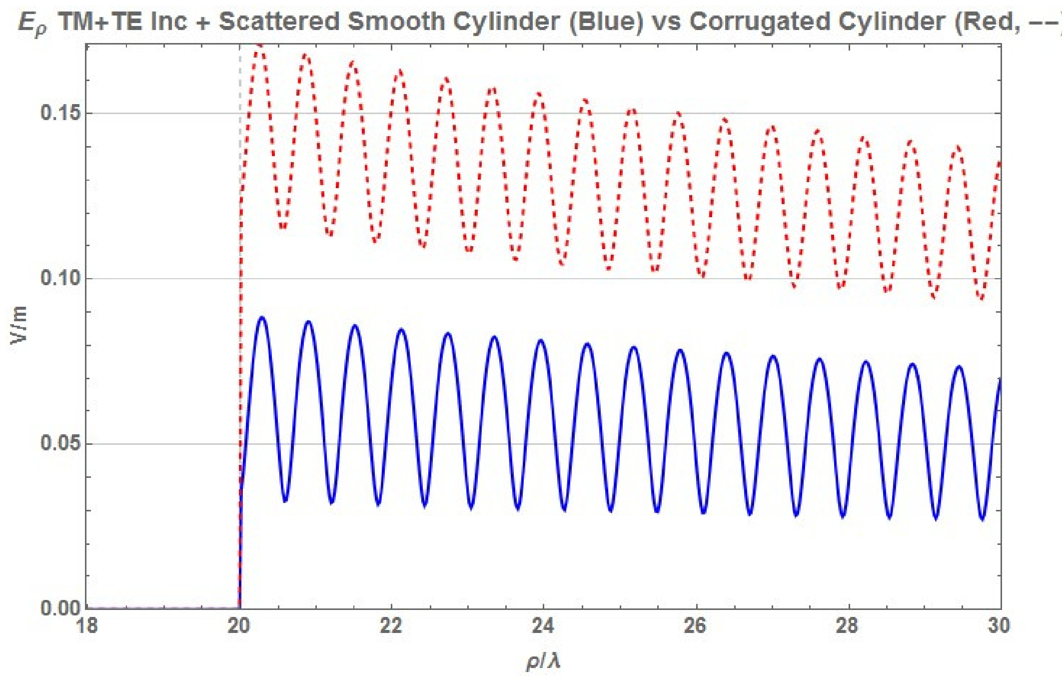


Figure 7-65 XY Plot of Scattered + Incident Field Amplitude, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.20.0.0

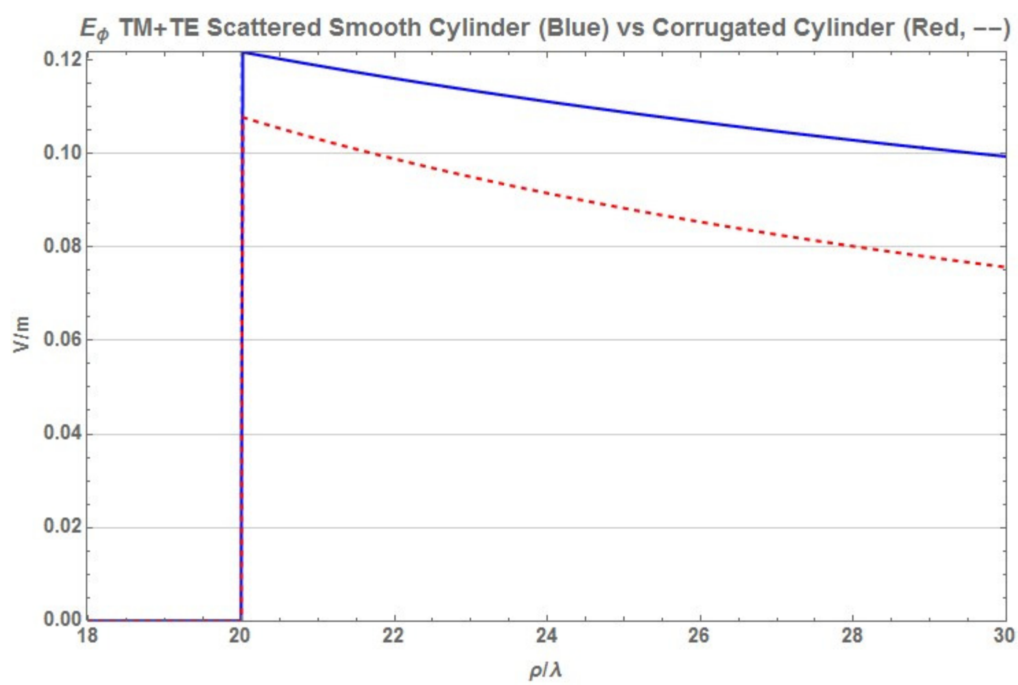


Figure 7-66 XY Plot of Scattered Field Amplitude Only, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.20.0.0

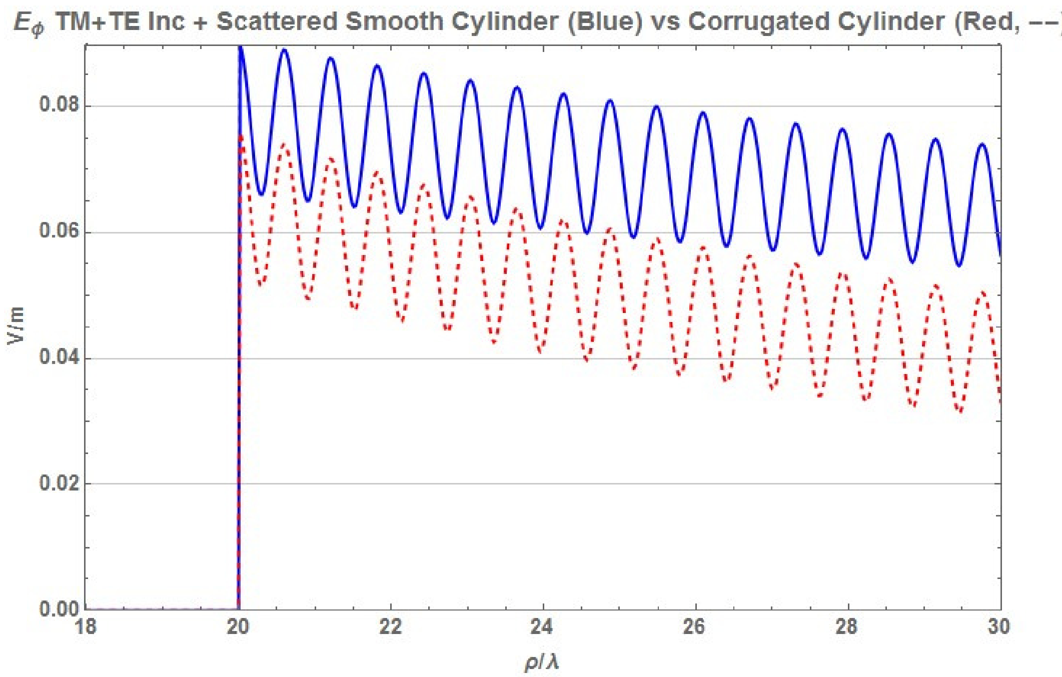


Figure 7-67 XY Plot of Scattered + Incident Field Amplitude, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.20.0.0

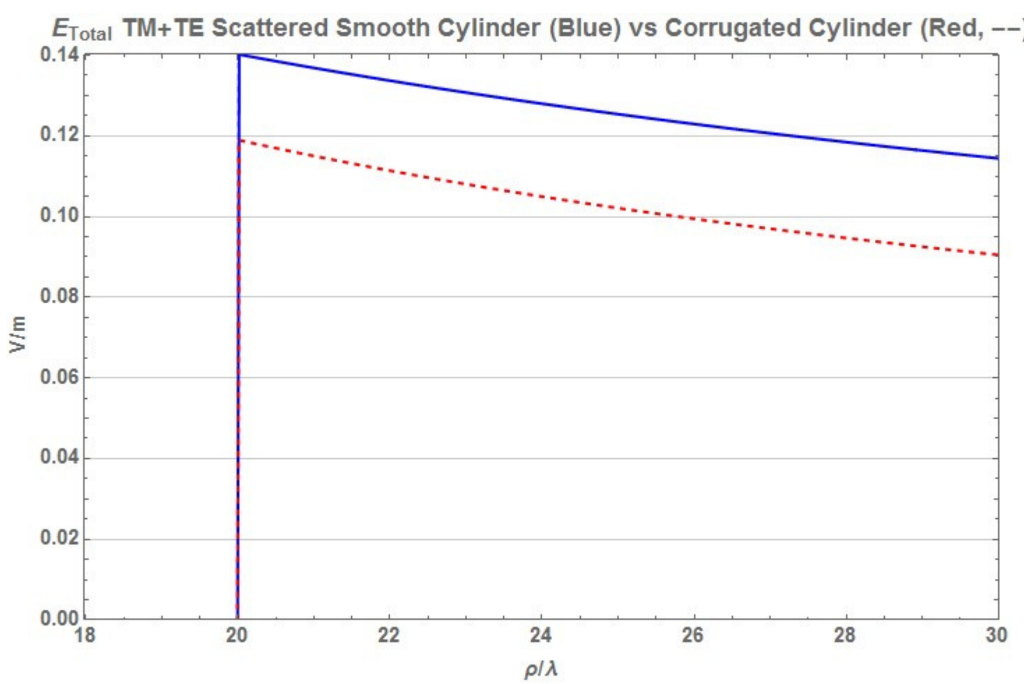


Figure 7-68 XY Plot of Scattered Field Amplitude Only, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.20.0.0

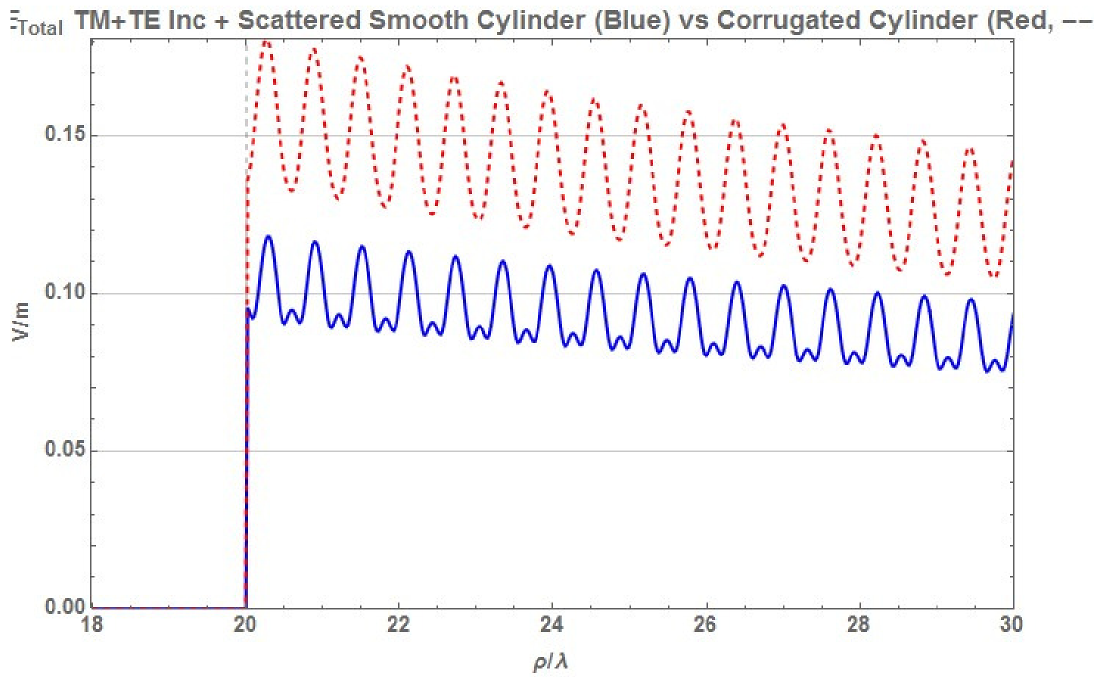


Figure 7-69 XY Plot of Scattered + Incident Field Amplitude, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.20.0.0

7.5.5 Run b.2.0.0 (b=2λ, a=b*.001, ρ2=2λ, m=0)

Table 12 Detailed parameters summary for changing φ plots of Run b.2.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	1.50105×10^{10} Hz	-	-	-	-
Frequency	0.019986 m	-	-	-	-
a	0.002 λ	-	-	-	-
b	2. λ	-	-	-	-
ρ1	1.998 λ	-	-	-	-
ρ2	2. λ	-	-	-	-
ϕ range	-	0.327273 Deg	359.673 Deg	0.654545 Deg	550
ρ (observed)	20. λ	-	-	-	-
z (observed)	500.5 a && 1.001 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
ϕi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.004	-	-	-	-
max allowable l	0.8537	-	-	-	-
Boundary	Boundary b	-	-	-	-

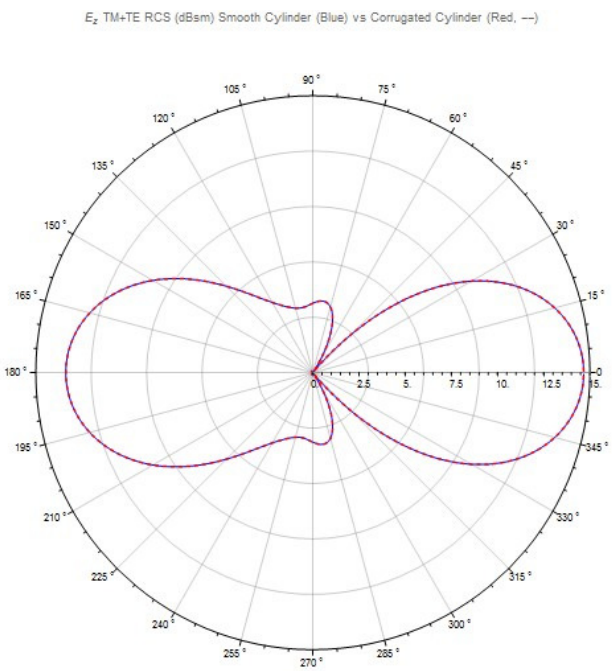


Figure 7-70 Polar Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run b.2.0.0

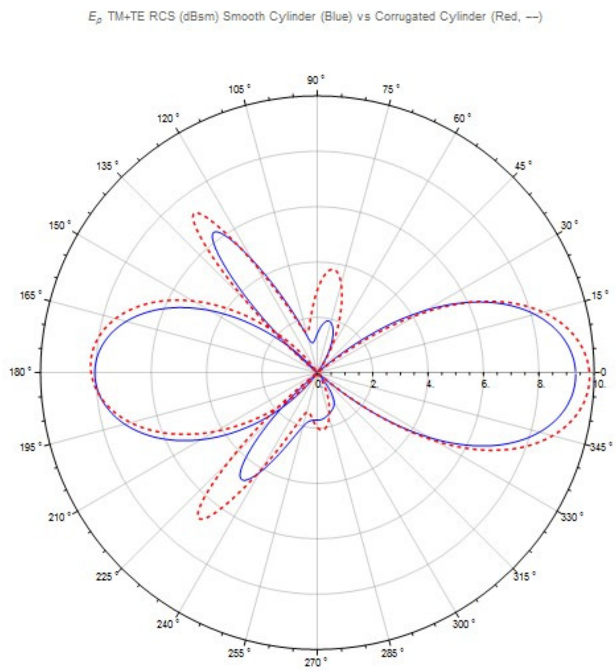


Figure 7-71 Polar Plot form of RCS dBsm for Eρ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run b.2.0.0

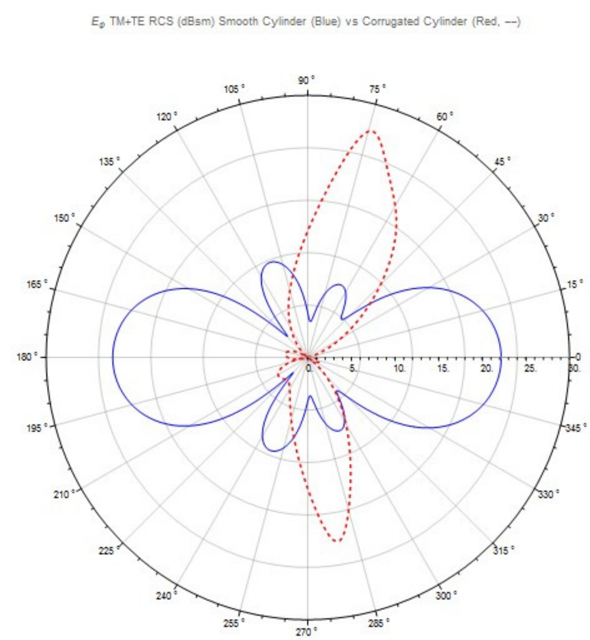


Figure 7-72 Polar Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run b.2.0.0

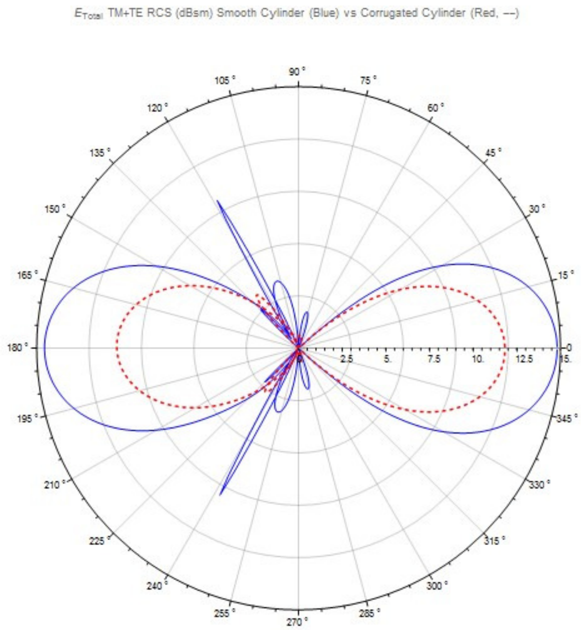


Figure 7-73 Polar Plot form of RCS dBsm for ETotal of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.2.0.0

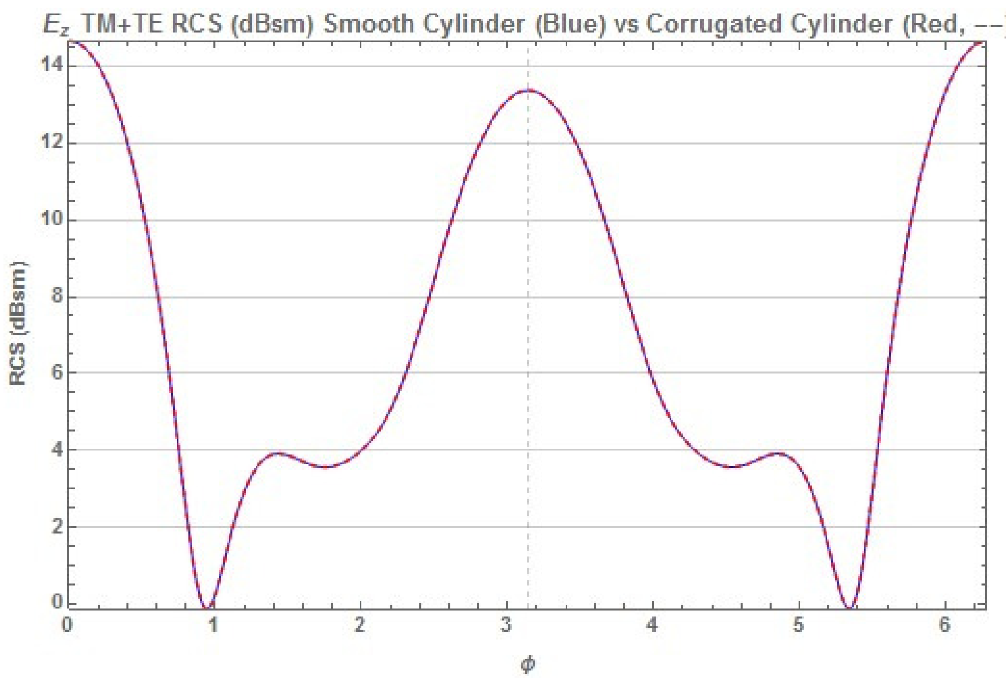


Figure 7-74 XY Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.2.0.0

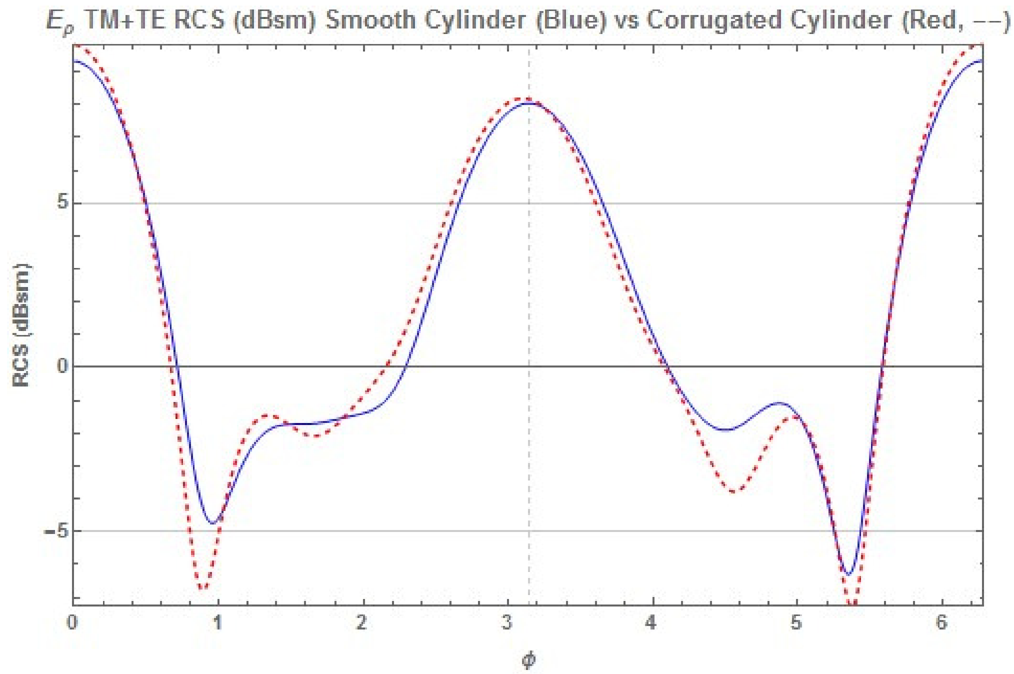


Figure 7-75 XY Plot form of RCS dBsm for Eρ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.2.0.0

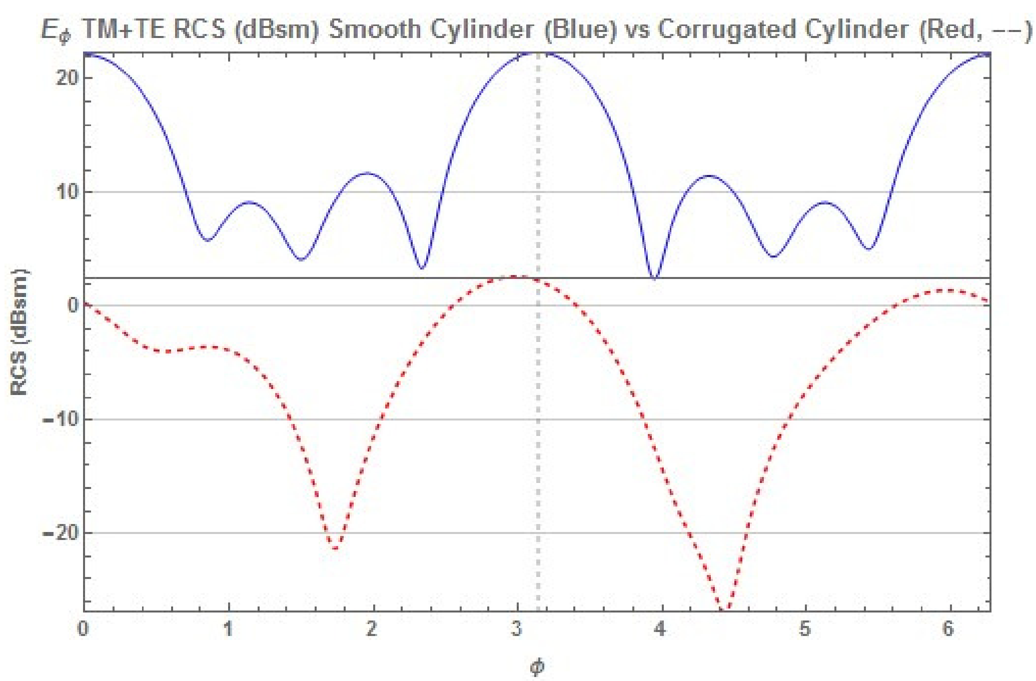


Figure 7-76 XY Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.2.0.0

.

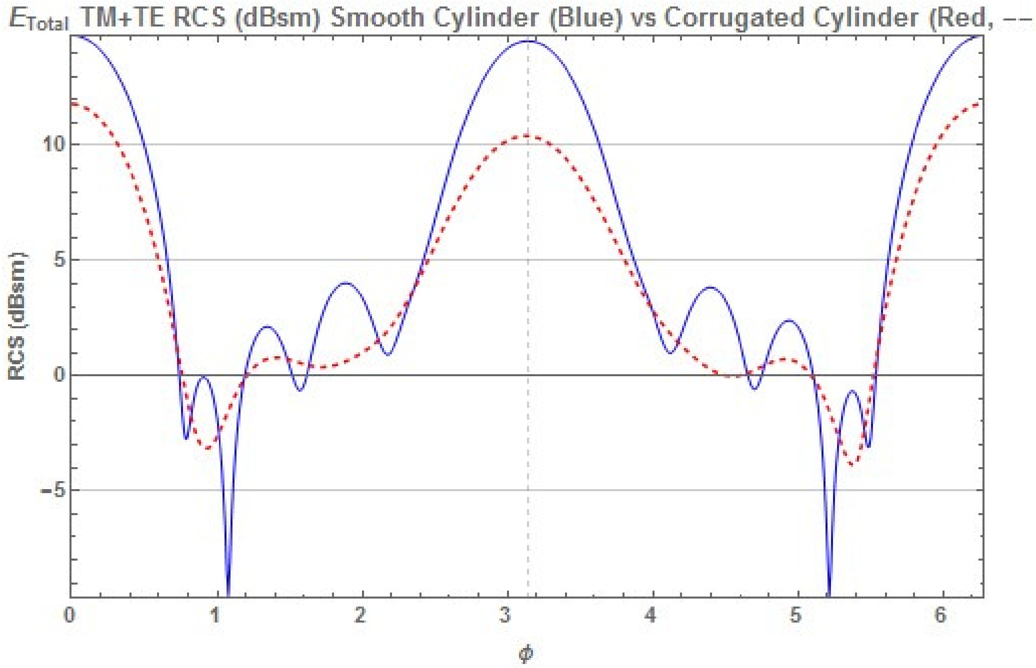


Figure 7-77 XY Plot form of RCS dBsm for ETotal of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.2.0.0

Table 13 Detailed parameters summary for changing ρ plots of Run b.2.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	0.019986 m	-	-	-	-
Frequency	1.50105×10^{10} Hz	-	-	-	-
a	0.002 λ	-	-	-	-
b	2. λ	-	-	-	-
ρ1	1.998 λ	-	-	-	-
ρ2	2. λ	-	-	-	-
ρ range	-	1.7982 λ	12. λ	0.0185825 λ	550
ϕ (observed)	37. Deg	-	-	-	-
z (observed)	500.5 a && 1.001 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
ϕi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.004	-	-	-	-
max allowable l	0.8537	-	-	-	-
Boundary	Boundary b	-	-	-	-

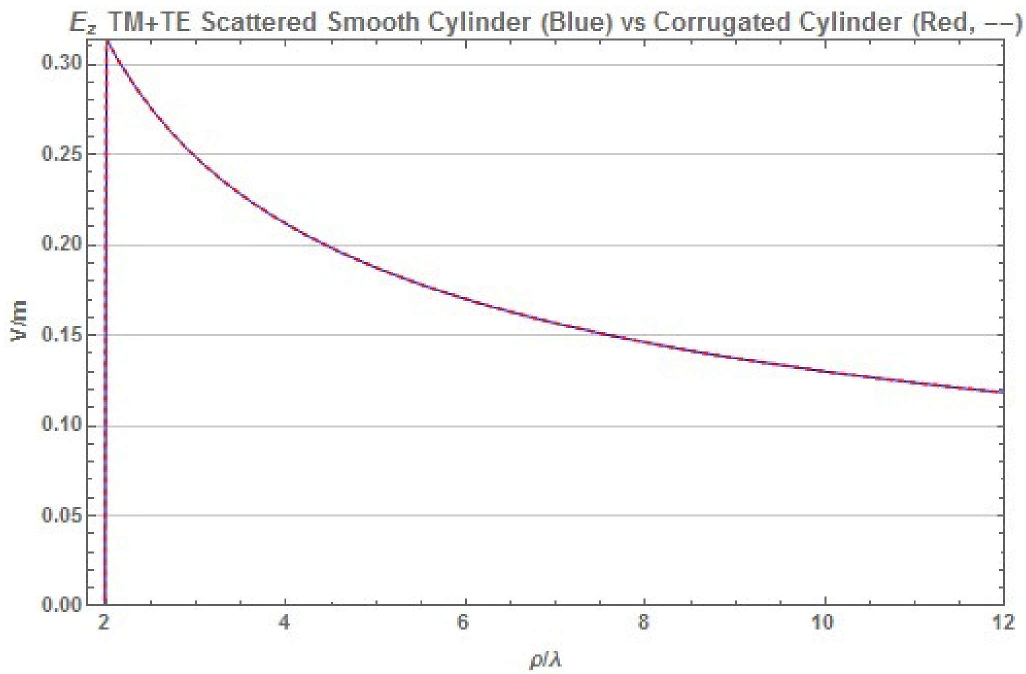


Figure 7-78 XY Plot of Scattered Field Amplitude Only, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.2.0.0

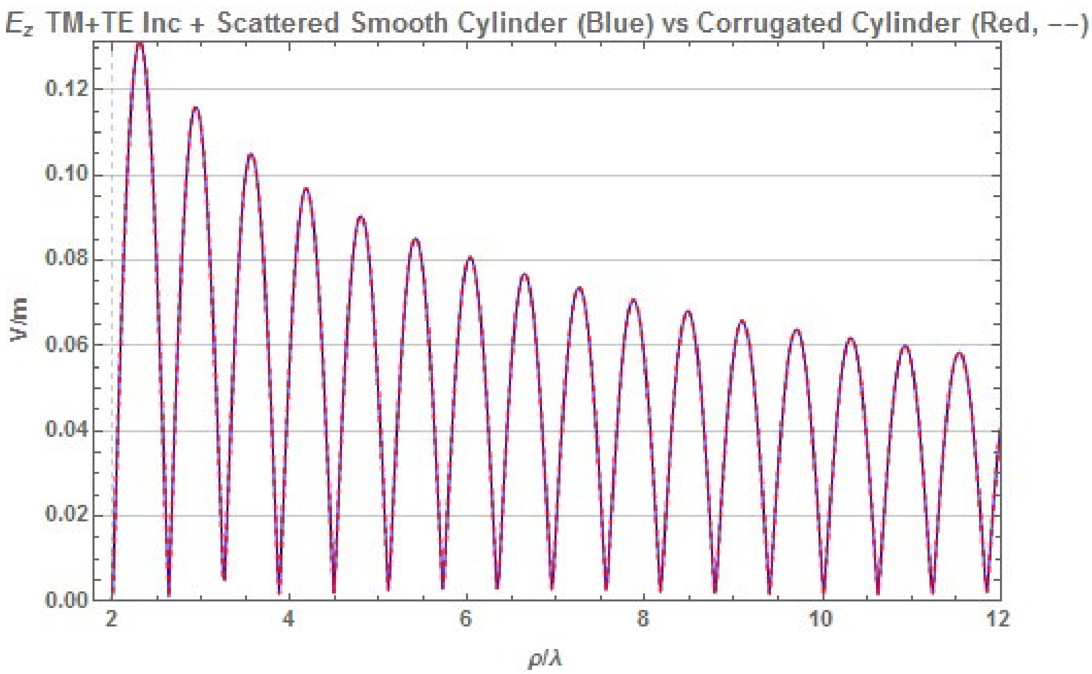


Figure 7-79 XY Plot of Scattered + Incident Field Amplitude, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.2.0.0

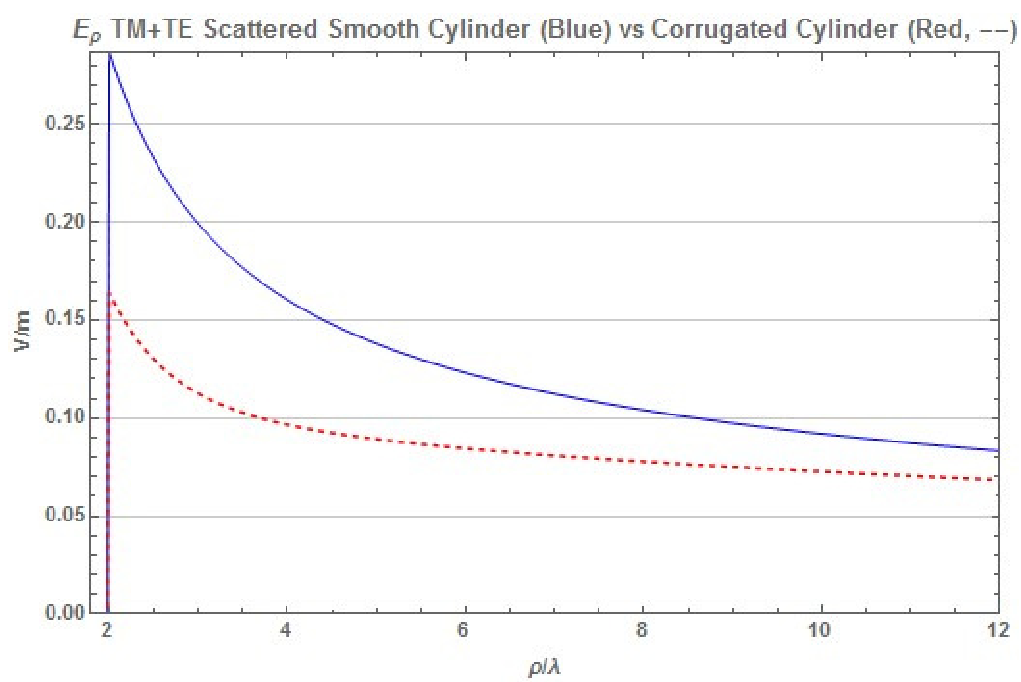


Figure 7-80 XY Plot of Scattered Field Amplitude Only, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.2.0.0

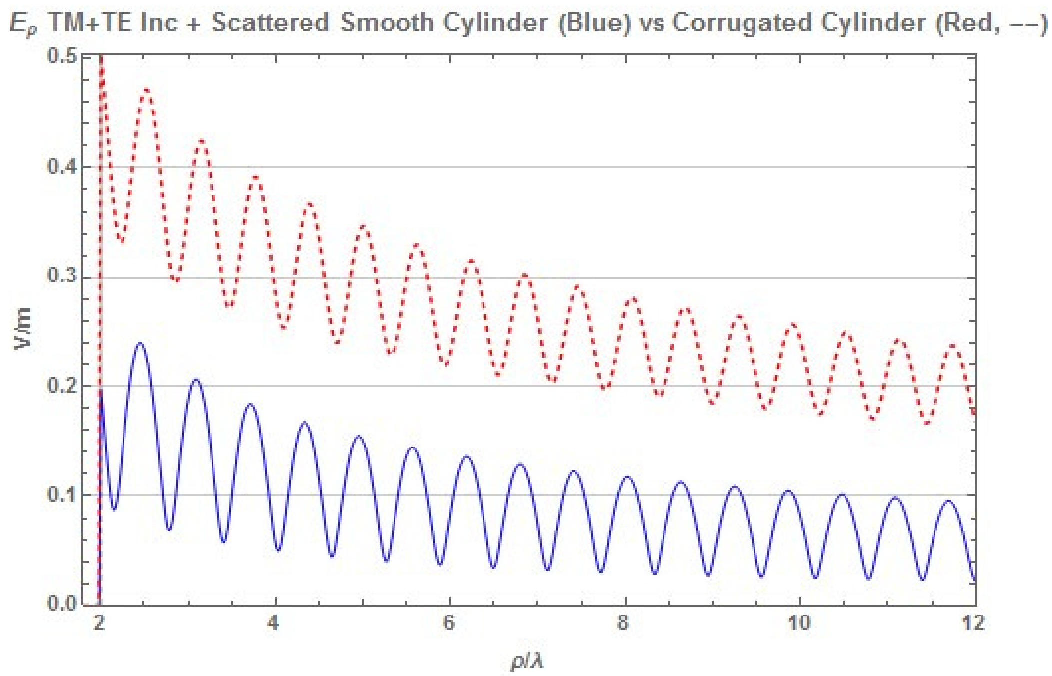


Figure 7-81 XY Plot of Scattered + Incident Field Amplitude, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.2.0.0

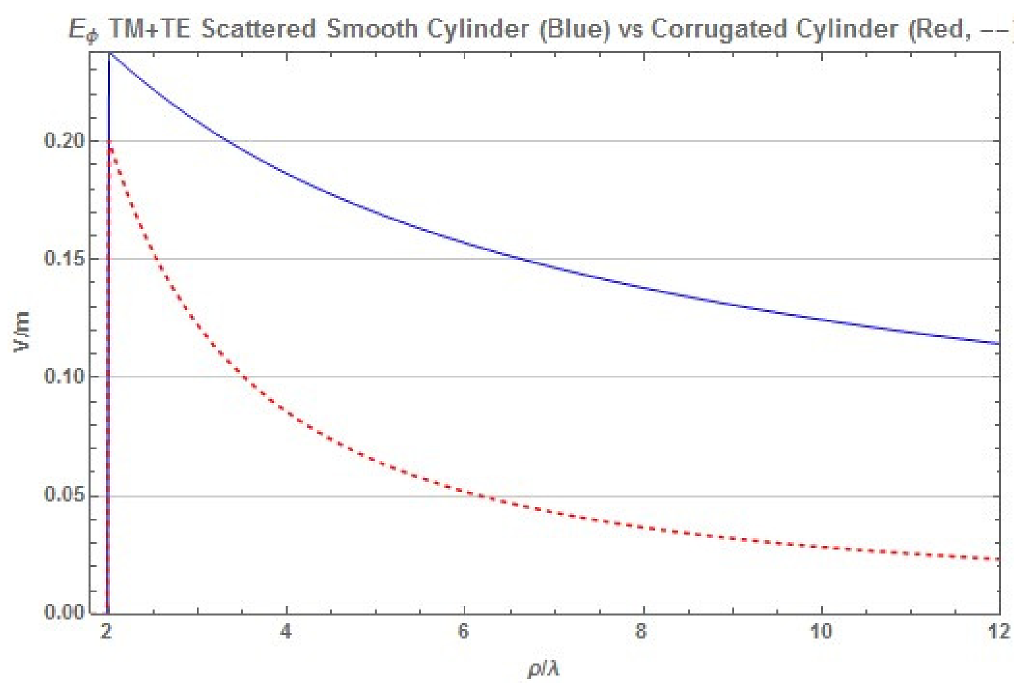


Figure 7-82 XY Plot of Scattered Field Amplitude Only, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.2.0.0

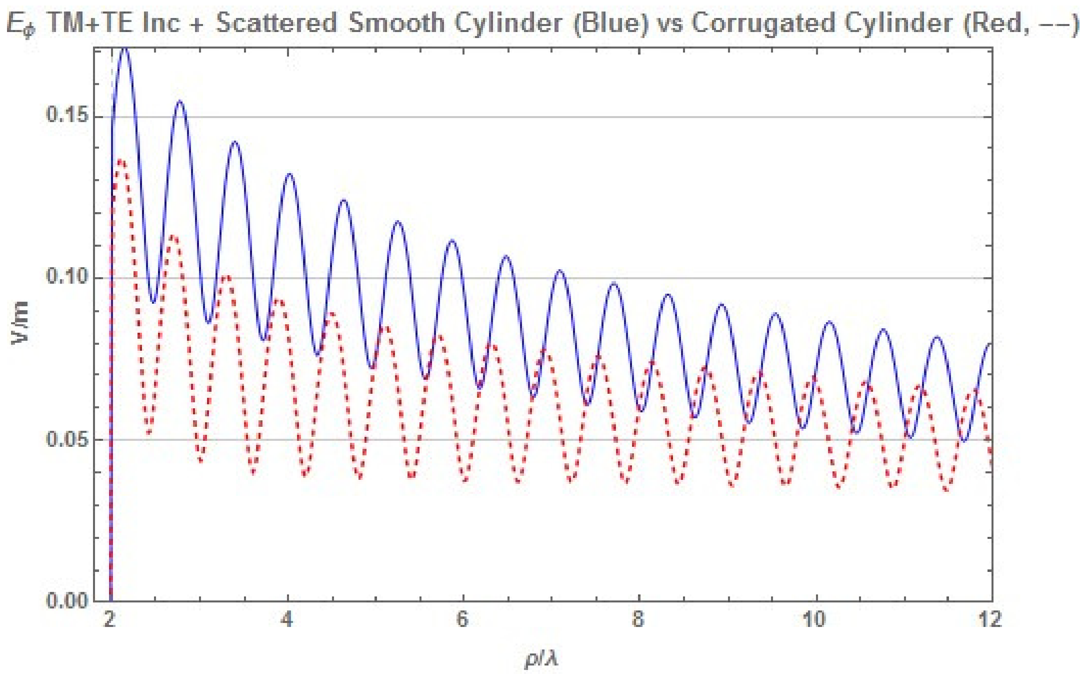


Figure 7-83 XY Plot of Scattered + Incident Field Amplitude, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.2.0.0

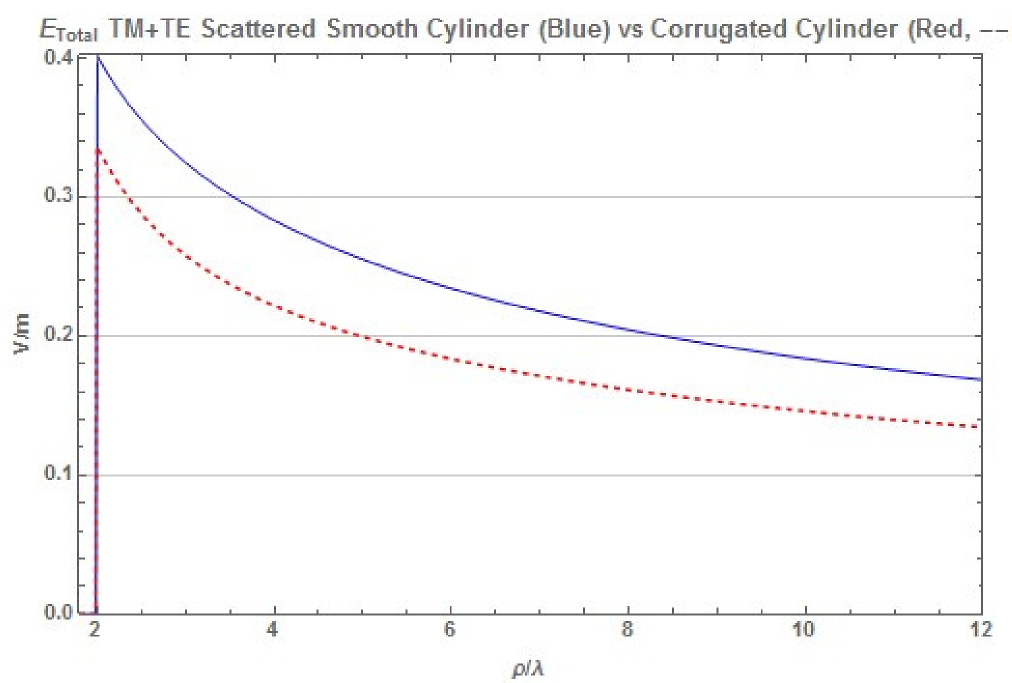


Figure 7-84 XY Plot of Scattered Field Amplitude Only, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.2.0.0

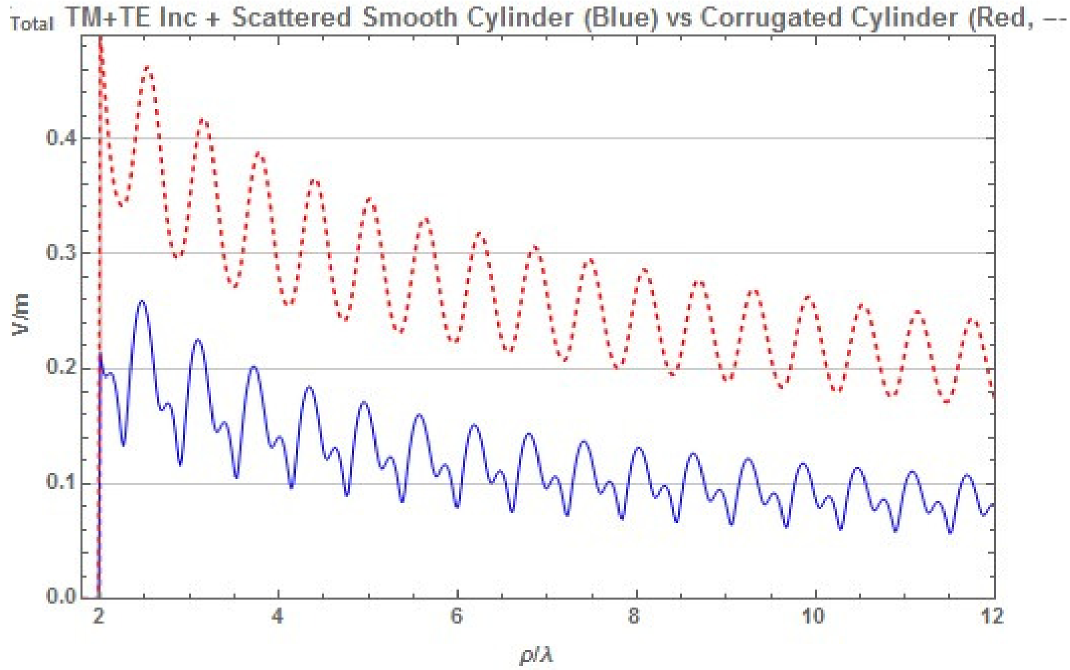


Figure 7-85 XY Plot of Scattered + Incident Field Amplitude, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.2.0.0

7.5.6 Run b.0.1.0.0 (b=0.1λ, a=b*.001, ρ2=0.1λ, m=0)

Table 14 Detailed parameters summary for changing φ plots of Run b.0.1.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	–	–	–	–
H0	0.00265258 A/m	–	–	–	–
λ	1.50105×10^{10} Hz	–	–	–	–
Frequency	0.019986 m	–	–	–	–
a	0.0001 λ	–	–	–	–
b	0.1 λ	–	–	–	–
ρ1	0.0999 λ	–	–	–	–
ρ2	0.1 λ	–	–	–	–
ϕ range	–	0.327273 Deg	359.673 Deg	0.654545 Deg	550
ρ (observed)	1. λ	–	–	–	–
z (observed)	500.5 a && 0.05005 λ	–	–	–	–
Matching Points	–	–	–	–	7
θi	55. Deg	–	–	–	–
ϕi	37. Deg	–	–	–	–
n	–	-3	3	1	7
l	–	0	0	1	1
m	–	0	0	1	1
max allowable m	0.0002	–	–	–	–
max allowable l	0.042685	–	–	–	–
Boundary	Boundary b	–	–	–	–

2

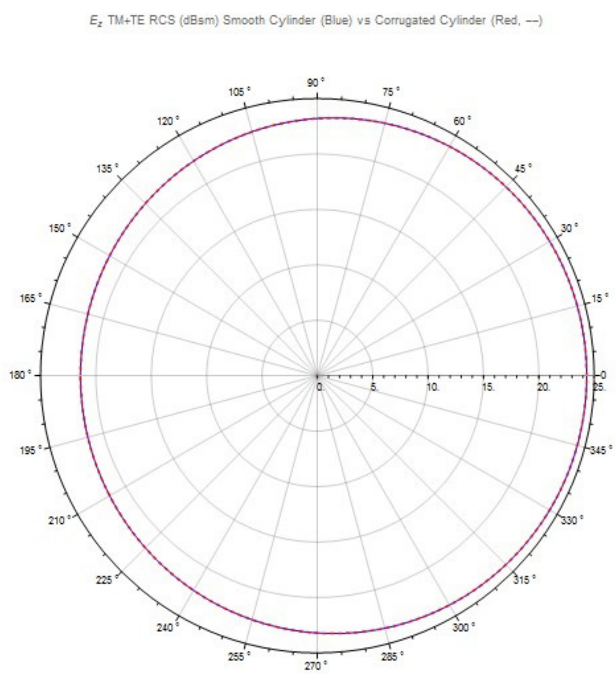


Figure 7-86 Polar Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run b.0.1.0.0

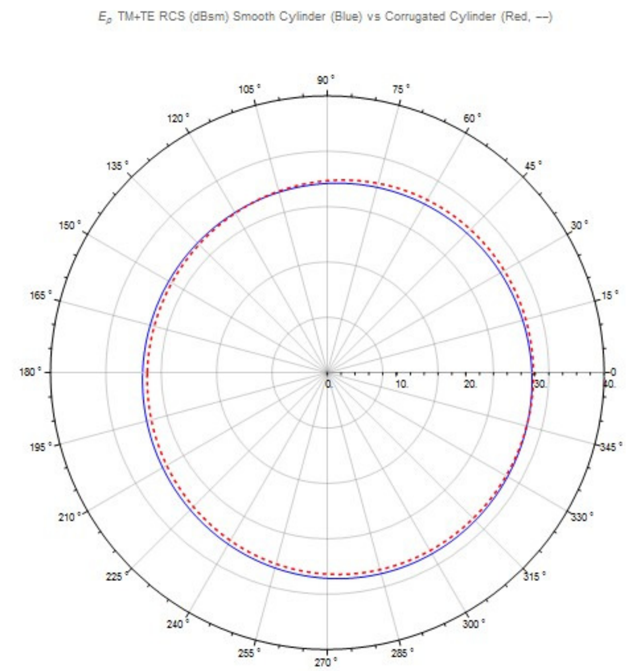


Figure 7-87 Polar Plot form of RCS dBsm for Eρ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run b.0.1.0.0

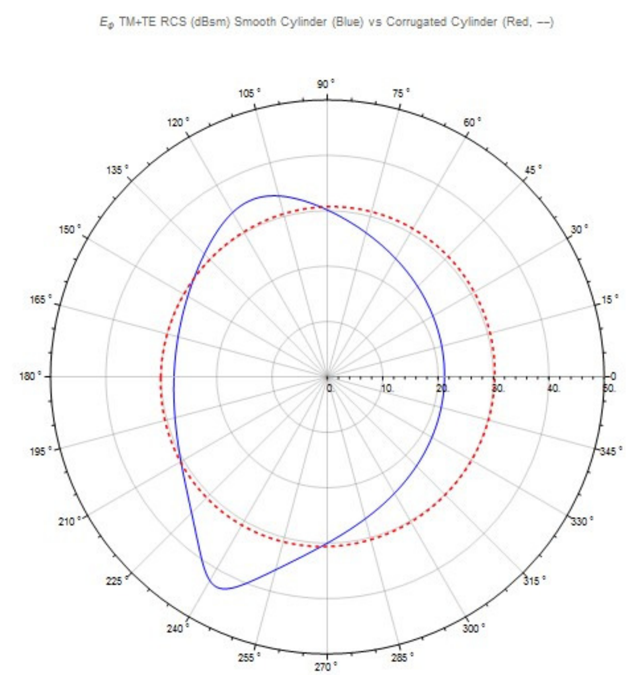


Figure 7-88 Polar Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run b.0.1.0.0

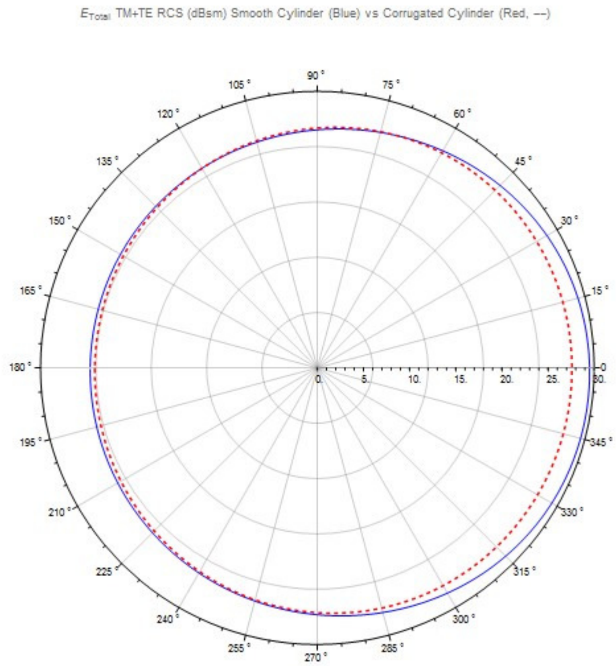


Figure 7-89 Polar Plot form of RCS dBsm for ETotal of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.0.1.0.0

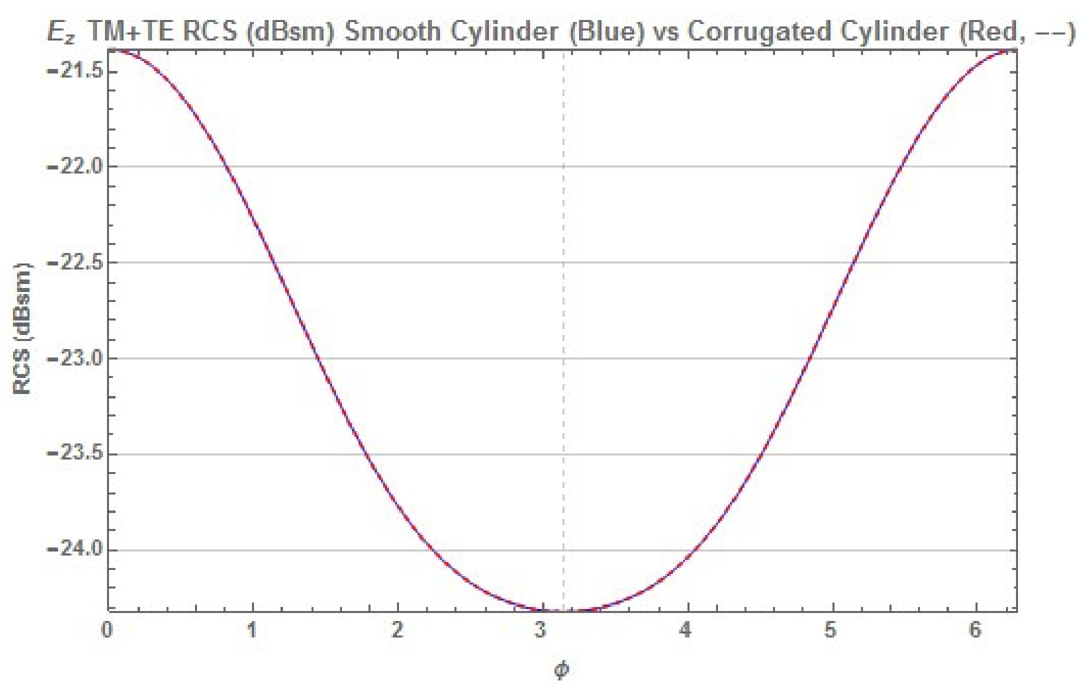


Figure 7-90 XY Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.0.1.0.0

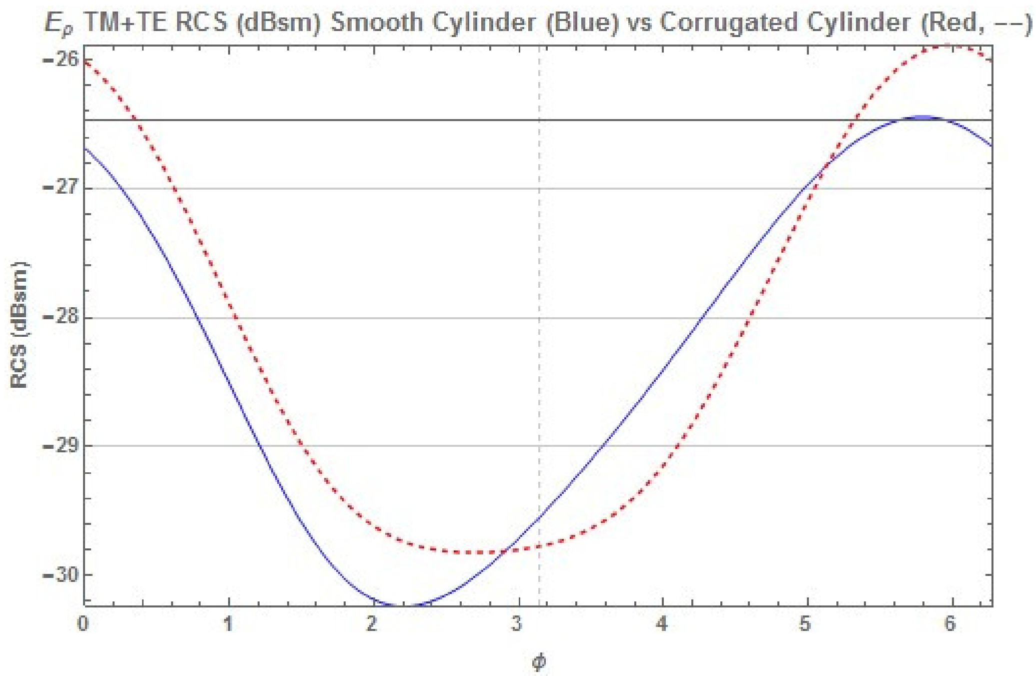


Figure 7-91 XY Plot form of RCS dBsm for Eρ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.0.1.0.0

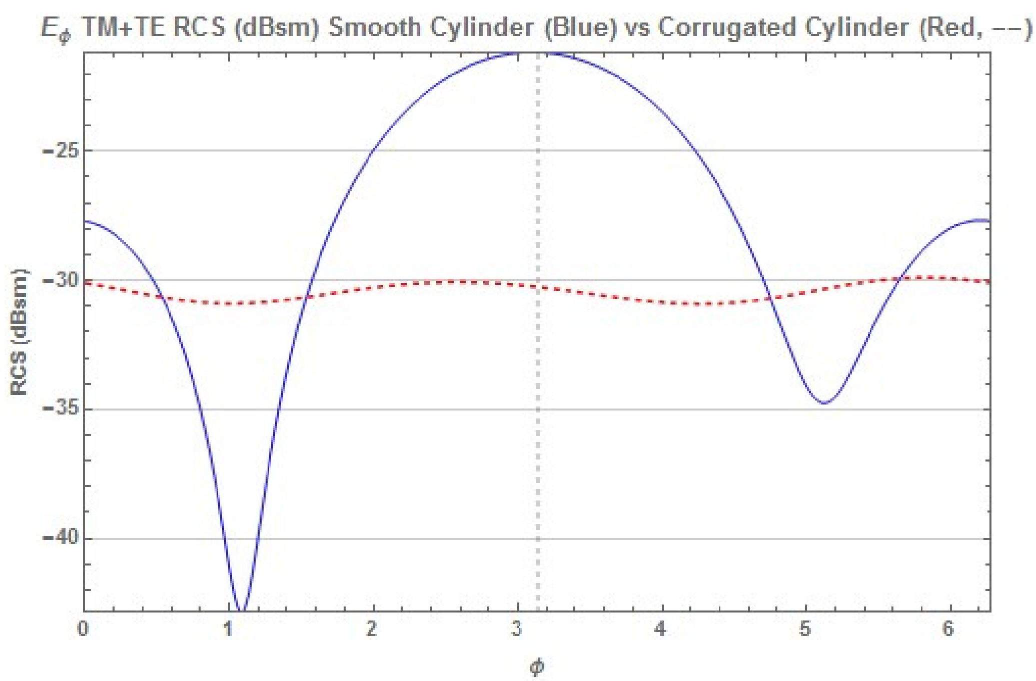


Figure 7-92 XY Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.0.1.0.0

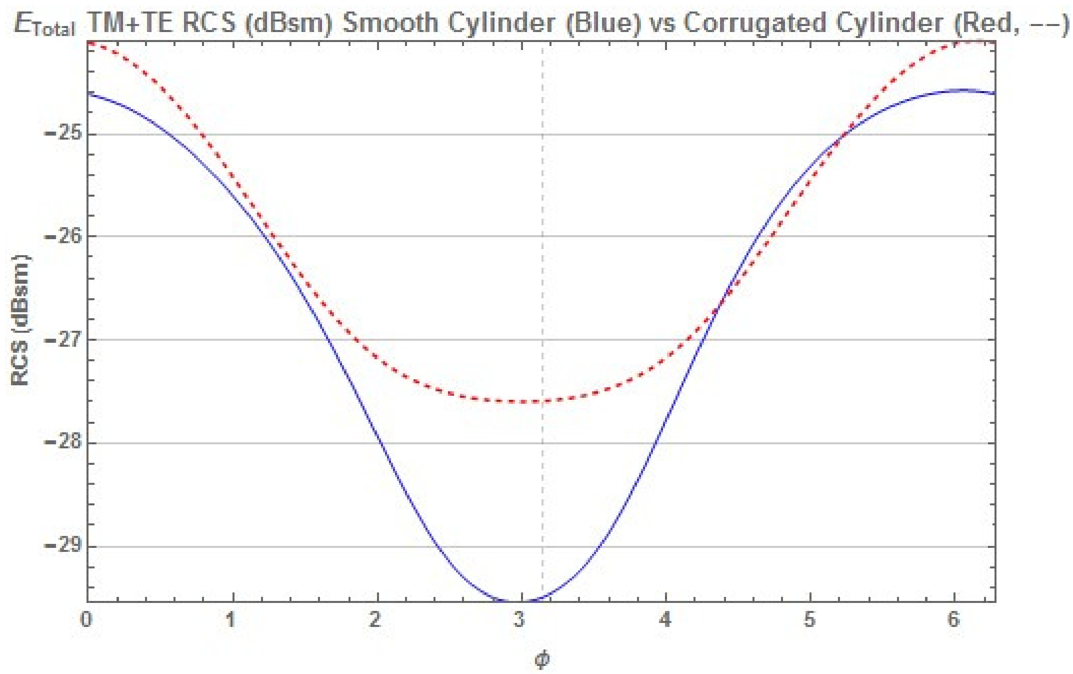


Figure 7-93 XY Plot form of RCS dBsm for E_{Total} of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.0.1.0.0

Table 15 Detailed parameters summary for changing ρ plots of Run b.0.1.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	0.019986 m	-	-	-	-
Frequency	1.50105×10^{10} Hz	-	-	-	-
a	0.0001 λ	-	-	-	-
b	0.1 λ	-	-	-	-
ρ1	0.0999 λ	-	-	-	-
ρ2	0.1 λ	-	-	-	-
ρ range	-	0.08991 λ	10.1 λ	0.0182333 λ	550
ϕ (observed)	37. Deg	-	-	-	-
z (observed)	500.5 a && 0.05005 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
ϕi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.0002	-	-	-	-
max allowable l	0.042685	-	-	-	-
Boundary	Boundary b	-	-	-	-

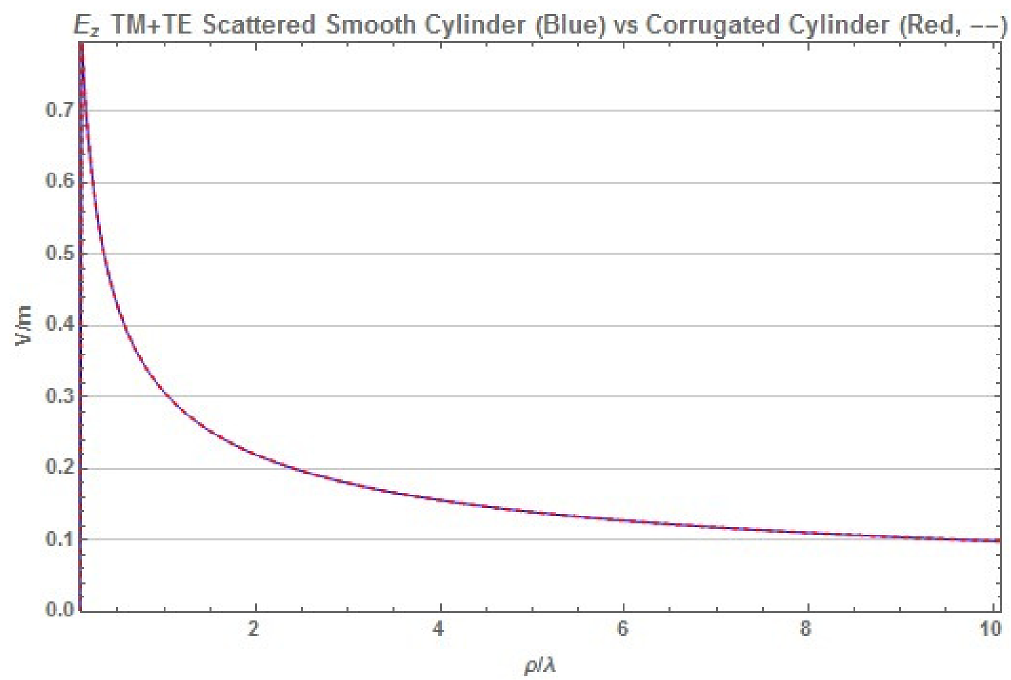


Figure 7-94 XY Plot of Scattered Field Amplitude Only, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.0.10.0.0

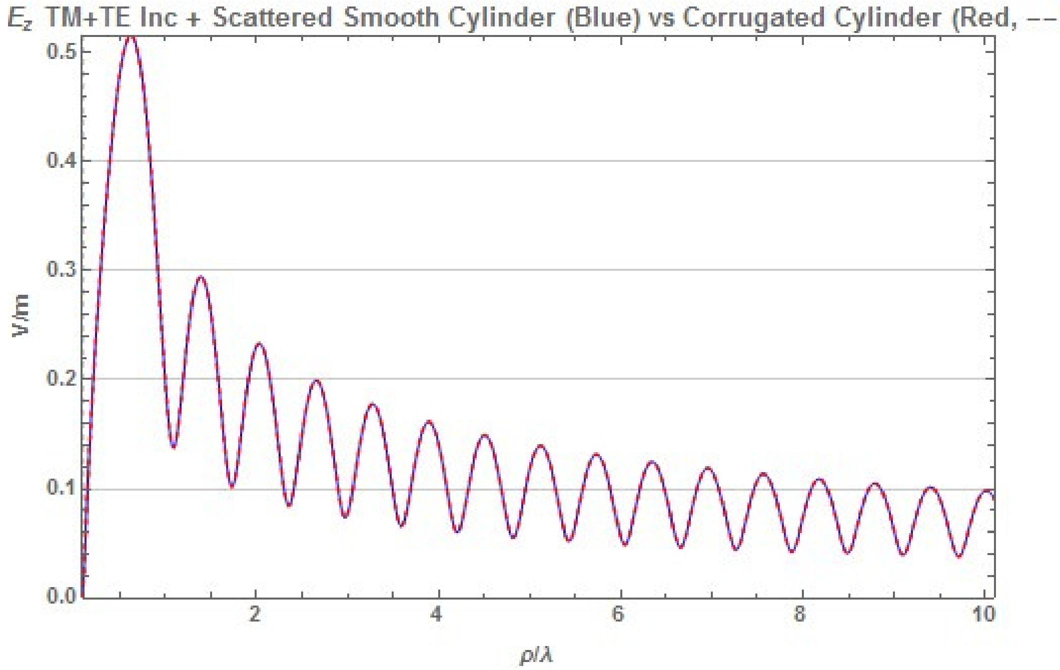


Figure 7-95 XY Plot of Scattered + Incident Field Amplitude Only, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.0.10.0.0

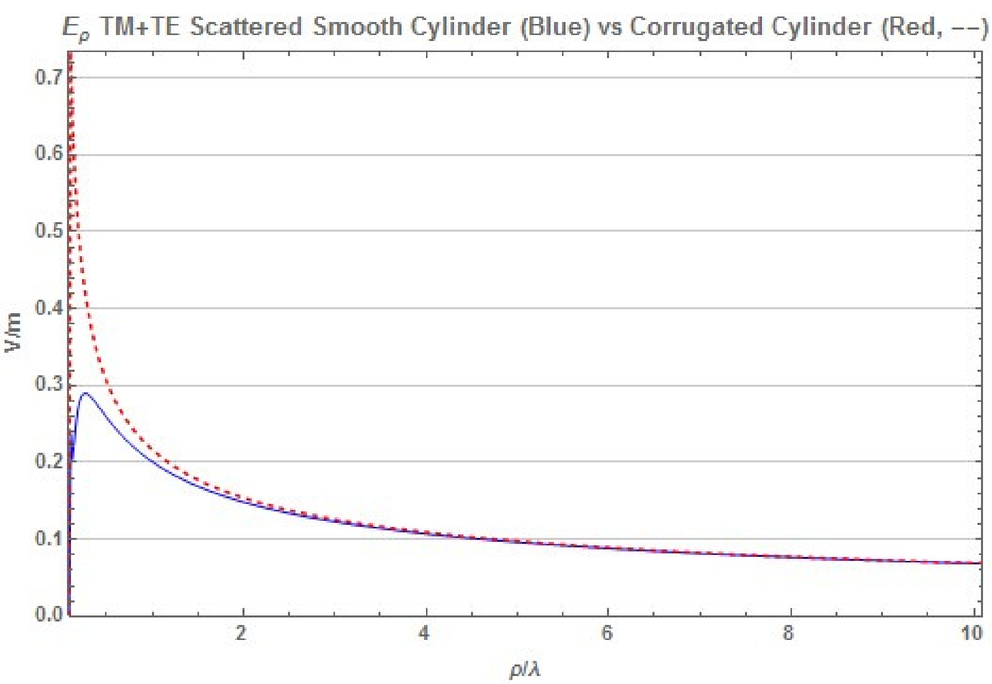


Figure 7-96 XY Plot of Scattered Field Amplitude Only, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.0.10.0.0

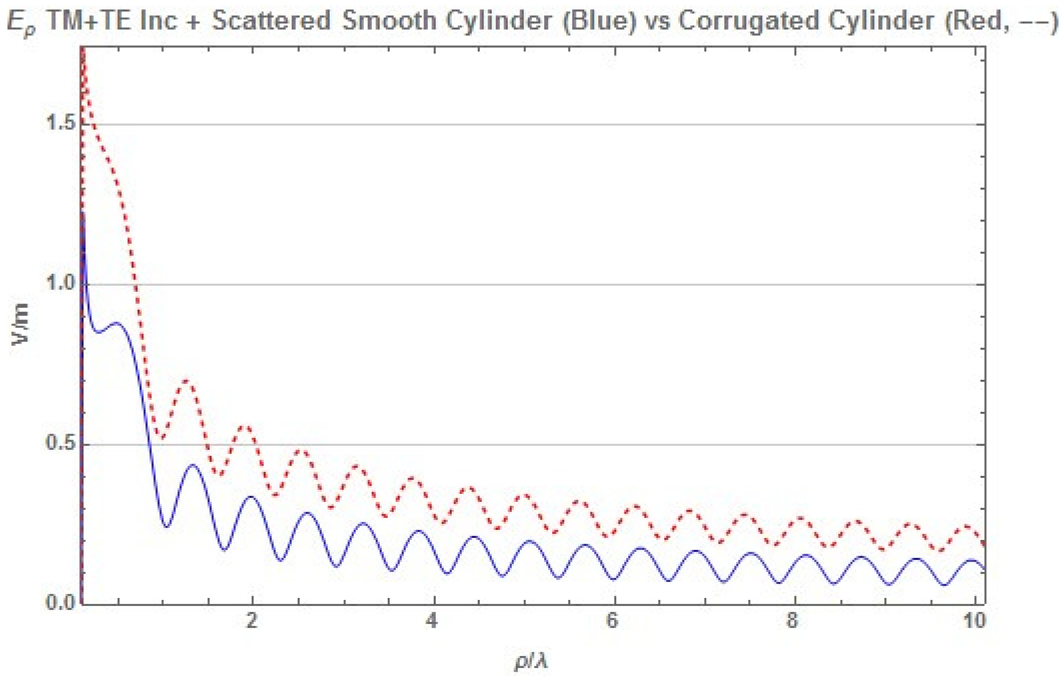


Figure 7-97 XY Plot of Scattered + Incident Field Amplitude Only, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.0.10.0.0

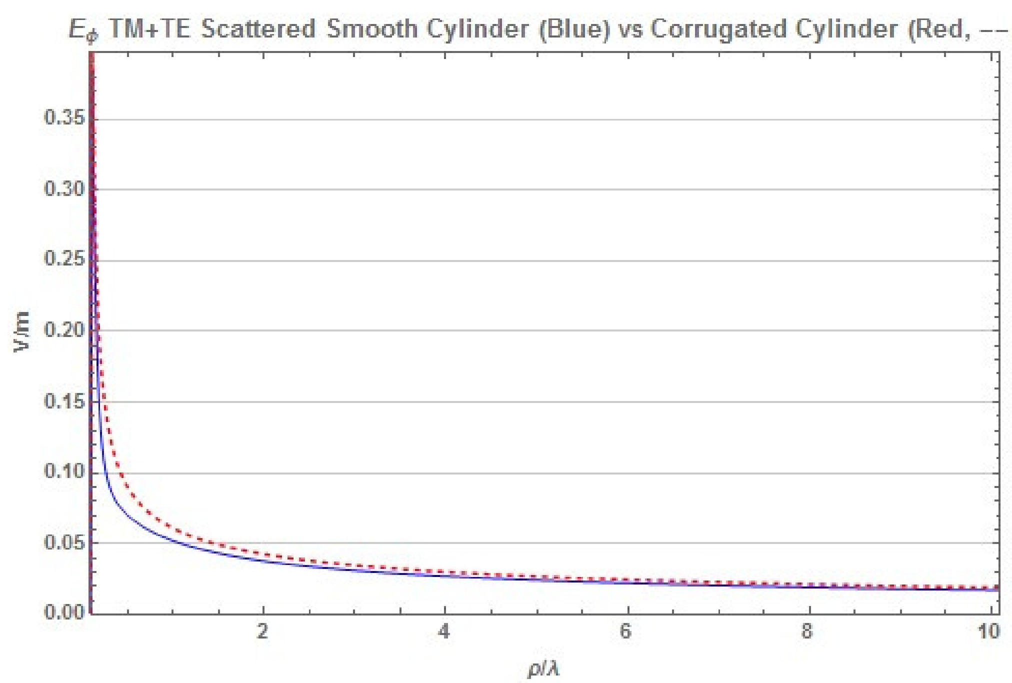


Figure 7-98 XY Plot of Scattered Field Amplitude Only, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.0.10.0.0

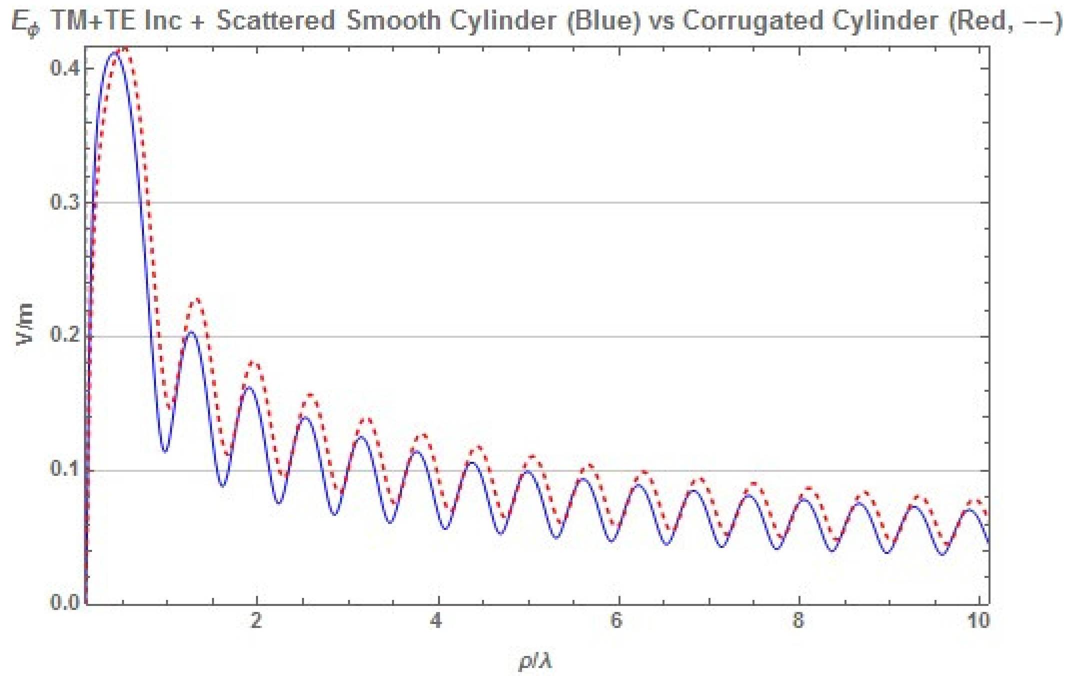


Figure 7-99 XY Plot of Scattered + Incident Field Amplitude Only, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.0.10.0.0

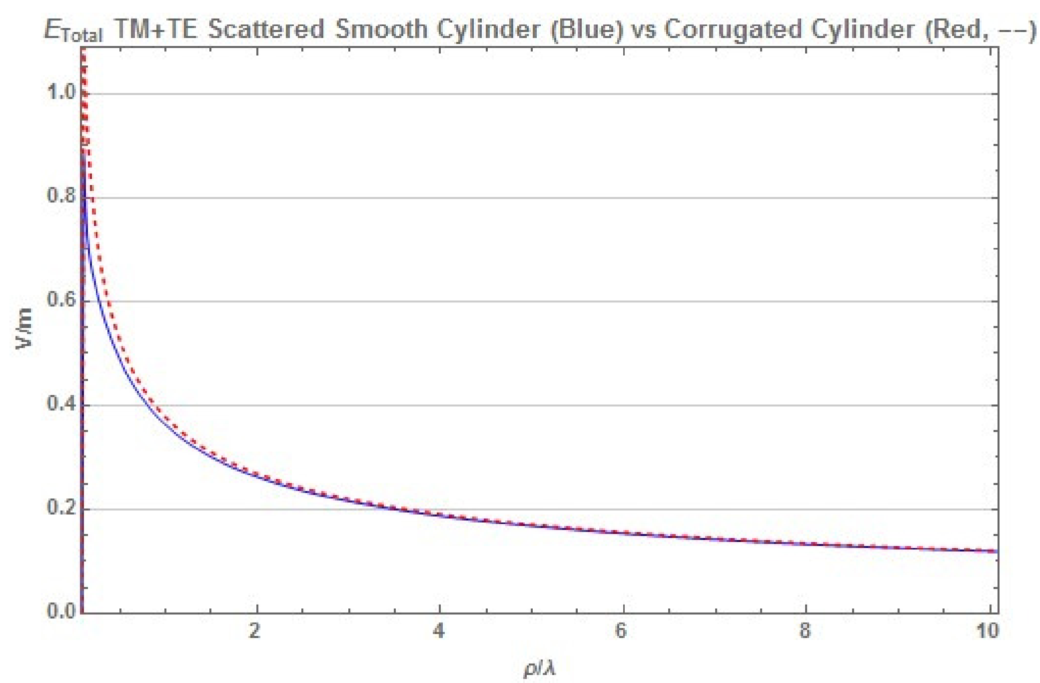


Figure 7-100 XY Plot of Scattered Field Amplitude Only, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.0.10.0.0

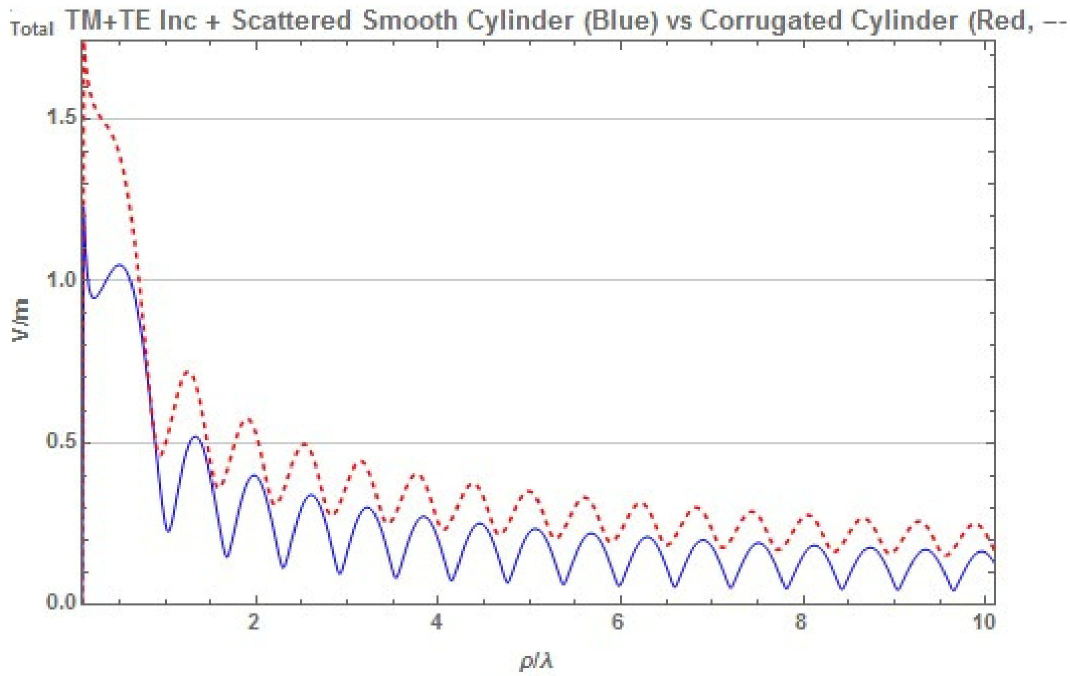


Figure 7-101 XY Plot of Scattered + Incident Field Amplitude Only, for ETotal, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run b.0.10.0.0

7.5.7 Run a_plus_b.20.0.0 (b=20λ, a=b*.001, ρ2=20λ, m=0)

Table 16 Detailed parameters summary for changing φ plots of Run a_plus_b.20.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	1.50105×10^{10} Hz	-	-	-	-
Frequency	0.019986 m	-	-	-	-
a	0.02 λ	-	-	-	-
b	20. λ	-	-	-	-
ρ1	19.98 λ	-	-	-	-
ρ2	20. λ	-	-	-	-
ϕ range	-	0.327273 Deg	359.673 Deg	0.654545 Deg	550
ρ (observed)	200. λ	-	-	-	-
z (observed)	0.25 a && 0.005 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
ϕi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.04	-	-	-	-
max allowable l	8.537	-	-	-	-
Boundary	Boundary a+b	-	-	-	-

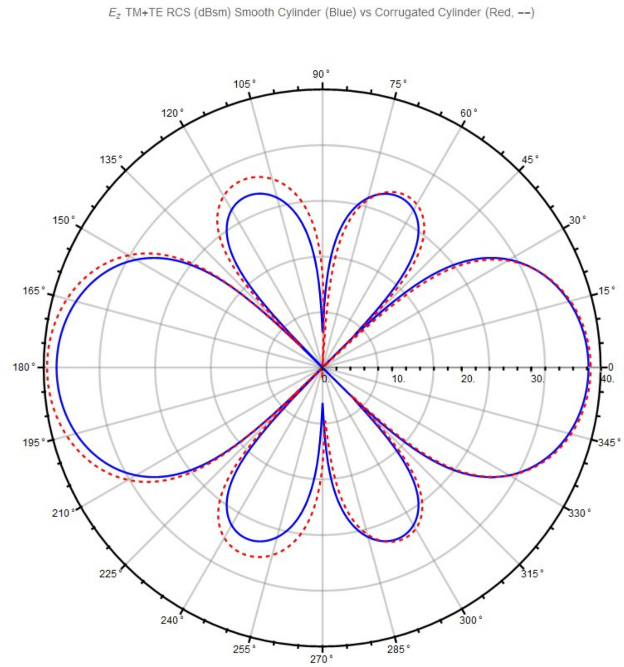


Figure 7-102 Polar Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

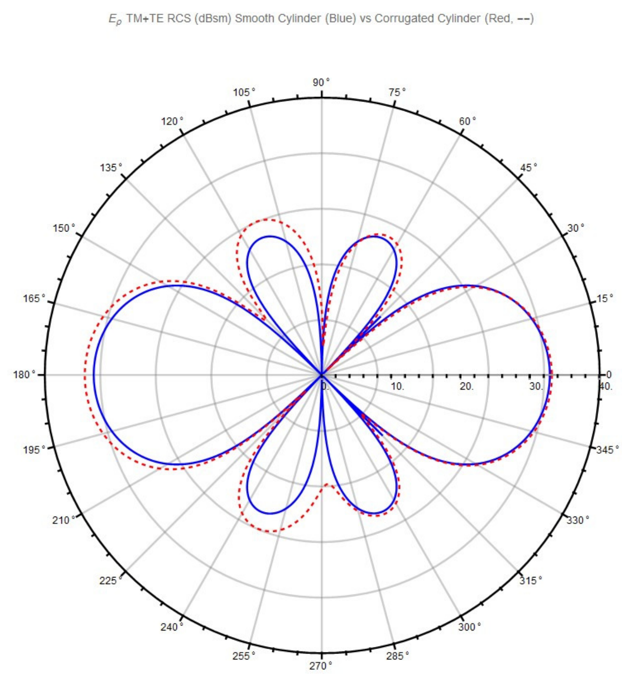


Figure 7-103 Polar Plot form of RCS dBsm for Eρ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

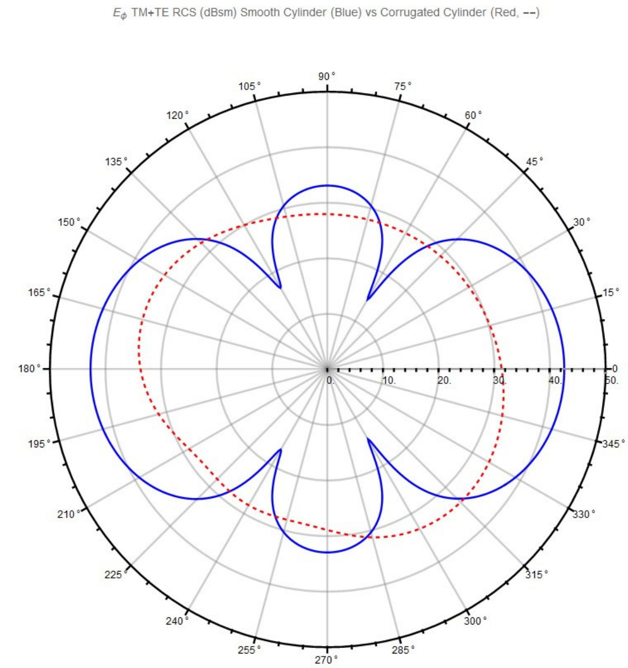


Figure 7-104 Polar Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

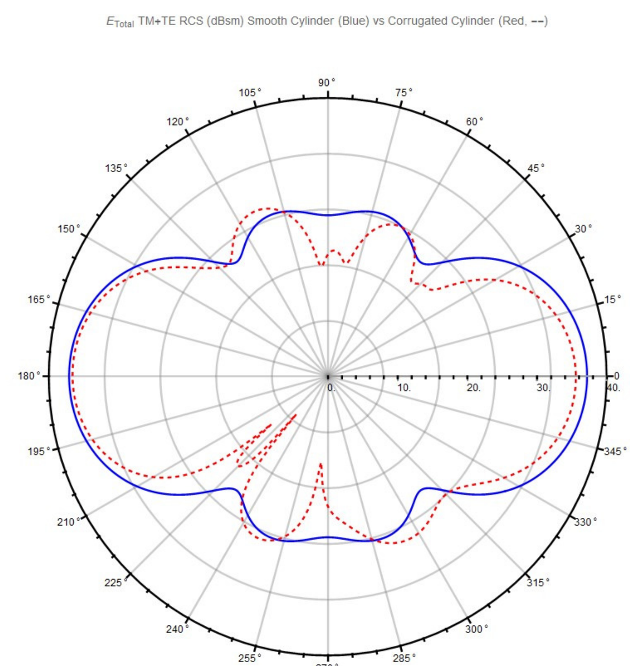


Figure 7-105 Polar Plot form of RCS dBsm for E_{Total} of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

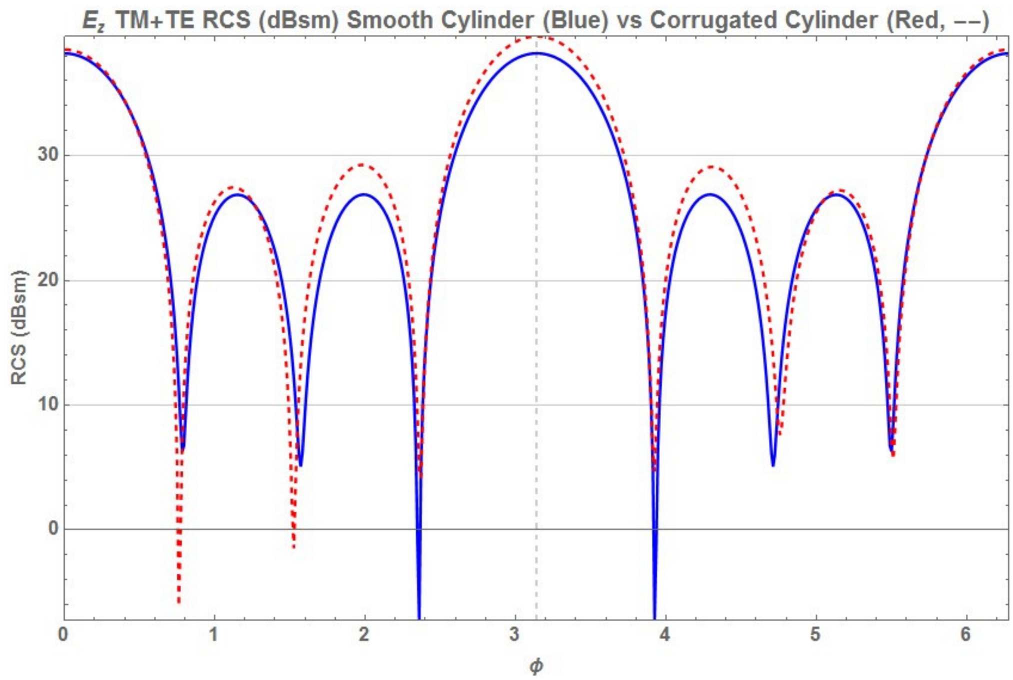


Figure 7-106 XY Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

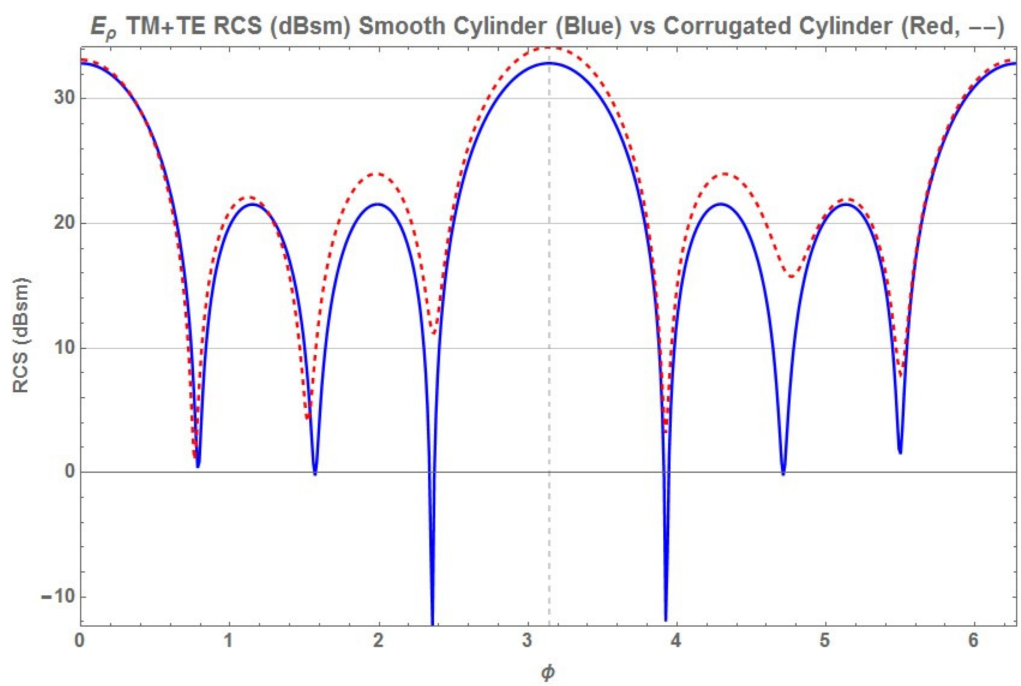


Figure 7-107 XY Plot form of RCS dBsm for E_ρ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

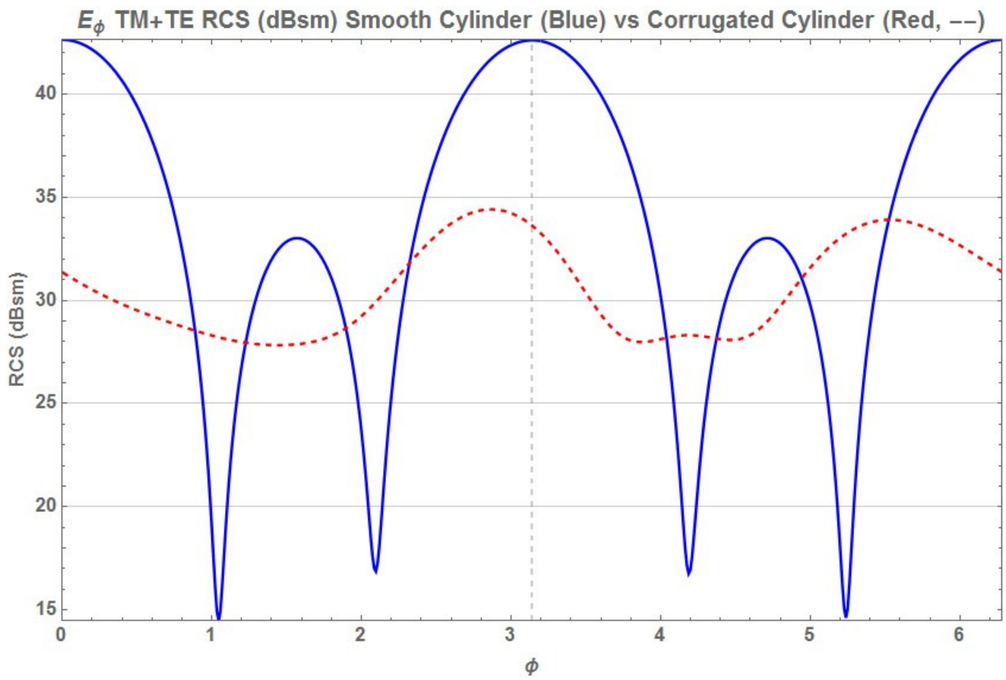


Figure 7-108 XY Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

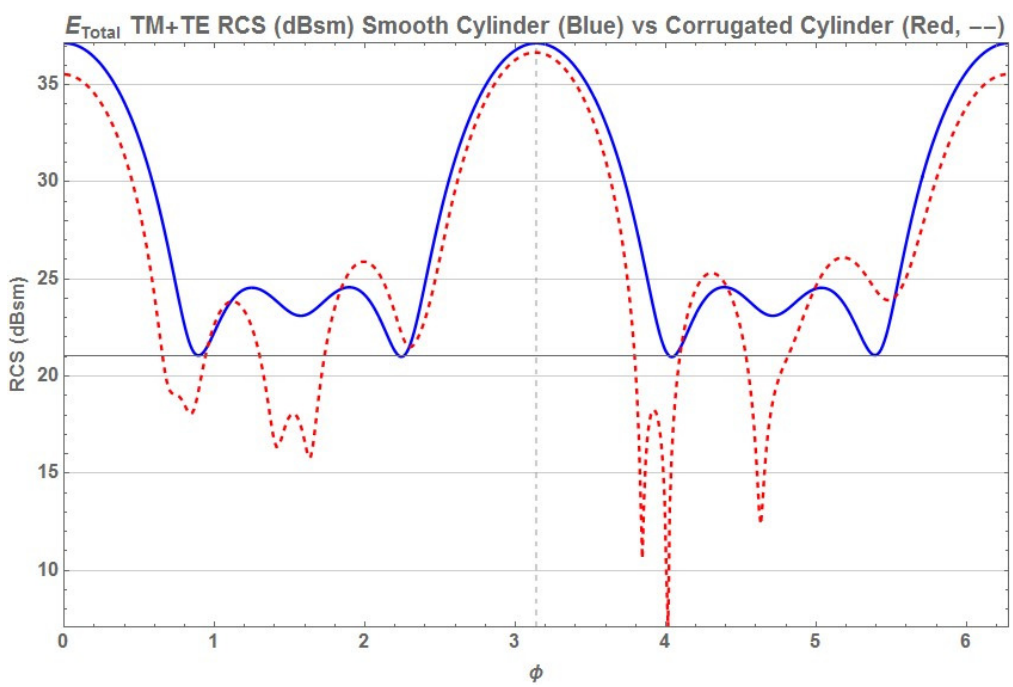


Figure 7-109 XY Plot form of RCS dBsm for E_{Total} of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

Table 17 Detailed parameters summary for changing ρ plots of Run a_plus_b.20.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	0.019986 m	-	-	-	-
Frequency	1.50105×10^{10} Hz	-	-	-	-
a	0.02 λ	-	-	-	-
b	20. λ	-	-	-	-
ρ1	19.98 λ	-	-	-	-
ρ2	20. λ	-	-	-	-
ρ range	-	17.982 λ	30. λ	0.0218907 λ	550
ϕ (observed)	37. Deg	-	-	-	-
z (observed)	0.25 a && 0.005 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
ϕi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.04	-	-	-	-
max allowable l	8.537	-	-	-	-
Boundary	Boundary a+b	-	-	-	-

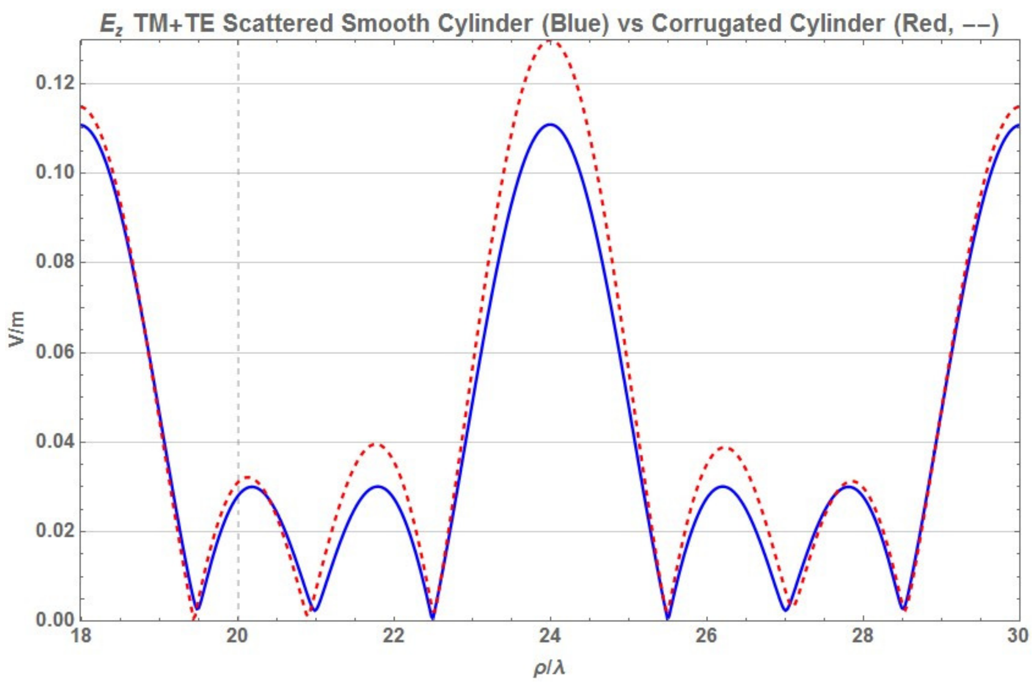


Figure 7-110 XY Plot of Scattered Field Amplitude Only, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

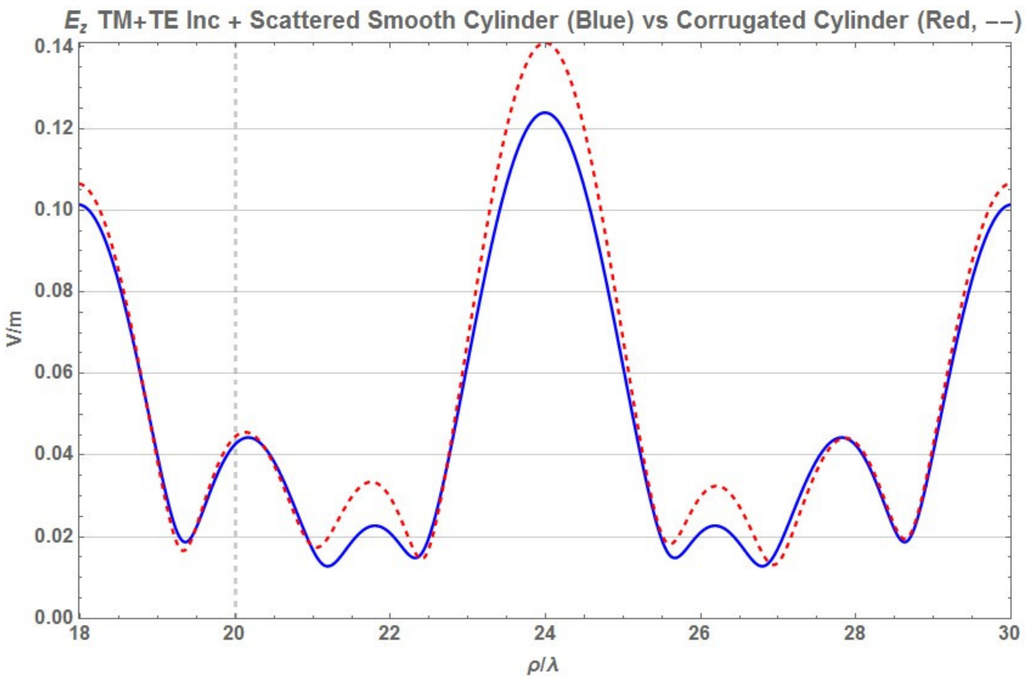


Figure 7-111 XY Plot of Scattered + Incident Field Amplitude, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

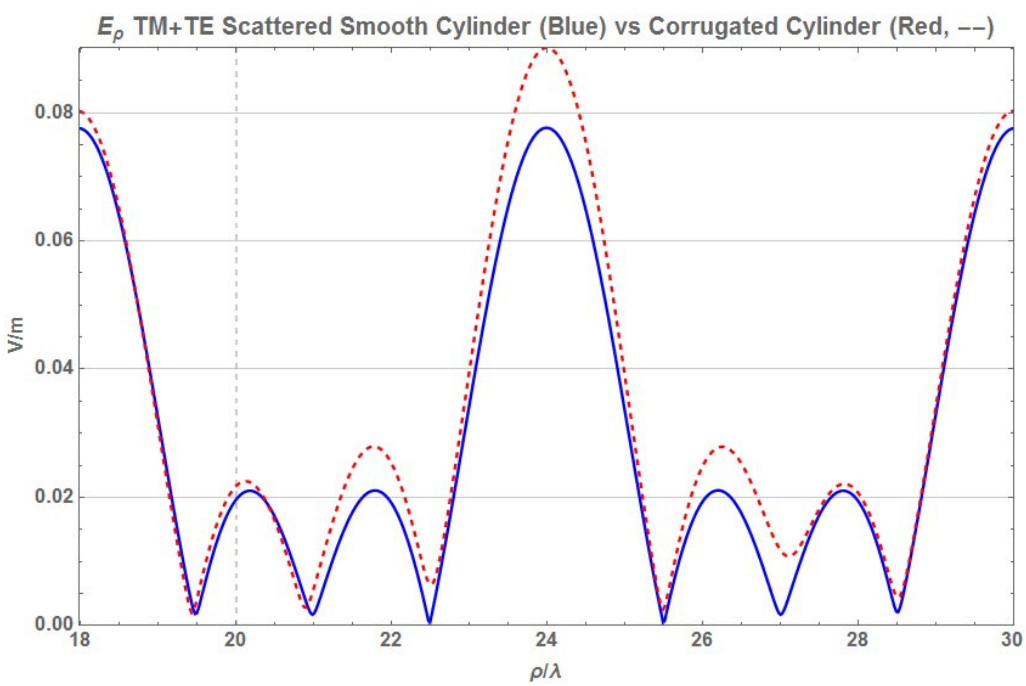


Figure 7-112 XY Plot of Scattered Field Amplitude Only, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

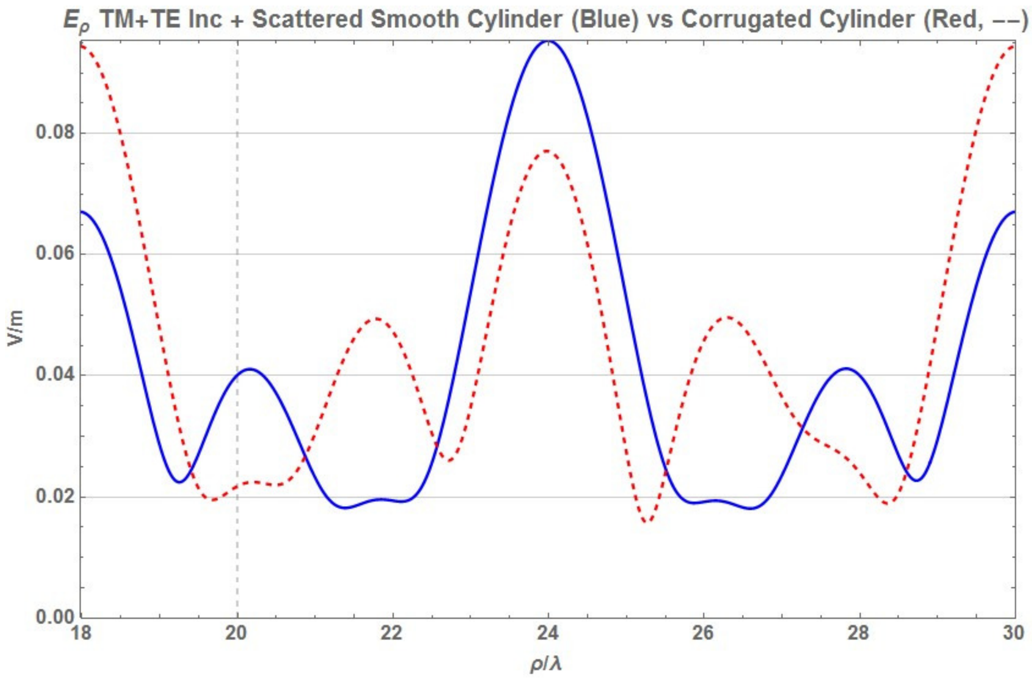


Figure 7-113 XY Plot of Scattered + Incident Field Amplitude, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

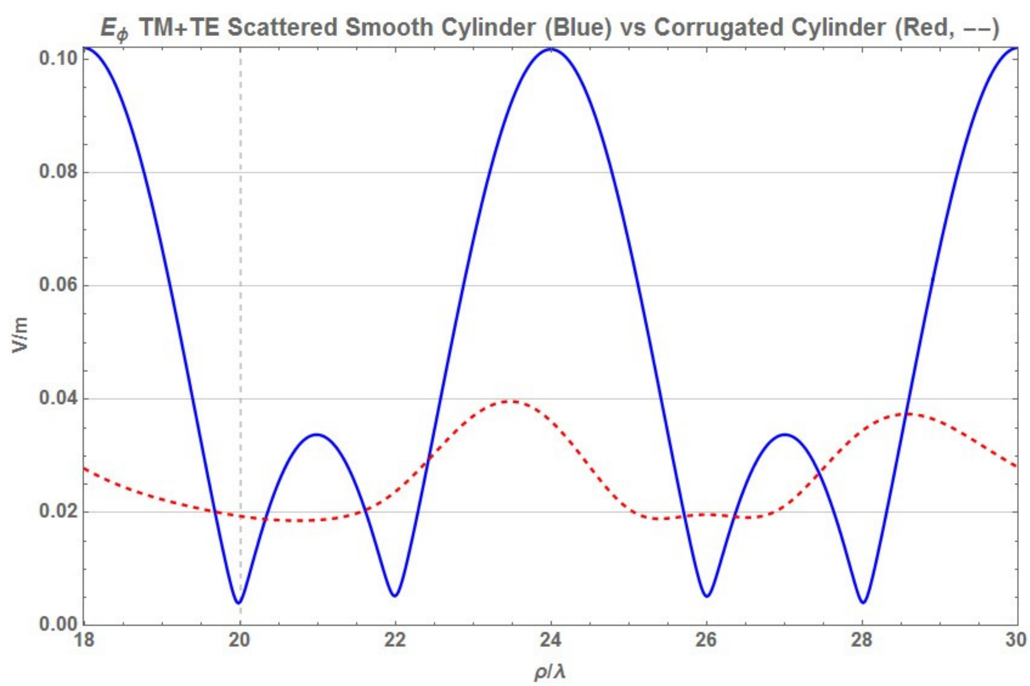


Figure 7-114 XY Plot of Scattered Field Amplitude Only, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

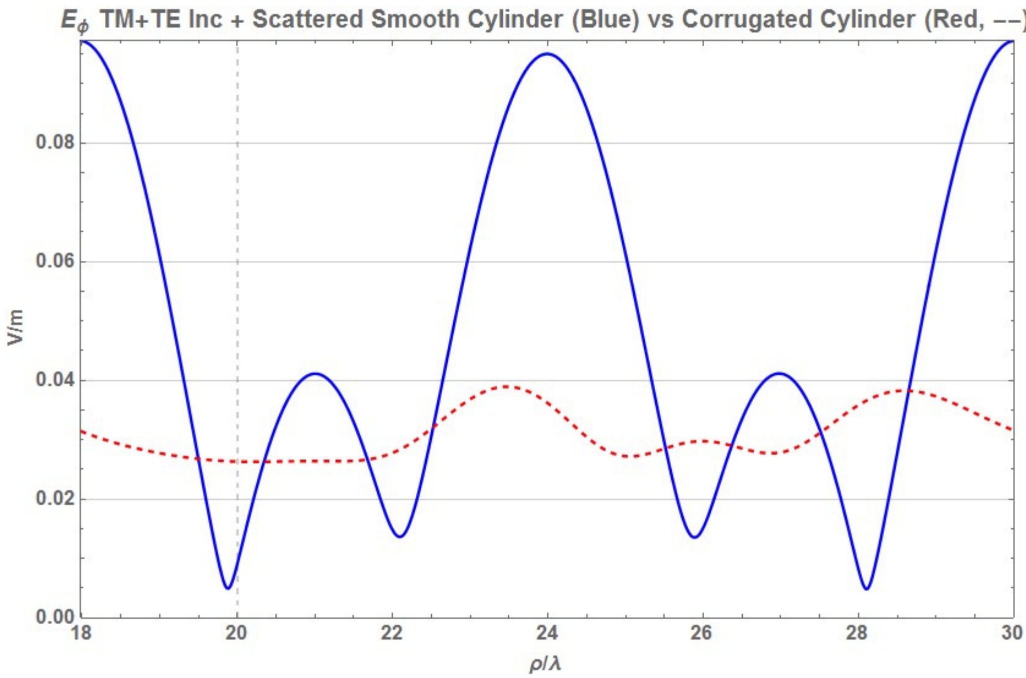


Figure 7-115 XY Plot of Scattered + Incident Field Amplitude, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

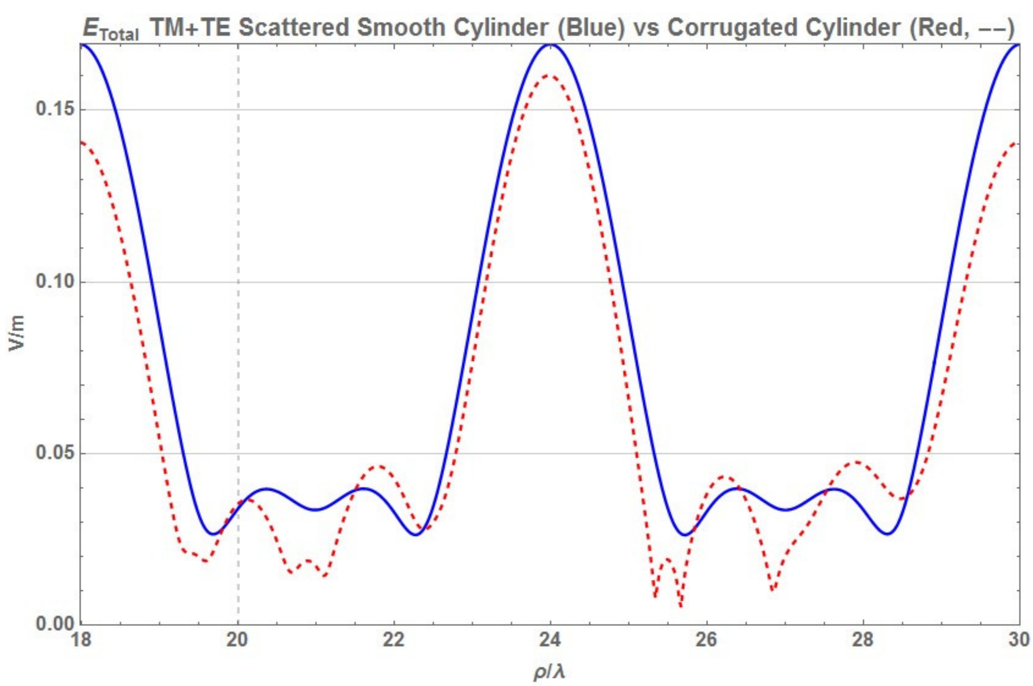


Figure 7-116 XY Plot of Scattered Field Amplitude Only, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

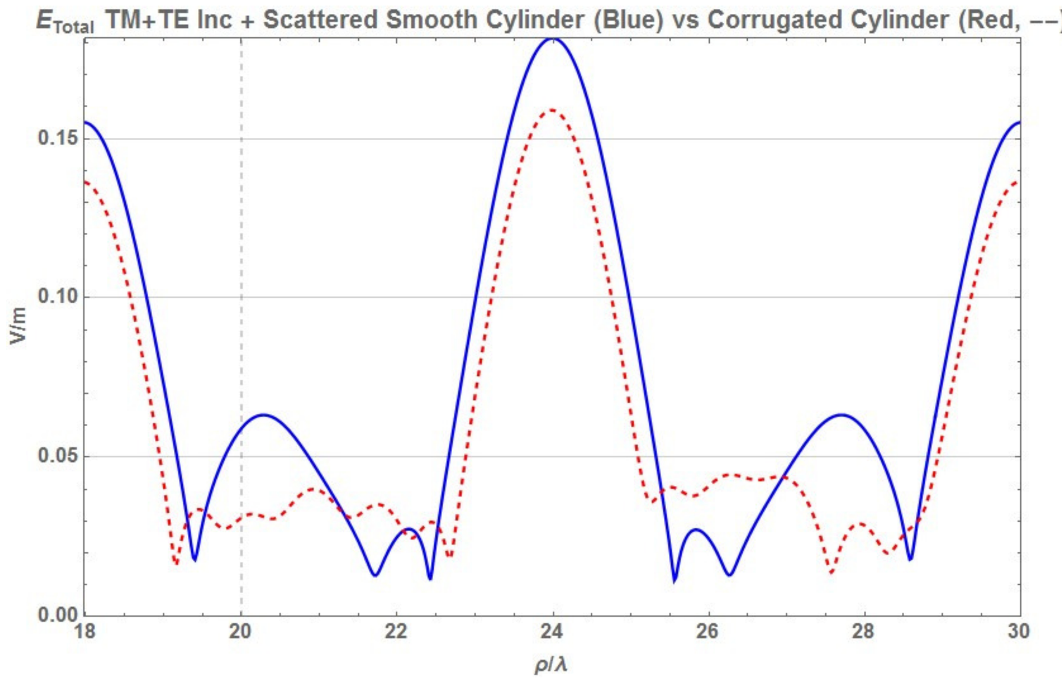


Figure 7-117 XY Plot of Scattered + Incident Field Amplitude, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.20.0.0

7.5.8 Run a_plus_b.2.0.0 (b=2λ, a=b*.001, ρ2=2λ, m=0)

Table 18 Detailed parameters summary for changing φ plots of Run a_plus_b.2.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	1.50105×10^{10} Hz	-	-	-	-
Frequency	0.019986 m	-	-	-	-
a	0.002 λ	-	-	-	-
b	2. λ	-	-	-	-
ρ1	1.998 λ	-	-	-	-
ρ2	2. λ	-	-	-	-
ϕ range	-	0.327273 Deg	359.673 Deg	0.654545 Deg	550
ρ (observed)	20. λ	-	-	-	-
z (observed)	0.25 a && 0.0005 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
ϕi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.004	-	-	-	-
max allowable l	0.8537	-	-	-	-
Boundary	Boundary a+b	-	-	-	-

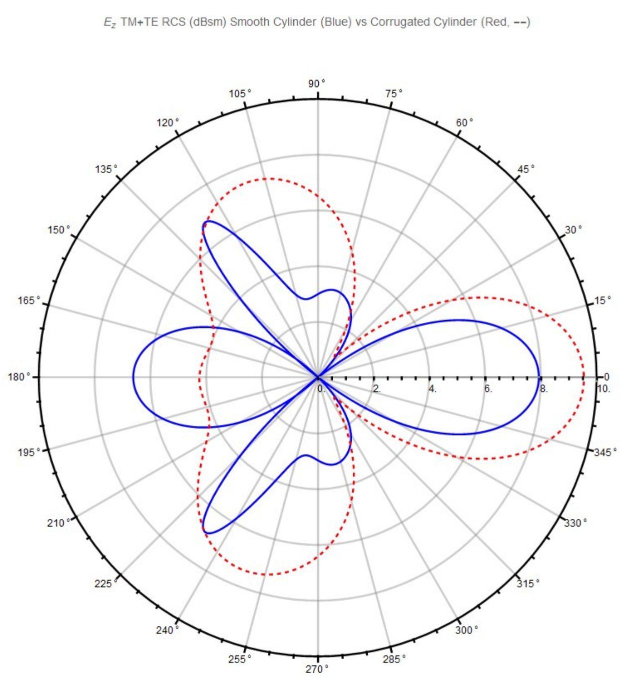


Figure 7-118 Polar Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0

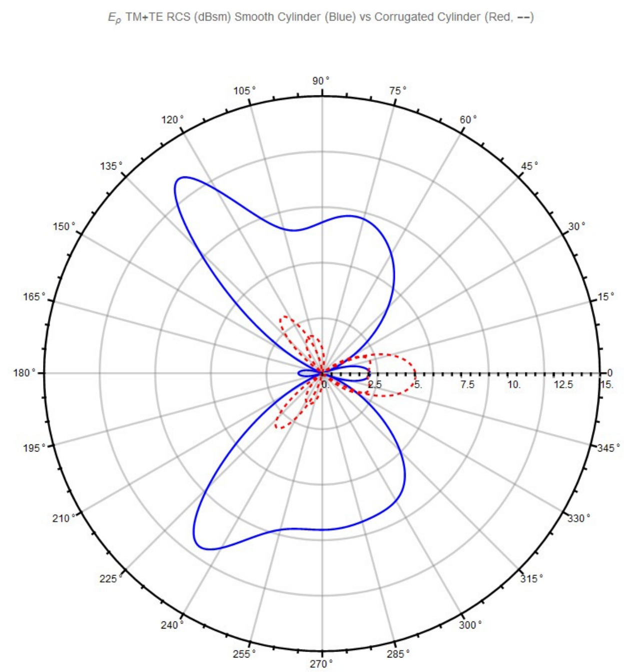


Figure 7-119 Polar Plot form of RCS dBsm for Eρ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0

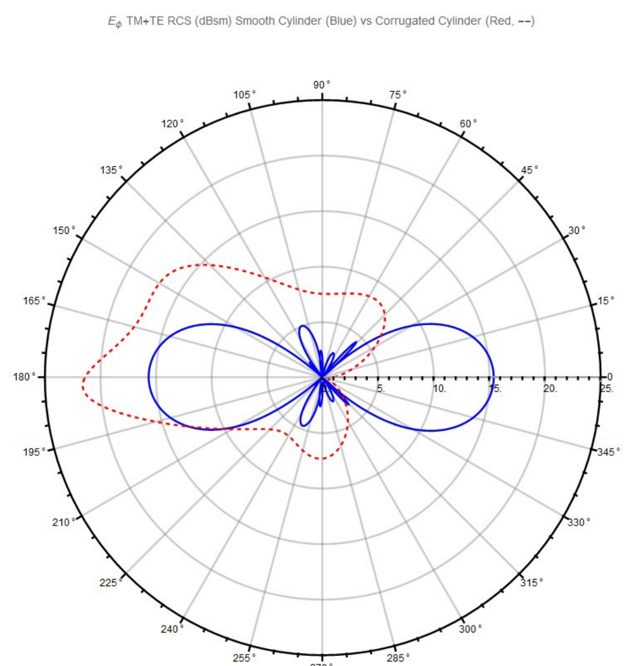


Figure 7-120 Polar Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0

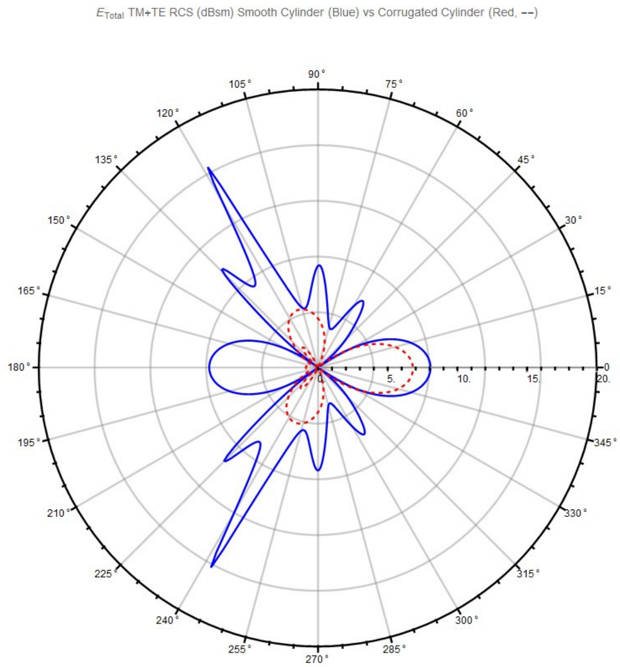


Figure 7-121 Polar Plot form of RCS dBsm for ETotal of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0

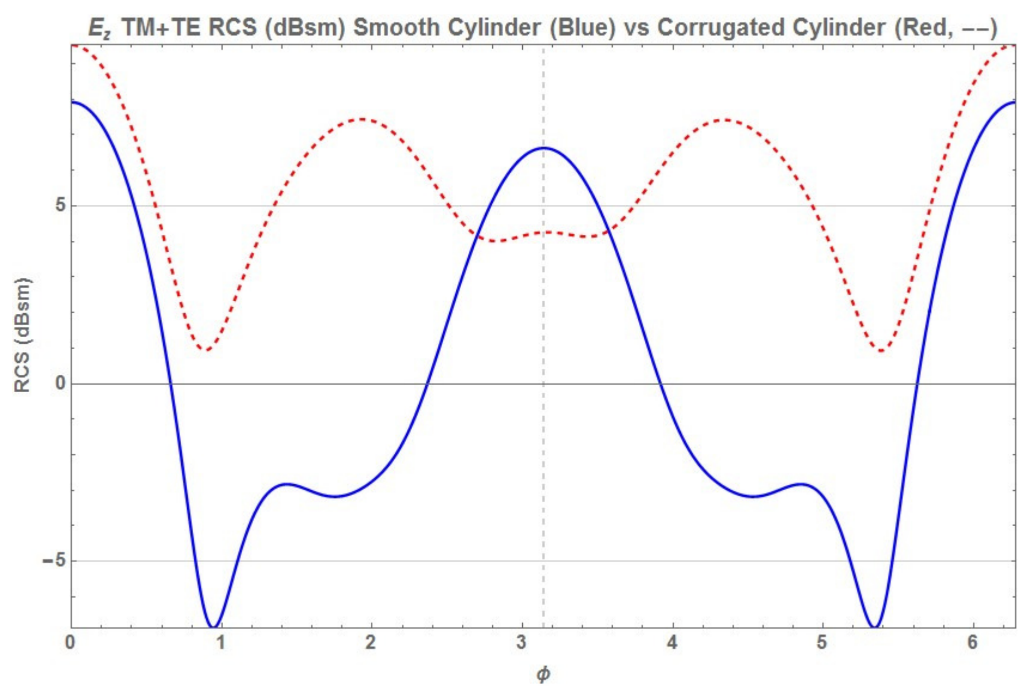


Figure 7-122 XY Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0

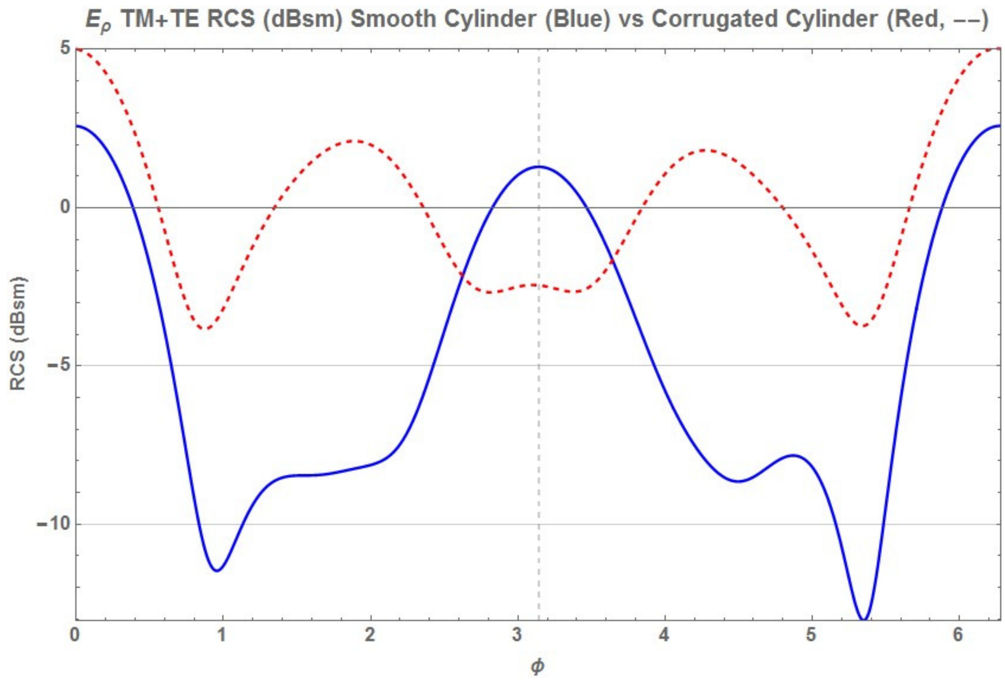


Figure 7-123 XY Plot form of RCS dBsm for Eρ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0

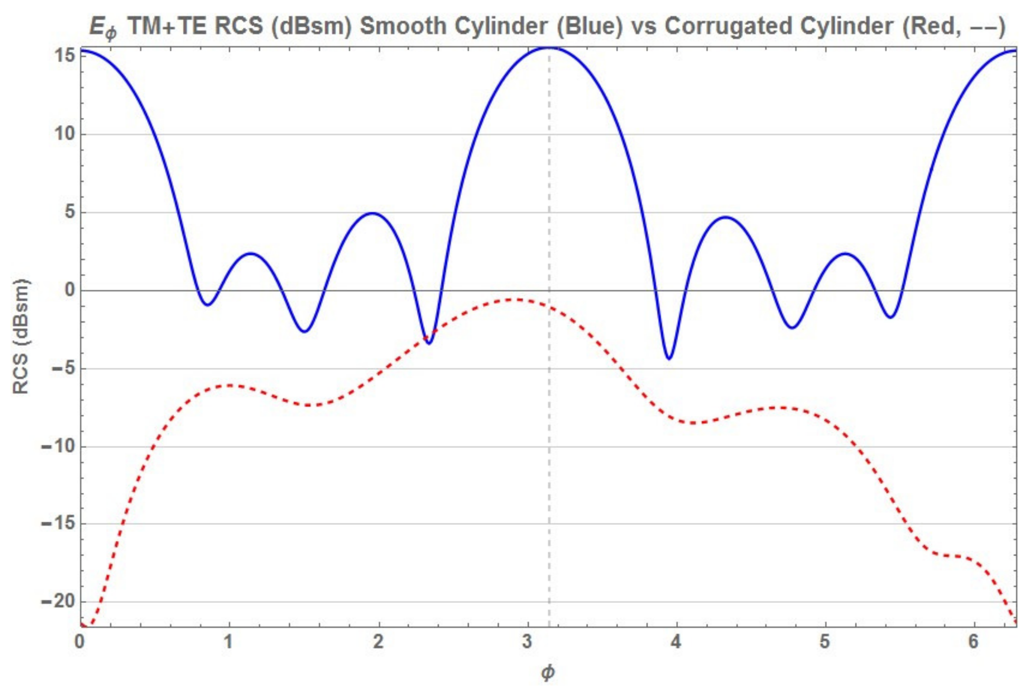
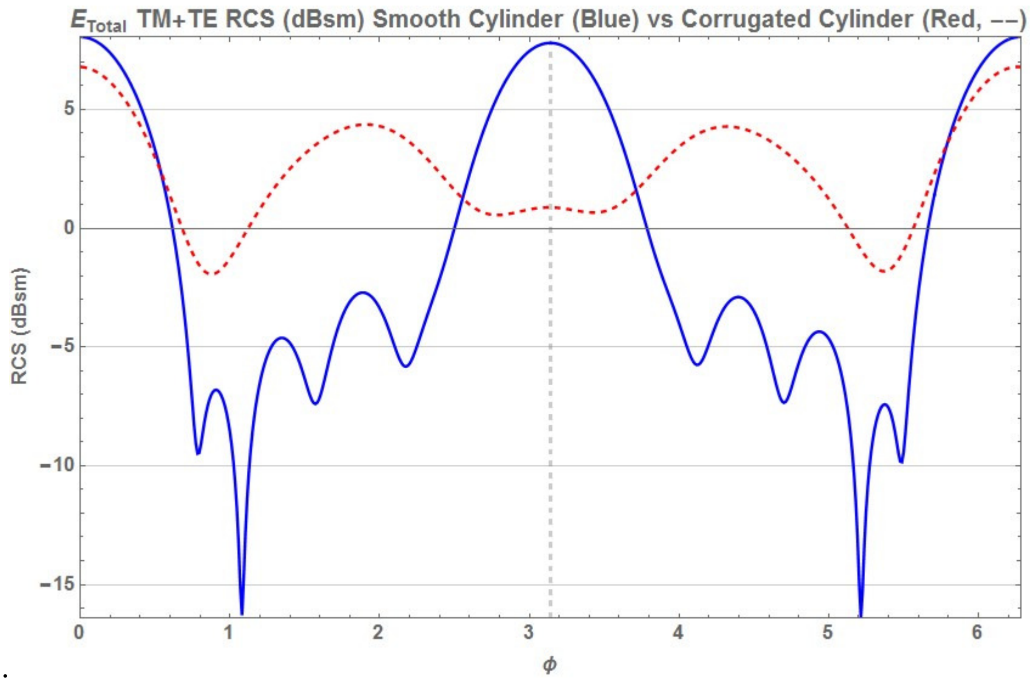


Figure 7-124 XY Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0



.

Figure 7-125 XY Plot form of RCS dBsm for ETotal of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0

Table 19 Detailed parameters summary for changing ρ plots of Run a_plus_b.2.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	0.019986 m	-	-	-	-
Frequency	1.50105×10^{10} Hz	-	-	-	-
a	0.002 λ	-	-	-	-
b	2. λ	-	-	-	-
ρ1	1.998 λ	-	-	-	-
ρ2	2. λ	-	-	-	-
ρ range	-	1.7982 λ	12. λ	0.0185825 λ	550
ϕ (observed)	37. Deg	-	-	-	-
z (observed)	0.25 a && 0.0005 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
ϕi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.004	-	-	-	-
max allowable l	0.8537	-	-	-	-
Boundary	Boundary a+b	-	-	-	-

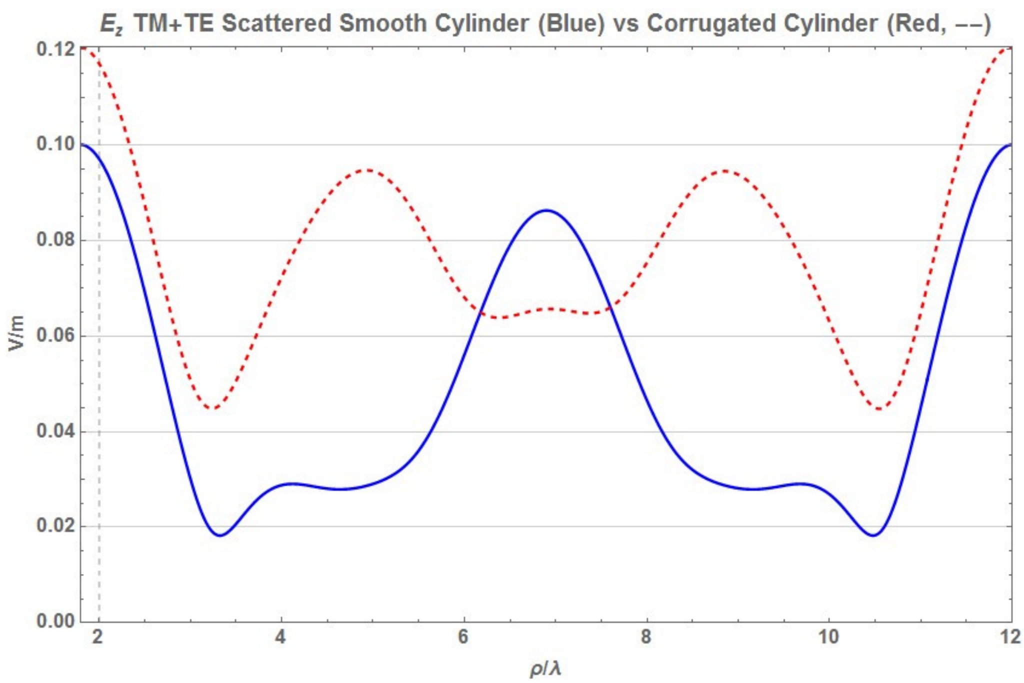


Figure 7-126 XY Plot of Scattered Field Amplitude Only, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0

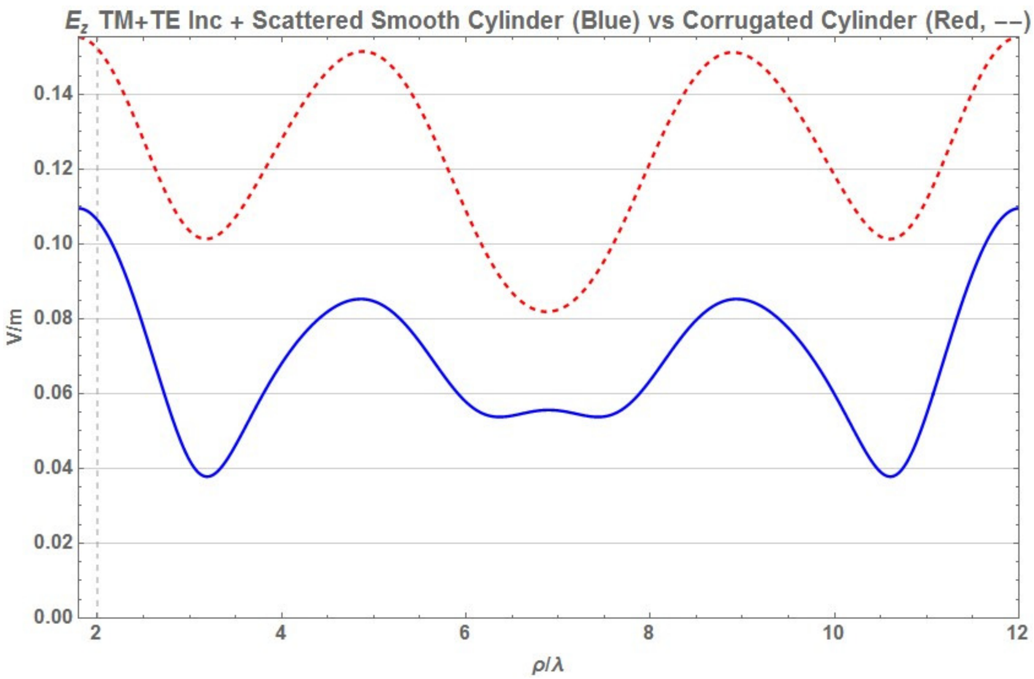


Figure 7-127 XY Plot of Scattered + Incident Field Amplitude, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0

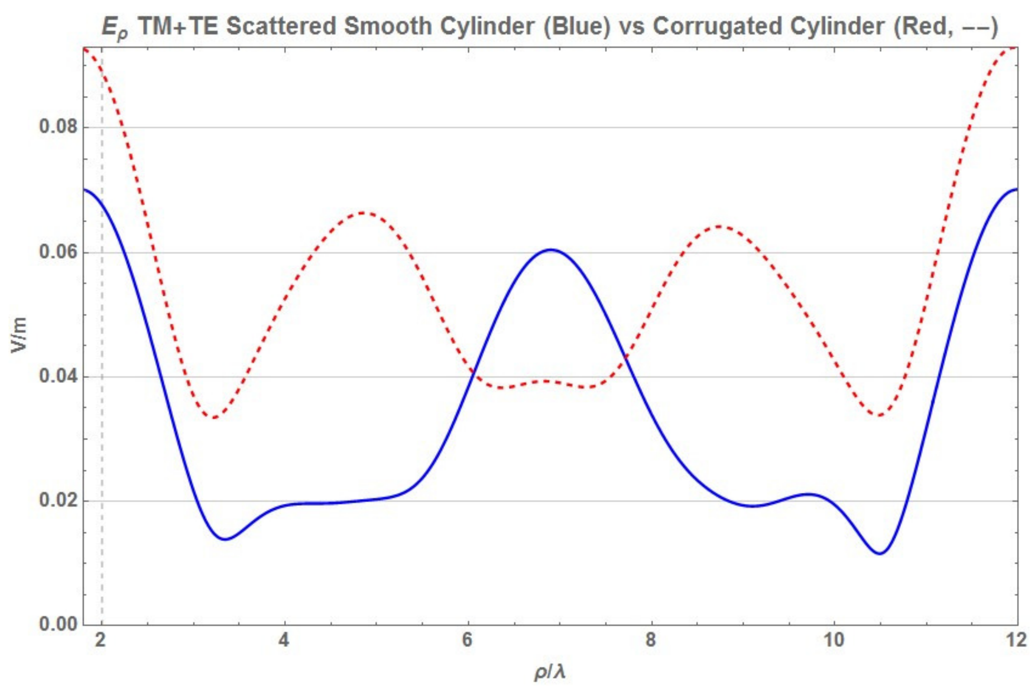


Figure 7-128 XY Plot of Scattered Field Amplitude Only, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0

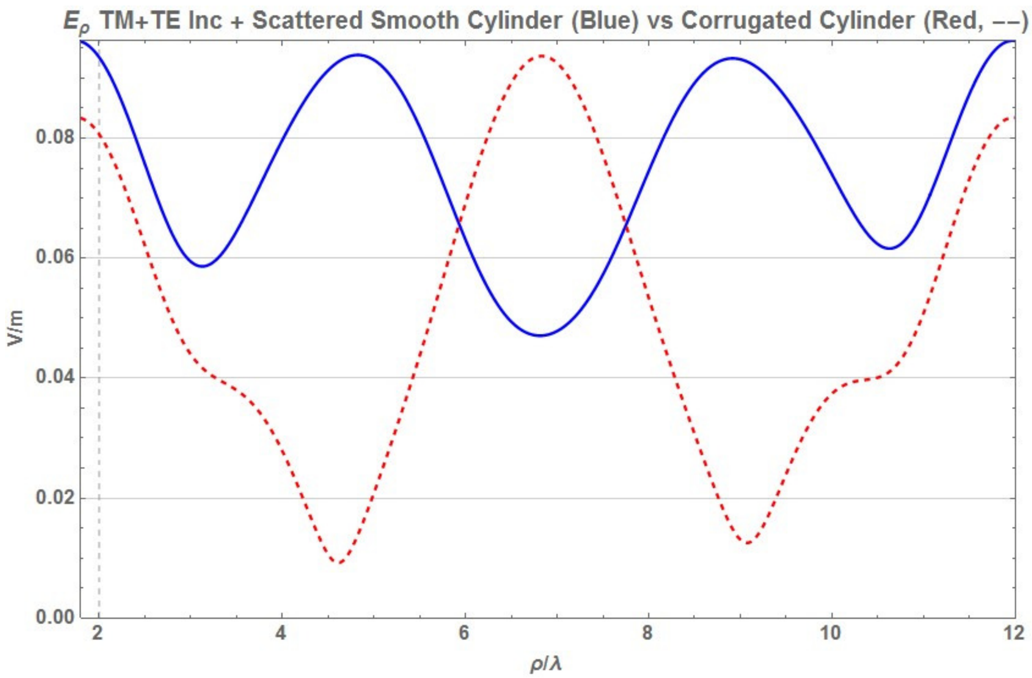


Figure 7-129 XY Plot of Scattered + Incident Field Amplitude, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0

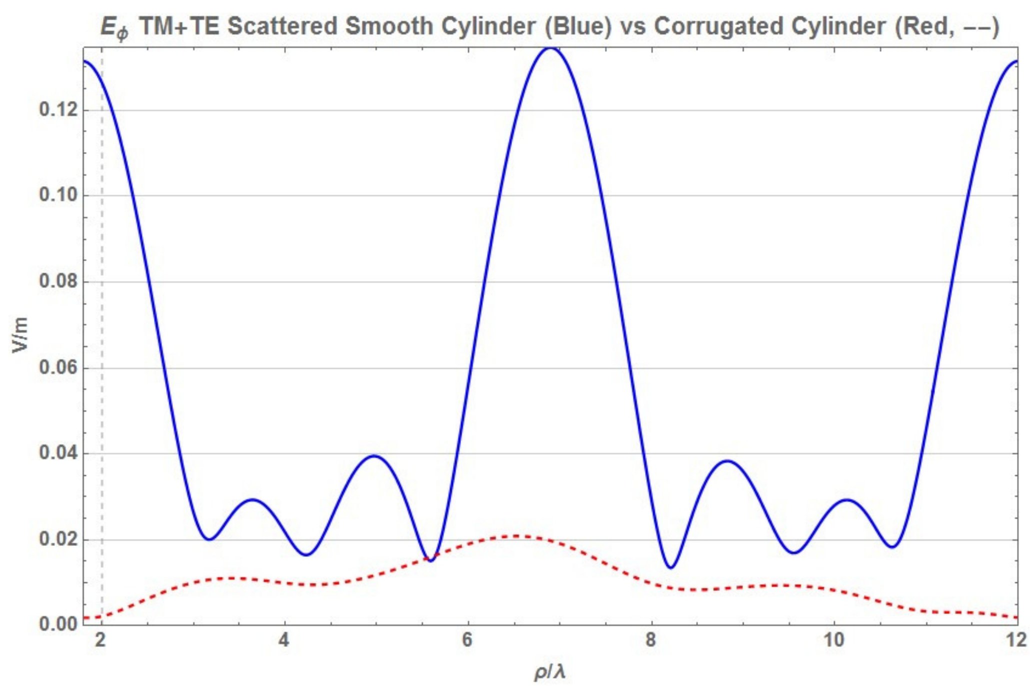


Figure 7-130 XY Plot of Scattered Field Amplitude Only, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0

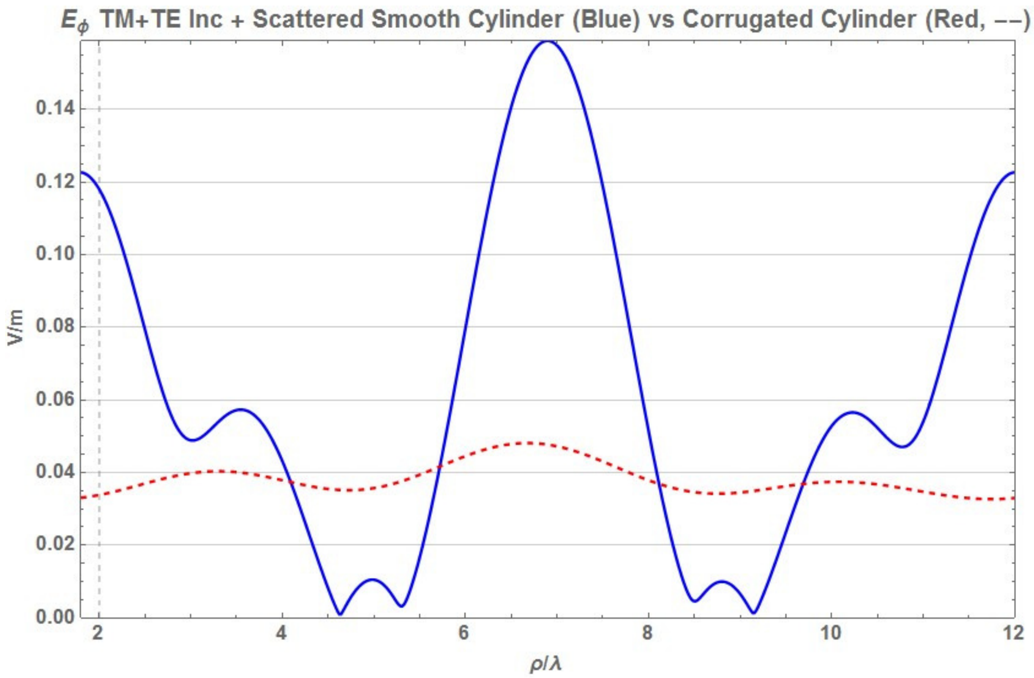


Figure 7-131 XY Plot of Scattered + Incident Field Amplitude, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0

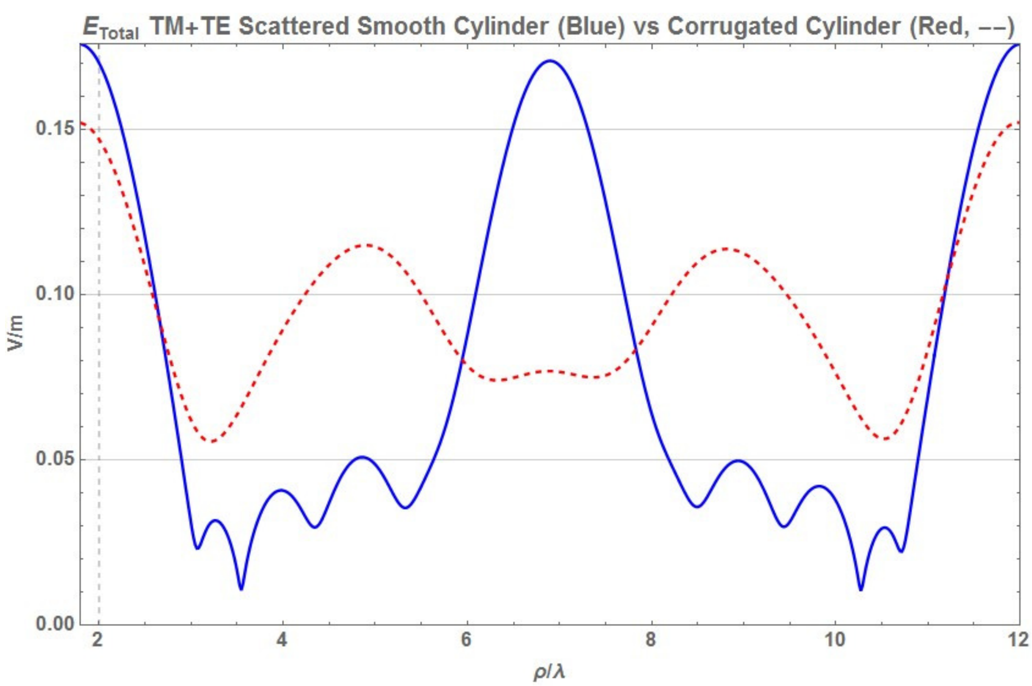


Figure 7-132 XY Plot of Scattered Field Amplitude Only, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0

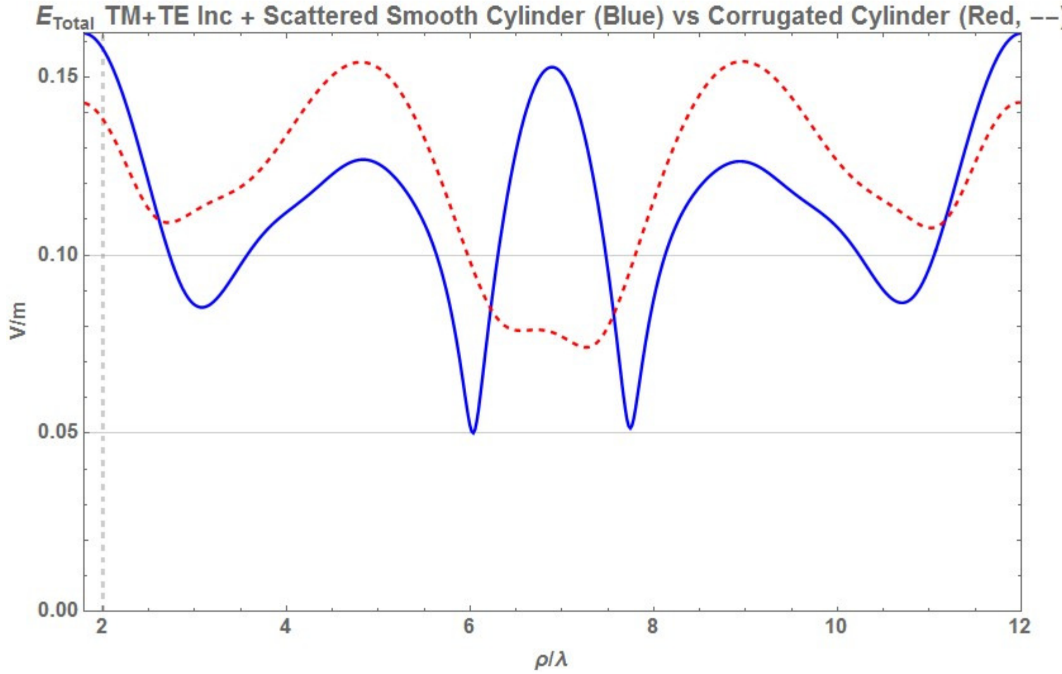


Figure 7-133 XY Plot of Scattered + Incident Field Amplitude, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.2.0.0

7.5.9 Run a_plus_b.0.1.0.0 (b=0.1λ, a=b*.001, ρ2=0.1λ, m=0)

Table 20 Detailed parameters summary for changing φ plots of Run a_plus_b.0.1.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	0.00265258 A/m	-	-	-	-
λ	1.50105×10^{10} Hz	-	-	-	-
Frequency	0.019986 m	-	-	-	-
a	0.0001 λ	-	-	-	-
b	0.1 λ	-	-	-	-
ρ1	0.0999 λ	-	-	-	-
ρ2	0.1 λ	-	-	-	-
ϕ range	-	0.327273 Deg	359.673 Deg	0.654545 Deg	550
ρ (observed)	1. λ	-	-	-	-
z (observed)	0.25 a && 0.000025 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	55. Deg	-	-	-	-
ϕi	37. Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.0002	-	-	-	-
max allowable l	0.042685	-	-	-	-
Boundary	Boundary a+b	-	-	-	-

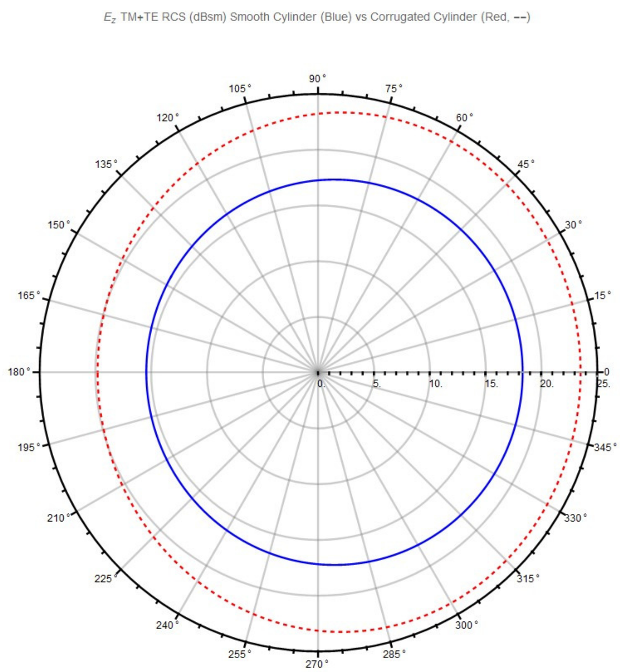


Figure 7-134 Polar Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a_plus_b.0.1.0.0

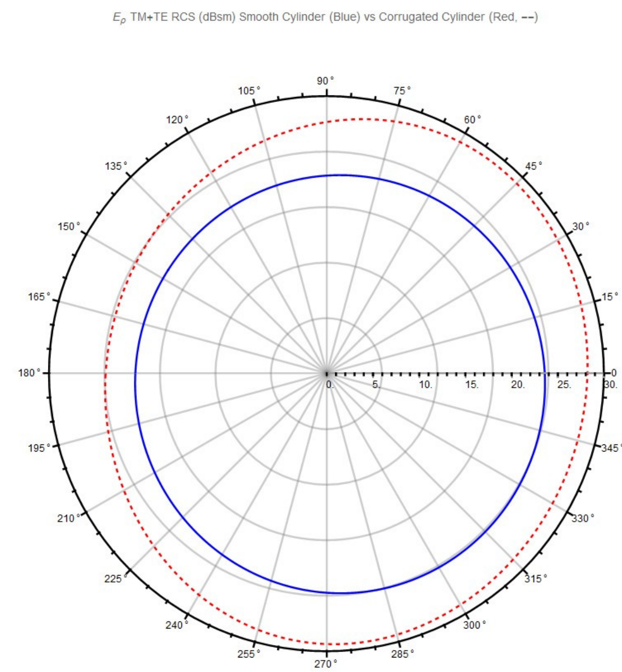


Figure 7-135 Polar Plot form of RCS dBsm for Eρ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a_plus_b.0.1.0.0

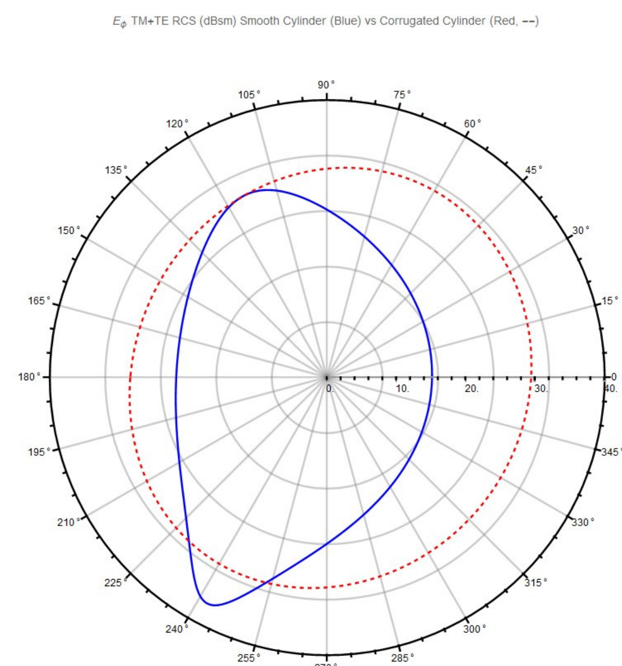


Figure 7-136 Polar Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE modes and Corrugated Cylinder hybrid mode for Run a_plus_b.0.1.0.0

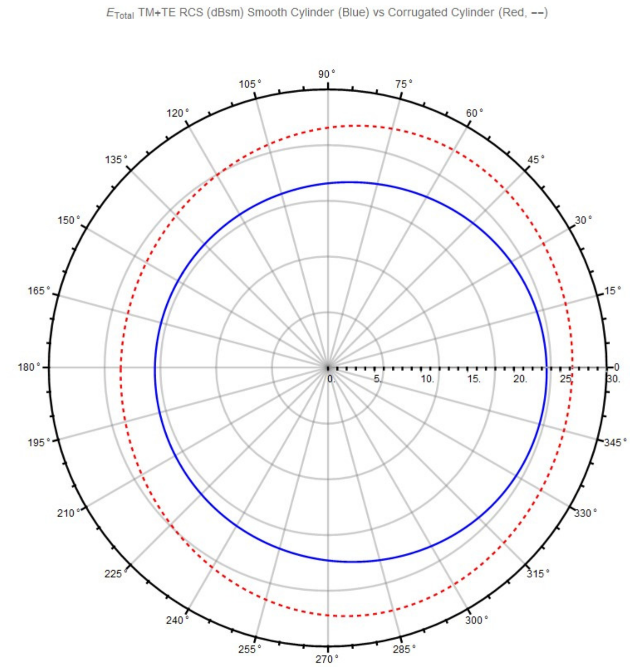


Figure 7-137 Polar Plot form of RCS dBsm for ETotal of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.0.1.0.0

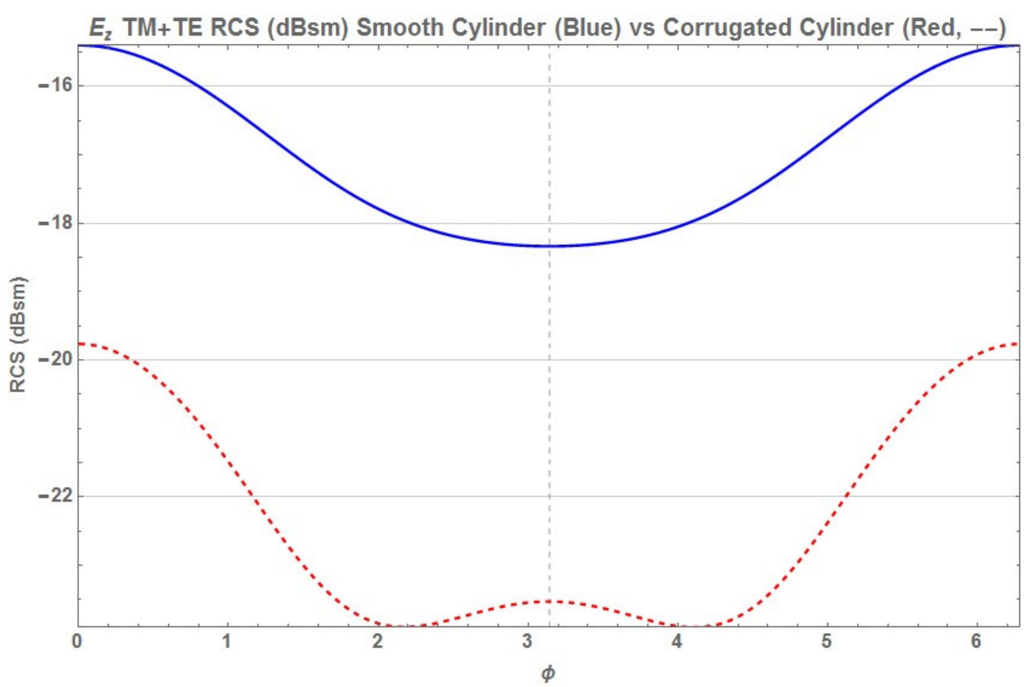


Figure 7-138 XY Plot form of RCS dBsm for Ez of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.0.1.0.0

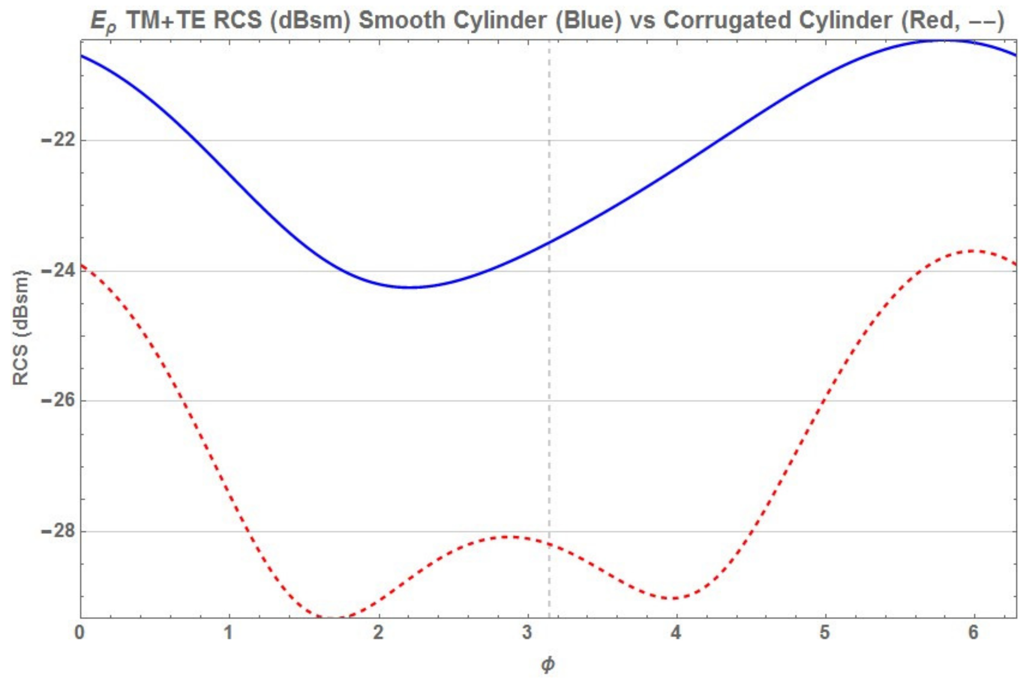


Figure 7-139 XY Plot form of RCS dBsm for Eρ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.0.1.0.0

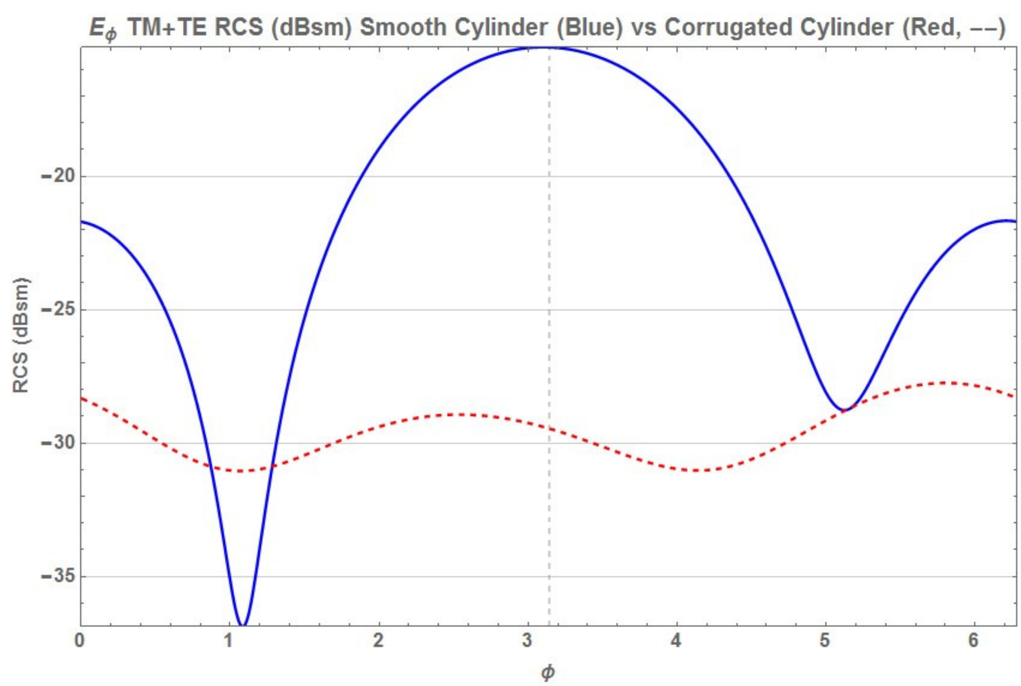


Figure 7-140 XY Plot form of RCS dBsm for Eφ of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.0.1.0.0

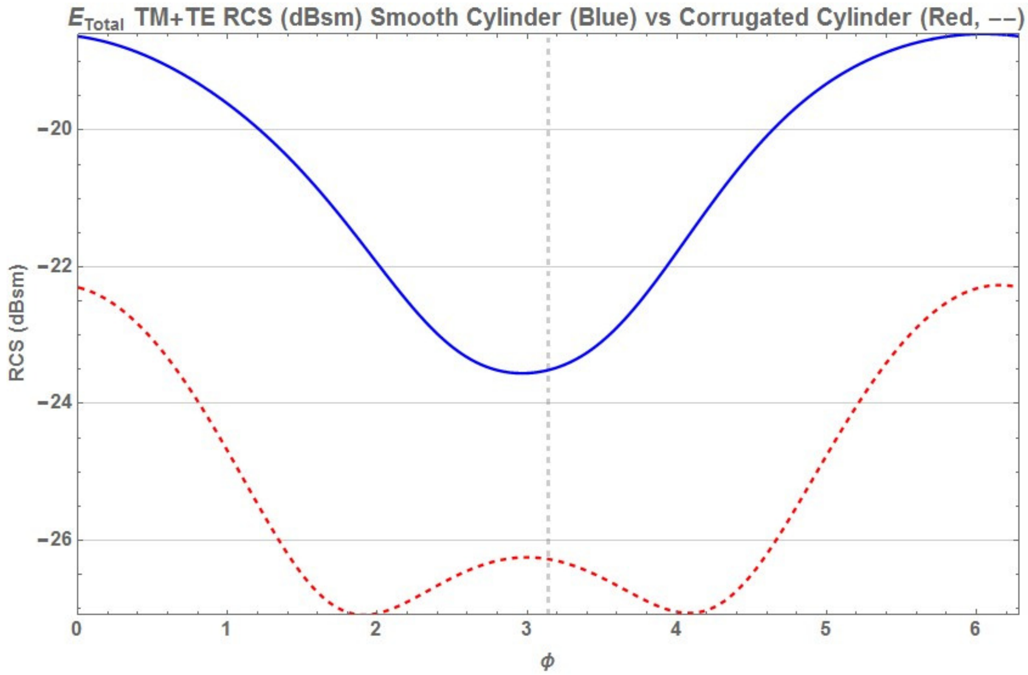


Figure 7-141 XY Plot form of RCS dBsm for E_{Total} of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.0.1.0.0

Table 21 Detailed parameters summary for changing ρ plots of Run a_plus_b.0.1.0.0

	value	min	max	delta	Qty of Points
E0	1. V/m	–	–	–	–
H0	0.00265258 A/m	–	–	–	–
λ	0.019986 m	–	–	–	–
Frequency	1.50105×10^{10} Hz	–	–	–	–
a	0.0001 λ	–	–	–	–
b	0.1 λ	–	–	–	–
ρ1	0.0999 λ	–	–	–	–
ρ2	0.1 λ	–	–	–	–
ρ range	–	0.08991 λ	10.1 λ	0.0182333 λ	550
ϕ (observed)	37. Deg	–	–	–	–
z (observed)	0.25 a && 0.000025 λ	–	–	–	–
Matching Points	–	–	–	–	7
θi	55. Deg	–	–	–	–
ϕi	37. Deg	–	–	–	–
n	–	-3	3	1	7
l	–	0	0	1	1
m	–	0	0	1	1
max allowable m	0.0002	–	–	–	–
max allowable l	0.042685	–	–	–	–
Boundary	Boundary a+b	–	–	–	–

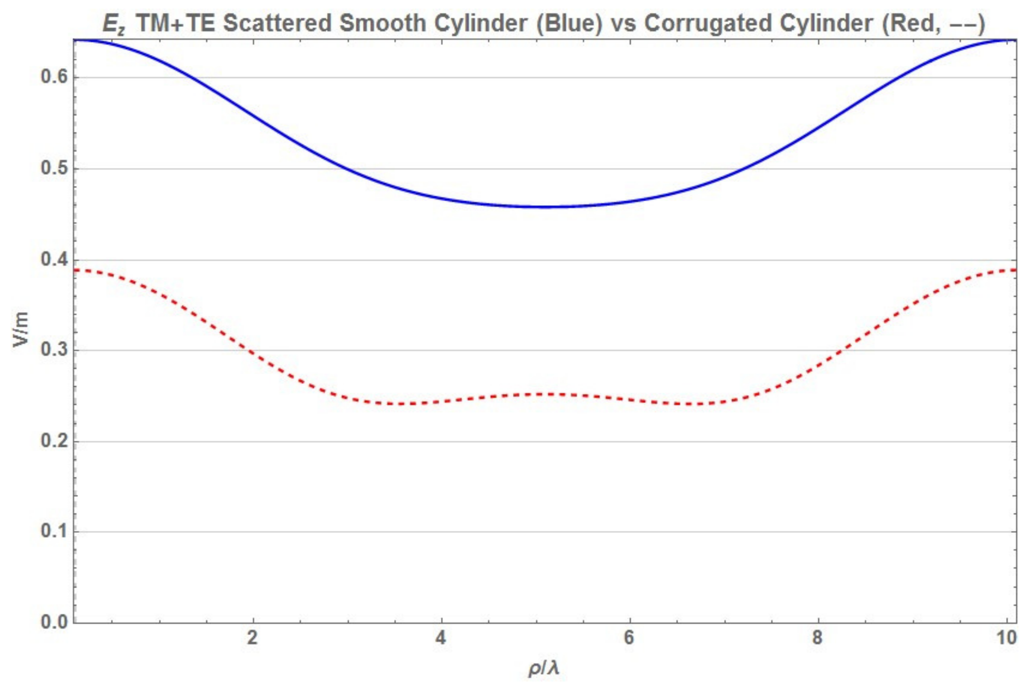


Figure 7-142 XY Plot of Scattered Field Amplitude Only, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.0.10.0.0

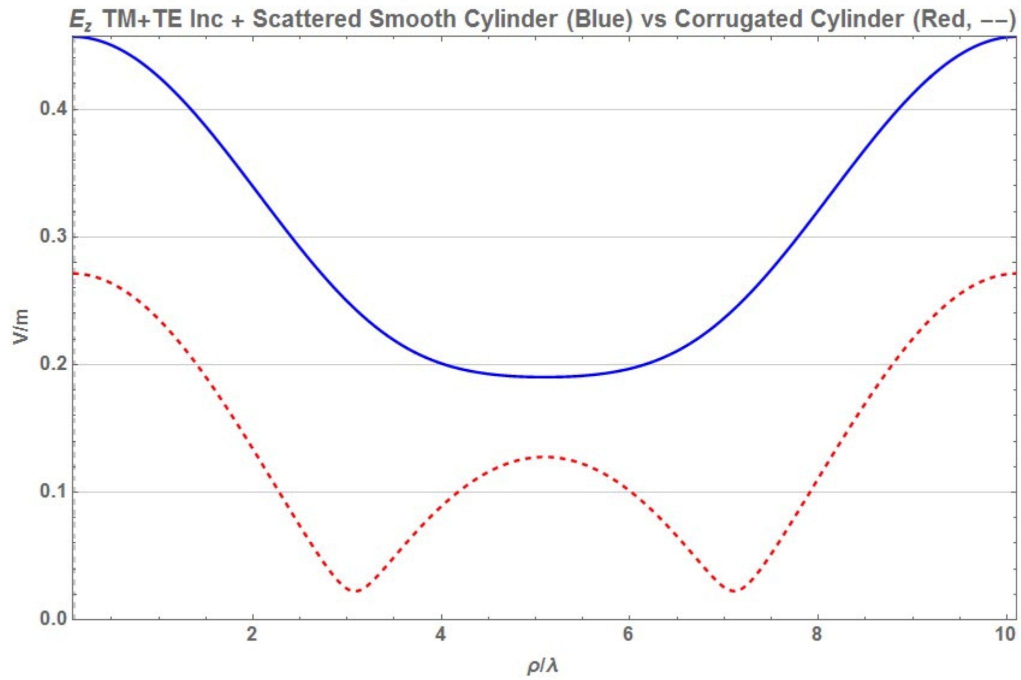


Figure 7-143 XY Plot of Scattered + Incident Field Amplitude Only, for Ez, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.0.10.0.0

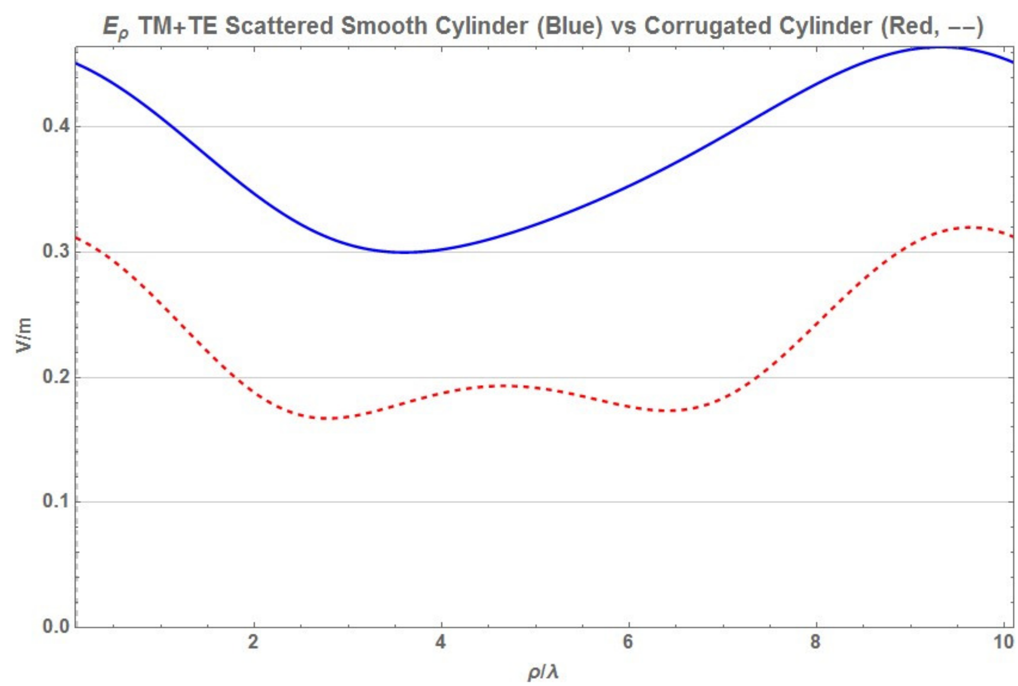


Figure 7-144 XY Plot of Scattered Field Amplitude Only, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.0.10.0.0

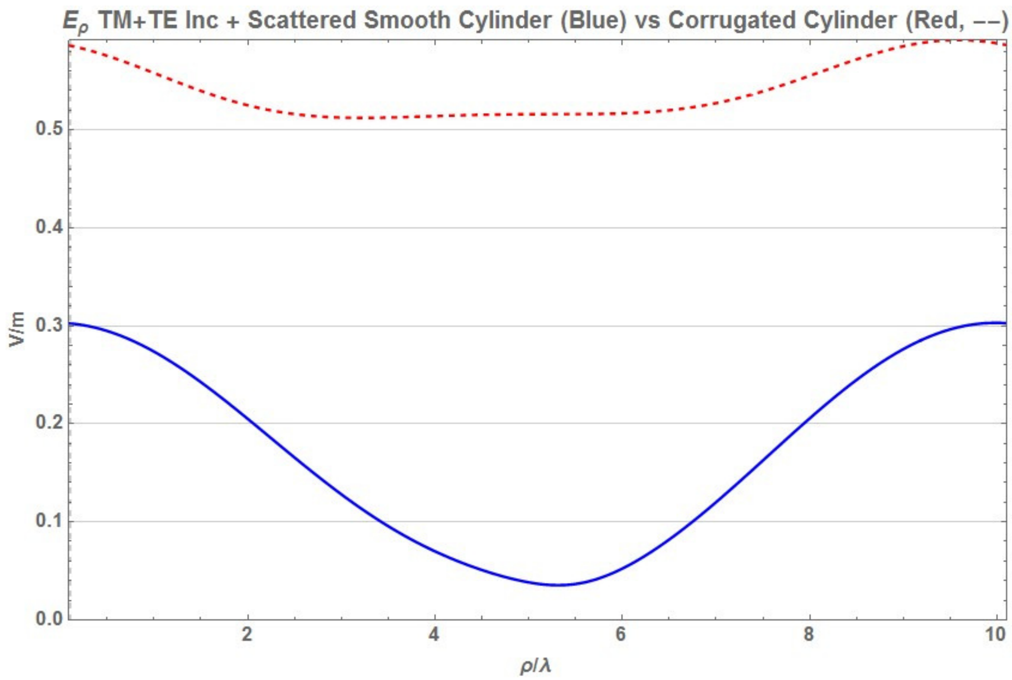


Figure 7-145 XY Plot of Scattered + Incident Field Amplitude Only, for Eρ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.0.10.0.0

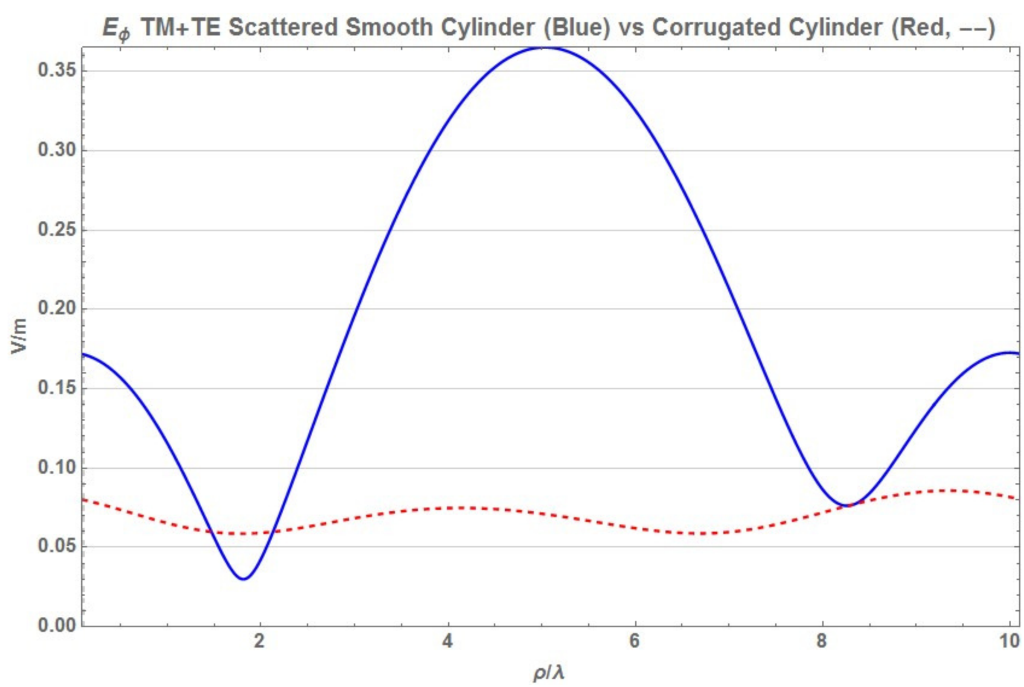


Figure 7-146 XY Plot of Scattered Field Amplitude Only, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.0.10.0.0

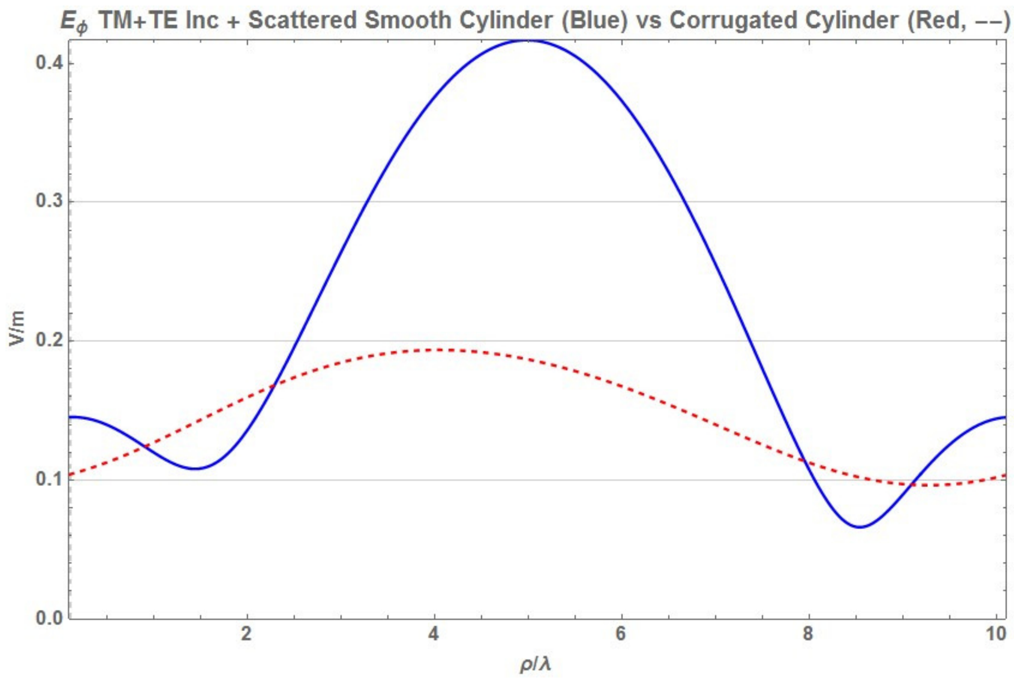


Figure 7-147 XY Plot of Scattered + Incident Field Amplitude Only, for Eφ, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.0.10.0.0

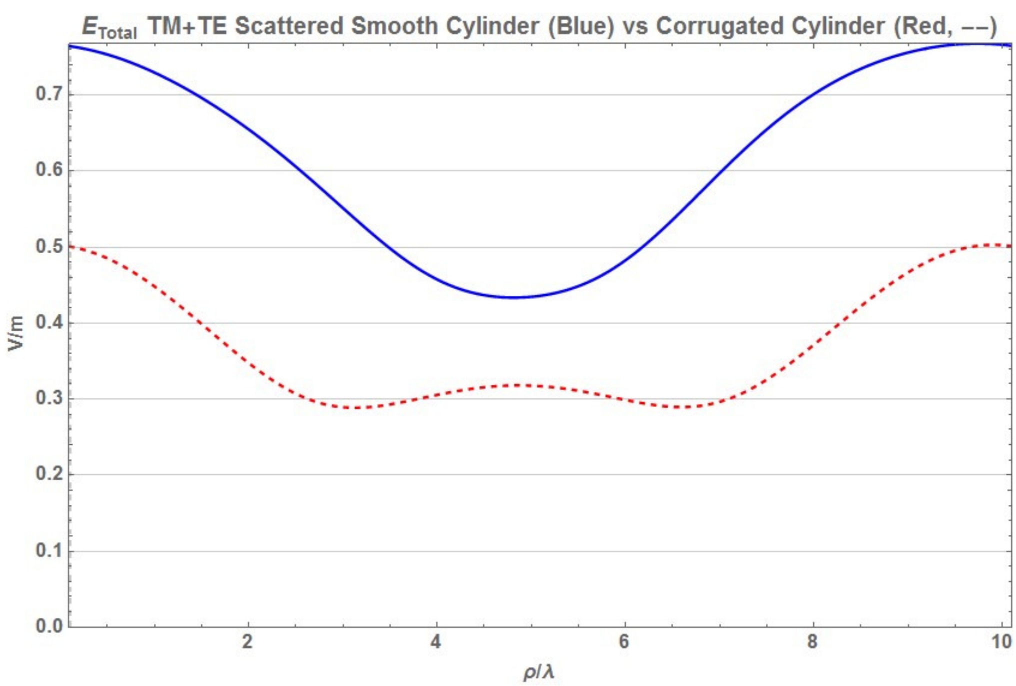


Figure 7-148 XY Plot of Scattered Field Amplitude Only, for E_{Total}, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.0.10.0.0

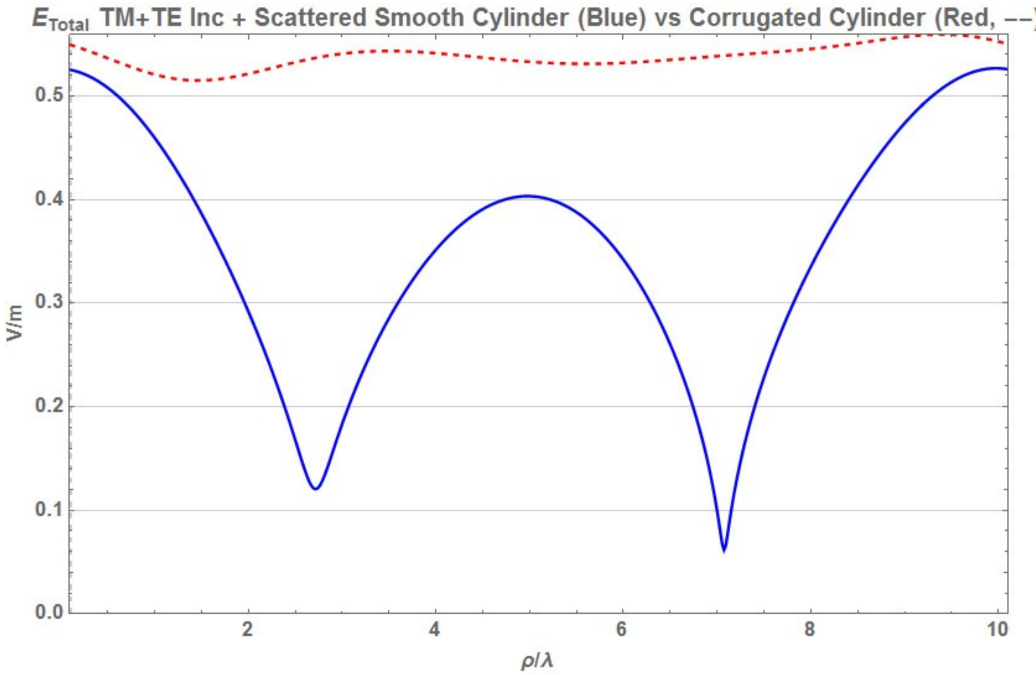


Figure 7-149 XY Plot of Scattered + Incident Field Amplitude Only, for ETotal, of the Smooth Cylinder TM + TE mode and Corrugated Cylinder hybrid mode for Run a_plus_b.0.10.0.0

7.5.10 Comparison to Other Corrugated Cylinder Methods

Table 22 Detailed parameters summary for changing φ plots of Run a_plus_b.compare

	value	min	max	delta	Qty of Points
E0	1. V/m	-	-	-	-
H0	2.65258×10^{-7} A/m	-	-	-	-
λ	1.50105×10^{10} Hz	-	-	-	-
Frequency	0.019986 m	-	-	-	-
a	0.2 λ	-	-	-	-
b	0.1 λ	-	-	-	-
ρ1	0.4 λ	-	-	-	-
ρ2	0.5 λ	-	-	-	-
φ range	-	0.327273 Deg	359.673 Deg	0.654545 Deg	550
ρ (observed)	10. λ	-	-	-	-
z (observed)	0.25 a && 0.05 λ	-	-	-	-
Matching Points	-	-	-	-	7
θi	85. Deg	-	-	-	-
φi	0.01 Deg	-	-	-	-
n	-	-3	3	1	7
l	-	0	0	1	1
m	-	0	0	1	1
max allowable m	0.4	-	-	-	-
max allowable l	0.273853	-	-	-	-
Boundary	Boundary a+b	-	-	-	-

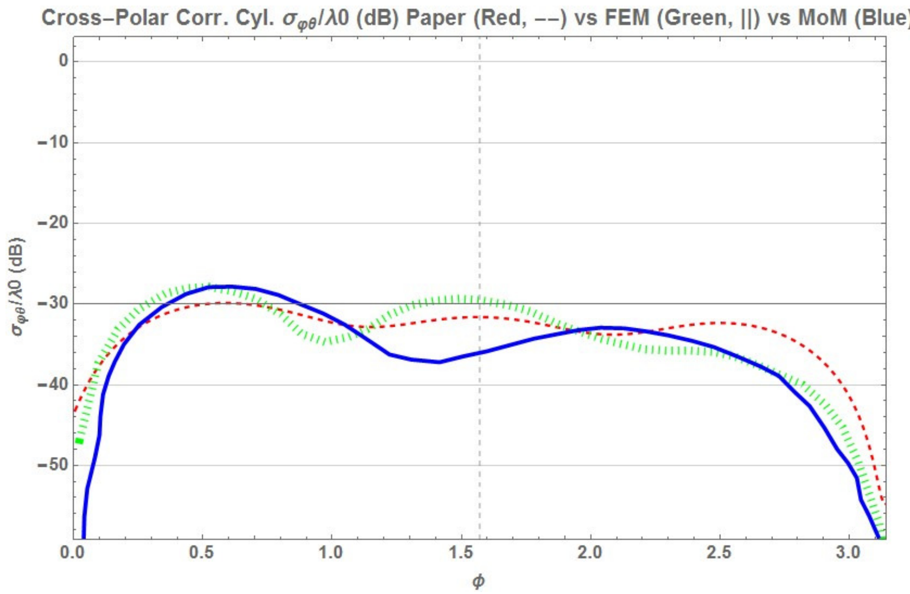
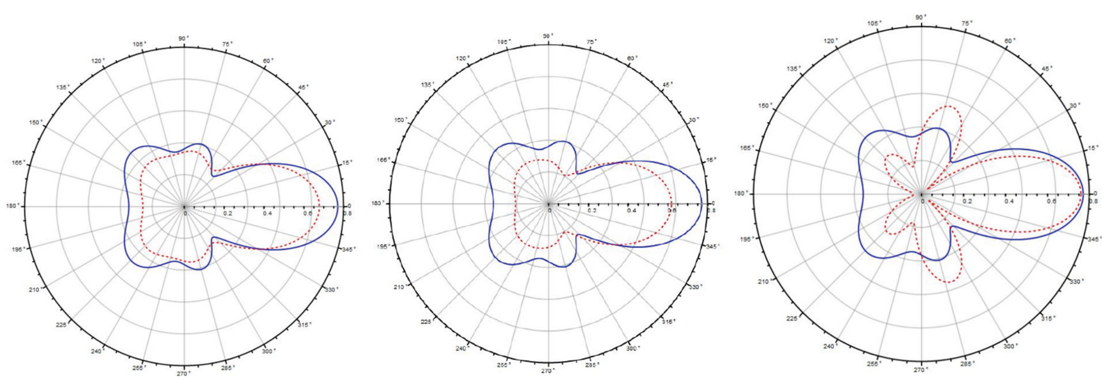


Figure 7-150 XY Plot form of Cross-Polar Corrugated Cylinder $\sigma_{\phi\theta}/\lambda_0$ (dB) for Run a_plus_b.Compared with results of the finite element method (FEM) and method of moments (MoM) from [27]

7.5.11 Varied Dielectric Constant with Comparisons

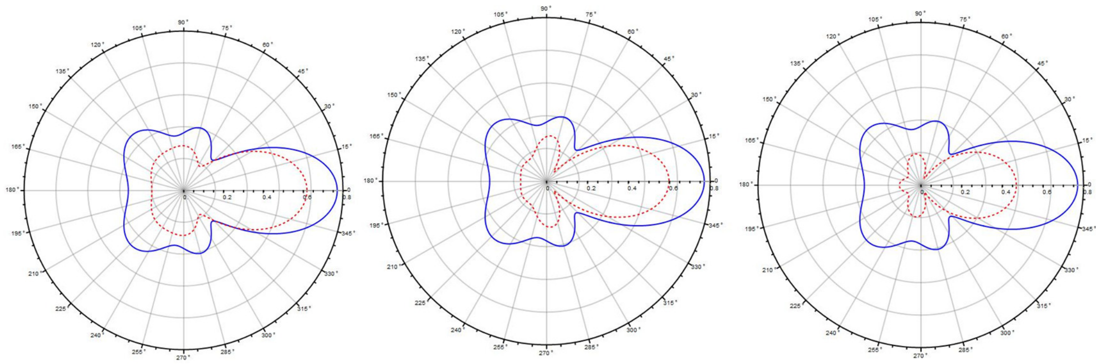
The scattered axial field (E_z for TM_z mode) vs φ is plotted below in Figure 7-151 for several cases of interest with a varying dielectric constant, ε_r., where ε_r is selected to be a real value only. It is evident that the corrugated cylinder with dielectric loading has a generally reduced scattered field compared to the smooth cylinder.



(a) (b) (c)

Figure 7-151 Comparison of scattered axial fields of a smooth cylinder (solid line) with the corrugated cylinder (dotted line) with lossless dielectric loading of dielectric constant: $\varepsilon_r=1$ a) $\varepsilon_r=4$; b) $\varepsilon_r=9$; c)

Now, the case of lossy dielectric loading (ε_r is a complex value) is examined with results of the scattered axial field (E_z for TM_z mode) vs φ is plotted below in Figure 7-152 for several cases. ε_r.



(a) (b) (c)

Figure 7-152 Axial scattered fields of a smooth cylinder (solid line) and corrugated cylinder (dotted line) with lossy dielectric loading. Dielectric constants: a) ε_r=4 – j1; b) ε_r=6.29; c) ε_r=6.29 – j1.73

It is evident that a complex permittivity yields a generally smaller scattered field. The dielectric constants in (b) and (c) were chosen to correspond to those found in [33]. These results can also be found in [34] with some additional discussion.

7.6 Conclusions

This dissertation presented an alternate method to calculate the scattered field of a corrugated cylinder. The method of utilizing a hybrid mode of TM_z and TE_z with a radial waveguide representation of the corrugation was demonstrated.

In the comparison between the periodic corrugated cylinder model from this research and the model of a smooth cylinder, there was a lot of agreement between the fields of both models with the exception of the E_ϕ fields. There are a few reasons for this:

- This technique is an approximation and as such, will have errors

- More specifically, the corrugated cylinder is approximated to a smooth cylinder, by shrinking the ‘a’ dimension, though not eliminated, so there will be some artifacts that make it different than a smooth cylinder
- Also, the cross-polarization nature of the problem allows fields to manifest themselves between TE and TM modes

What’s important to note is what the limitations are and what the capabilities are of a given method, to know when to best apply it or to seek an alternative method.

Overall, good agreement was attained between the periodic corrugated cylinder model from this research and the model of a smooth cylinder, for small corrugation openings approximating a smooth cylinder, where the relative dimensions of the corrugated cylinder where much greater than λ (optical region as indicated in Figure 7-1). There was some agreement in this same comparison at the Rayleigh scattering region, where the relative dimensions of the corrugated cylinder where much less than λ. There was also good agreement attained between the periodic corrugated model of this research when compared to the referenced periodic corrugated cylinder FEM and MoM techniques, which was modeled in the Rayleigh scattering region.

It is concluded that the techniques discussed in this dissertation is most suitable for the optical region, where the λ of interest is much smaller than the dimensions of the periodic corrugated cylinder of interest. Also, from the results when compared to the FEM and MoM techniques, it is concluded that the Rayleigh scattering region, where the λ of interest is much larger than the dimensions of the periodic corrugated cylinder of interest, is suitable for the technique presented in this research.

7.7 Open Questions for Future Research

There were limitations discovered of the presented method that merits future research. Many challenges were faced, especially in the numerical solving of the unknown expansion coefficients which could benefit from improved methods. Also, there were many techniques that were investigated but not implemented. Here is a list of future work that can enhance or build on this research:

- Improve or replace the computationally expensive loop solve method.
- Summation truncation using the symmetry of the summation as in [12, p. 603].
- Using Poisson's sum formula to reduce any of the infinite series equations prior to truncation [34].
- Comparison models for varying the permittivity of region I, representing the dielectric loading of the periodic corrugated cylinder
- Using Eigenvalues and Eigenvectors solutions approach for solving the unknown expansion coefficients
- Using a continuous periodic function approximation to asymptotically represent the periodicity of the corrugated cylinder
- Applying the asymptotic boundary condition method, [26], for improving solution agreement in regimes where the relative dimensions of the corrugated cylinder are that of λ.
- Construction and testing of a physical model for further comparison

APPENDICES

APPENDIX A

Mathematica® code for the modeling and comparison of the scattered field of a periodic cylinder and smooth circular cylinder

Plane-wave Scattering of a Periodic Corrugated Cylinder

```
In[44721]:= ClearAll["Global`*"]
SetDirectory["C:\Users\Samuel\Google Drive\Dissertation\FinalData"];
(* Output Directory*)
```

Electromagnetic Constants, Adjustable, & Calculated Parameters

```
In[44723]:= zerooffset = 0; (*Factor to eliminate zeros that would show up in denominators*)
```

Electromagnetic Constants

```
In[44724]:= μ0 = (4 π)/10^7;
In[44725]:= ε0 = 1/((36 π) 10^9);
In[44726]:= c = 1/Sqrt[ε0 μ0];
In[44727]:= eta0 = Sqrt[μ0/ε0];
```

Incident Electric and Magnetic Field Parameters

```
In[44728]:= E0 = 1;
In[44729]:= H0 = E0/eta0;
(*H0=1;*)
In[44730]:= lambda0 = 0.019986;
In[44731]:= (* ϕi -->
 see section 'User Inputs-Geometric Parameters' for value of incident phi, ϕi *)
```

Printed by Wolfram Mathematica Student Edition


```
In[44732]:= θi = (55 π/180);
θs = π - θi; (*Balanis, p 615*)
```

Material Electromagnetic Parameters

```
In[44734]:= de1 = 1;  (*Region 1 relative dielectric constant*)
In[44735]:= de2 = 1;  (*Region 2 relative dielectric constant*)
In[44736]:= dm1 = 1;  (*Region 1 relative permeability constant*)
In[44737]:= dm2 = 1;  (*Region 2 relative permeability constant*)
```

Electromagnetic Calculated Parameters

```
In[44738]:= f0 = c / lambda0;            (* Frequency (hertz or 1/seconds)*)
In[44739]:= ω0 = 2 * Pi * f0;            (* Angular velocity (rad/seconds)*)
In[44740]:= ε1 = de1 * ε0;               (* Region 1 permitivity (farads/meter)*)
In[44741]:= ε2 = de2 * ε0;               (* Region 2 permitivity (farads/meter)*)
In[44742]:= μ1 = dm1 * μ0;               (* Region 1 permeability (henries/meter)*)
In[44743]:= μ2 = dm2 * μ0;               (* Region 2 permeability (henries/meter)*)
In[44744]:= eta1 = Sqrt[μ1 / ε1];        (* Intrinsic impedance in regin 1 (ohms)*)
In[44745]:= eta2 = Sqrt[μ2 / ε2];        (* Intrinsic impedance in regin 2 (ohms)*)
In[44746]:= beta0 = ω0 * Sqrt[μ0 * ε0];  (* Beta (wavenumber) in free space (rad/meter)*)
In[44747]:= k1 = ω0 * Sqrt[μ1 * ε1];     (* Wavenumber in region 1 (rad/meter)*)
In[44748]:= k2 = ω0 * Sqrt[μ2 * ε2];     (* Wavenumber in region 2 (rad/meter)*)
In[44749]:= Z2 = ω0 * μ2 / k2;           (*Intrinsic Impedance of region 2 *)
```

Wavenumber of Fields

```
In[44750]:= kzm[m_] := N[(- (m + zerooffset) * Pi) / a];          (*Z-axis of field in region 1*)
In[44751]:= kρm[m_] := Sqrt[(k1^2) - (kzm[m]^2)] ;                 (*rho-axis of field in region 1*)
In[44752]:= kzl[l_] := (k2 * Cos[θi]) + (2 * Pi * l / (b + a));  (*Z-axis of field in region 2*)
In[44753]:= kρl[l_] := Sqrt[(k2^2) - (kzl[l]^2)] ;                 (*rho-axis of field in region 2*)
```

Corrugated Cylinder Region I, II, and 'b' Equations and

Printed by Wolfram Mathematica Student Edition

Variables

Region I Fields - Equations and Variables

Equations [NOT USED]

```
In[44754]:= EIz[Anm_] := hnmj ** Anm;
In[44755]:= HIz[Bnm_] := inmj ** Bnm;
In[44756]:= EIphi[Anm_, Bnm_] := (knmj ** Anm) + (lnmj ** Bnm);
In[44757]:= HIphi[Anm_, Bnm_] := (onmj ** Anm) + (pnmj ** Bnm);
In[44758]:= EIrho[Anm_, Bnm_] := (rnmj ** Anm ) + (snmj ** Bnm);
In[44759]:= HIrho[Anm_, Bnm_] := (tnmj ** Anm) + (unmj ** Bnm);
```

Variables

```
In[44760]:= hnmjeq[ϕ_, z_, ρ_] := e^(i n ϕ) Cos[kzm[m] * (z - a/2) ]
  (HankelH1[n, kρm[m] ρ] - HankelH2[n, kρm[m] ρ] HankelH1[n, kρm[m] ρ1] / HankelH2[n, kρm[m] ρ1]);

In[44761]:= inmjeq[ϕ_, z_, ρ_] :=
  e^(i n ϕ) Sin[kzm[m] (z - a/2)] (HankelH1[n, kρm[m] ρ] - (HankelH2[n, kρm[m] ρ]
      (kρm[m] HankelH1[n - 1, kρm[m] ρ1] - n HankelH1[n, kρm[m] ρ1] / ρ1)) /
    (kρm[m] HankelH2[n - 1, kρm[m] ρ1] - n HankelH2[n, kρm[m] ρ1] / ρ1));

In[44762]:= knmjeq[ϕ_, z_, ρ_] := 1/ρ n e^(i n ϕ) kzm[m]/kρm[m]^2 Cos[(z - a/2) kzm[m]]
  (HankelH1[n, kρm[m] ρ] - HankelH2[n, kρm[m] ρ] HankelH1[n, kρm[m] ρ1] / HankelH2[n, kρm[m] ρ1]);

In[44763]:= lnmjeq[ϕ_, z_, ρ_] := i μ1 ω0 e^(i n ϕ) 1/kρm[m]^2 Sin[(z - a/2) kzm[m]]
  (1/2 kρm[m] (HankelH1[n - 1, kρm[m] ρ] - HankelH1[n + 1, kρm[m] ρ]) -
    (kρm[m] (HankelH2[n - 1, kρm[m] ρ] - HankelH2[n + 1, kρm[m] ρ])
      (kρm[m] HankelH1[n - 1, kρm[m] ρ1] - n HankelH1[n, kρm[m] ρ1] / ρ1)) /
    (2 (kρm[m] HankelH2[n - 1, kρm[m] ρ1] - n HankelH2[n, kρm[m] ρ1] / ρ1)));
```

Printed by Wolfram Mathematica Student Edition


```
In[44764]:= onmjeq[ϕ_, z_, ρ_] := -i ε1 ω0 e^(i n ϕ) 1/kρm[m]^2 Cos[(z - a/2) kzm[m]]
  (1/2 kρm[m] (HankelH1[n - 1, kρm[m] ρ] - HankelH1[n + 1, kρm[m] ρ]) -
    (kρm[m] (HankelH2[n - 1, kρm[m] ρ] - HankelH2[n + 1, kρm[m] ρ])
      HankelH1[n, kρm[m] ρ1]) / (2 HankelH2[n, kρm[m] ρ1]));

In[44765]:= pnmjeq[ϕ_, z_, ρ_] :=
  1/ρ n e^(i n ϕ) kzm[m]/kρm[m]^2 Sin[(z - a/2) kzm[m]] (HankelH1[n, kρm[m] ρ] - (HankelH2[n, kρm[m] ρ]
      (kρm[m] HankelH1[n - 1, kρm[m] ρ1] - n HankelH1[n, kρm[m] ρ1] / ρ1)) /
    (kρm[m] HankelH2[n - 1, kρm[m] ρ1] - n HankelH2[n, kρm[m] ρ1] / ρ1));

In[44766]:= rnmjeq[ϕ_, z_, ρ_] := -i e^(i n ϕ) kzm[m]/kρm[m]^2 Cos[kzm[m] (z - a/2)]
  (1/2 kρm[m] (HankelH1[n - 1, kρm[m] ρ] - HankelH1[n + 1, kρm[m] ρ]) -
    (kρm[m] (HankelH2[n - 1, kρm[m] ρ] - HankelH2[n + 1, kρm[m] ρ])
      HankelH1[n, kρm[m] ρ1]) / (2 HankelH2[n, kρm[m] ρ1]));

In[44767]:= snmjeq[ϕ_, z_, ρ_] := (μ1 ω0)/ρ n e^(i n ϕ) 1/kρm[m]^2
  Sin[(z - a/2) kzm[m]] (HankelH1[n, kρm[m] ρ] - (HankelH2[n, kρm[m] ρ]
      (kρm[m] HankelH1[n - 1, kρm[m] ρ1] - n HankelH1[n, kρm[m] ρ1] / ρ1)) /
    (kρm[m] HankelH2[n - 1, kρm[m] ρ1] - n HankelH2[n, kρm[m] ρ1] / ρ1));

In[44768]:= tnmjeq[ϕ_, z_, ρ_] := - (ω0 ε1)/ρ n e^(i n ϕ) 1/kρm[m]^2 Cos[kzm[m] (z - a/2)]
  (HankelH1[n, kρm[m] ρ] - HankelH2[n, kρm[m] ρ] HankelH1[n, kρm[m] ρ1] / HankelH2[n, kρm[m] ρ1]);

In[44769]:= unmjeq[ϕ_, z_, ρ_] := -i e^(i n ϕ) kzm[m]/kρm[m]^2 Sin[(z - a/2) kzm[m]]
  (1/2 kρm[m] (HankelH1[n - 1, kρm[m] ρ] - HankelH1[n + 1, kρm[m] ρ]) -
    (kρm[m] (HankelH2[n - 1, kρm[m] ρ] - HankelH2[n + 1, kρm[m] ρ])
      (kρm[m] HankelH1[n - 1, kρm[m] ρ1] - n HankelH1[n, kρm[m] ρ1] / ρ1)) /
    (2 (kρm[m] HankelH2[n - 1, kρm[m] ρ1] - n HankelH2[n, kρm[m] ρ1] / ρ1)));
```

Printed by Wolfram Mathematica Student Edition

Region II Fields - Equations and Variables

Equations [NOT USED]

```
EIIzb[Cnlb_] := bnj + (anlj ** Cnlb);
EIIza[Cnla_] := bnj + (anlj ** Cnla);
HIIzb[Dnlb_] := fnj + (gnlj ** Dnlb);
HIIza[Dnla_] := fnj + (gnlj ** Dnla);
EIIphib[Cnlb_, Dnlb_] := dnj + ((cnlj ** Cnlb) + (enlj ** Dnlb));
EIIphia[Cnla_, Dnla_] := dnj + ((cnlj ** Cnla) + (enlj ** Dnla));
HIIphib[Cnlb_, Dnlb_] := mnj + ((nnlj ** Cnlb) + (qnlj ** Dnlb));
HIIphia[Cnla_, Dnla_] := mnj + ((nnlj ** Cnla) + (qnlj ** Dnla));
EIIrhob[Cnlb_, Dnlb_] := γnj + ((vnlj ** Cnlb) + (wnlj ** Dnlb));
(*Not utilized for solving unknown coeff.*)
EIIrhoa[Cnla_, Dnla_] := γnj + ((vnlj ** Cnla) + (wnlj ** Dnla));
(*Not utilized for solving unknown coeff.*)
HIIrhob[Cnlb_, Dnlb_] := δnj + ((xnlj ** Cnlb) + (ynlj ** Dnlb));
(*Not utilized for solving unknown coeff.*)
HIIrhoa[Cnla_, Dnla_] := δnj + ((xnlj ** Cnla) + (ynlj ** Dnla));
(*Not utilized for solving unknown coeff.*)
```

Variables

```
bnjeq[ϕ_, z_, ρ_] := Sum[E0 Sin[θi] e^(i k2 z Cos[θi]) BesselJ[n1, k2 ρ Sin[θi]] e^(i n1 ϕ) / i^n1, {n1, nmin, nmax}];
(*Keep in summation form, INCIDENT FIELD From E_z^II=[b_nj]+[a_nl][C_nl] *)
anljeq[ϕ_, z_, ρ_] := e^(i n ϕ) HankelH2[n, kρ1[1] ρ] / e^(i kz1[1] z); (*E_z^II=[b_nj]+[a_nl][C_nl]*)
fnjeq[ϕ_, z_, ρ_] := Sum[H0 Sin[θi] e^(i k2 z Cos[θi]) BesselJ[n1, k2 ρ Sin[θi]] e^(i n1 ϕ) / i^n1, {n1, nmin, nmax}];
(*Keep sum, INCIDENT FIELD From H_z^II=[f_nj]+[g_nl][D_nl]*)
gnljeq[ϕ_, z_, ρ_] := e^(i n ϕ) HankelH2[n, kρ1[1] ρ] / e^(i kz1[1] z); (*H_z^II=[f_nj]+[g_nlj][D_nl]*)
```



```
dnjeq[ϕ_, z_, ρ_] :=
  Sum[(-E0 Sin[ϕ] Cos[θi] + Z2 H0 Cos[ϕ]) e^(i k2 z Cos[θi]) BesselJ[n1, k2 ρ Sin[θi]] e^(i n1 ϕ) / i^n1, {n1, nmin, nmax}];
(*Keep sum, INCIDENT FIELD From E_ϕ^II=[d_nj]+([c_nlj][C_nl]+[e_nlj][D_nl])*)
cnljeq[ϕ_, z_, ρ_] := 1/ρ (n + zerooffset) e^(i n ϕ) kz1[1] HankelH2[n, kρ1[1] ρ] / (kρ1[1]^2 e^(i kz1[1] z));
(*E_ϕ^II=[d_nj]+([c_nlj][C_nl]+[e_nlj][D_nl])*)
(*enljeq[ϕ_,z_,ρ_] := i μ2 ω0 e^(i n ϕ) (HankelH2[n-1,kρ1[1] ρ]-HankelH2[n+1,kρ1[1] ρ]) / (2kρ1[1] e^(i z kz1[1]));
(*E_ϕ^II=[d_nj]+([c_nlj][C_nl]+[e_nlj][D_nl])*)*)
enljeq[ϕ_, z_, ρ_] :=
  i μ2 ω0 e^(i n ϕ) (HankelH2[n - 1, kρ1[1] ρ] - (n * HankelH2[n, kρ1[1] ρ] / ρ)) / (kρ1[1]^2 e^(i z kz1[1]));
(*E_ϕ^II=[d_nj]+([c_nlj][C_nl]+[e_nlj][D_nl])*) (*updated 11/12/2016*)
mnjeq[ϕ_, z_, ρ_] :=
  Sum[(-E0 Cos[ϕ]/Z2 - H0 Sin[ϕ] Cos[θi]) e^(i k2 z Cos[θi]) BesselJ[n1, k2 ρ Sin[θi]] e^(i n1 ϕ) / i^n1, {n1, nmin, nmax}];
(*Keep sum, INCIDENT FIELD From H_ϕ^II=[m_nj]+([n_nlj][C_nl]+[q_nlj][D_nl])*)
nnljeq[ϕ_, z_, ρ_] := -i ε2 ω0 e^(i n ϕ) (HankelH2[n - 1, kρ1[1] ρ] - HankelH2[n + 1, kρ1[1] ρ]) / (2 kρ1[1] e^(i z kz1[1]));
(*H_ϕ^II=[m_nj]+([n_nlj][C_nl]+[q_nlj][D_nl])*)
qnljeq[ϕ_, z_, ρ_] := 1/ρ e^(i n ϕ) (n + zerooffset) kz1[1] HankelH2[n, kρ1[1] ρ] / (kρ1[1]^2 e^(i kz1[1] z));
(*H_ϕ^II=[m_nj]+([n_nlj][C_nl]+[q_nlj][D_nl])*)
γnjeq[ϕ_, z_, ρ_] :=
  Sum[(E0 Cos[ϕ] Cos[θi] + Z2 H0 Sin[ϕ]) e^(i k2 z Cos[θi]) BesselJ[n1, k2 ρ Sin[θi]] e^(i n1 ϕ) / i^n1, {n1, nmin, nmax}];
(*Keep sum, INCIDENT FIELD From E_ρ^II=[γ_nj]+([v_nlj][C_nl]+[w_nlj][D_nl])*)
vnljeq[ϕ_, z_, ρ_] := -i e^(i n ϕ)
  (kz1[1] (HankelH2[n - 1, kρ1[1] ρ] - HankelH2[n + 1, kρ1[1] ρ])) / (2 kρ1[1] e^(i z kz1[1]));
(*E_ρ^II=[γ_nj]+([v_nlj][C_nl]+[w_nlj][D_nl])*)
wnljeq[ϕ_, z_, ρ_] := (μ2 ω0)/ρ (n + zerooffset) e^(i n ϕ) HankelH2[n, kρ1[1] ρ] / (kρ1[1]^2 e^(i kz1[1] z));
(*E_ρ^II=[γ_nj]+([v_nlj][C_nl]+[w_nlj][D_nl])*)
δnjeq[ϕ_, z_, ρ_] :=
  Sum[(-E0 Sin[ϕ]/Z2 + H0 Cos[ϕ] Cos[θi]) e^(i k2 z Cos[θi]) BesselJ[n1, k2 ρ Sin[θi]] e^(i n1 ϕ) / i^n1, {n1, nmin, nmax}];
(*Keep sum, INCIDENT FIELD From H_ρ^II=[δ_nj]+([x_nlj][C_nl]+[y_nlj][D_nl])*)
```



```
xnljeq[ϕ_, z_, ρ_] := -(ω0 ε2)/ρ e^(i n ϕ) (n + zerooffset) HankelH2[n, kρ1[1] ρ] / (kρ1[1]^2 e^(i kz1[1] z));
(*H_ρ^II=[δ_nj]+([x_nlj][C_nl]+[y_nlj][D_nl])*)
ynljeq[ϕ_, z_, ρ_] := -i e^(i n ϕ)
  (kz1[1] (HankelH2[n - 1, kρ1[1] ρ] - HankelH2[n + 1, kρ1[1] ρ])) / (2 kρ1[1] e^(i z kz1[1]));
(*H_ρ^II=[δ_nj]+([x_nlj][C_nl]+[y_nlj][D_nl])*)
```

Boundary 'b' Fields - Equations and Variables [REQUIRED FOR CODE OPERATION ONLY]

```
(*This section of code was implemented for a prior approach. The values
 and solutions of this section play no role in the final results,
however, removing them without compensating for the code structure
 change would render the entirety of this code in operable. Therefore,
this code section will remain intact and up to a future user to decide
 whether or not to remove this section and make the necessary code
 changes. It would help to have context in order to make the change,
though not necessary. The associated dissertation paper would
 suffice [PLANE-WAVE SCATTERING OF A PERIODIC CORRUGATED CYLINDER,
  by Samuel Garcia, Florida Atlantic University, Boca Raton, FL, November 2016]*)
```

Equations [NOT USED]

```
Ebz[Enl_] := Ebα ** Enl; (*Update with correct array length later*)
Ebphi[Enl_, Fnl_] := ((Ebα ** Enl) + (Ebα ** Fnl));
(*Update with correct array length later*)
Hbz[Fnl_] := δ2nlj ** Fnl;
Hbphi[Enl_, Fnl_] := ((δ3nlj ** Enl) + (δ4nlj ** Fnl));
Ebrho[Enl_, Fnl_] := ((δ5nlj ** Enl) + (δ6nlj ** Fnl));
Hbrho[Enl_, Fnl_] := ((δ7nlj ** Enl) + (δ8nlj ** Fnl));
```

Variables [REQUIRED FOR CODE OPERATION ONLY]

```
δ2nljeq[ϕ_, z_, ρ_] := e^(i n ϕ) HankelH2[n, kρ1[1] ρ] / e^(i kz1[1] z);
δ3nljeq[ϕ_, z_, ρ_] := -i ε2 ω0 e^(i n ϕ) (HankelH2[n - 1, kρ1[1] ρ] - HankelH2[n + 1, kρ1[1] ρ]) / (2 kρ1[1] e^(i z kz1[1]));
δ4nljeq[ϕ_, z_, ρ_] := 1/ρ e^(i n ϕ) n kz1[1] HankelH2[n, kρ1[1] ρ] / (kρ1[1]^2 e^(i kz1[1] z));
```



```
δ5nljeq[ϕ_, z_, ρ_] := -i e^(i n ϕ)
  (kz1[1] (HankelH2[n - 1, kρ1[1] ρ] - HankelH2[n + 1, kρ1[1] ρ])) / (2 kρ1[1] e^(i z kz1[1]));
δ6nljeq[ϕ_, z_, ρ_] := (μ2 ω0)/ρ n e^(i n ϕ) HankelH2[n, kρ1[1] ρ] / (kρ1[1]^2 e^(i kz1[1] z));
δ7nljeq[ϕ_, z_, ρ_] := -(ω0 ε2)/ρ e^(i n ϕ) n HankelH2[n, kρ1[1] ρ] / (kρ1[1]^2 e^(i kz1[1] z));
δ8nljeq[ϕ_, z_, ρ_] := -i e^(i n ϕ)
  (kz1[1] (HankelH2[n - 1, kρ1[1] ρ] - HankelH2[n + 1, kρ1[1] ρ])) / (2 kρ1[1] e^(i z kz1[1]));
```

Regular ("Smooth") Cylinder Equations

Smooth Cylinder Equations TM-Mode

```
an[ϕ_, z_, n1_] := -BesselJ[n1, k2 * ρ2 * Sin[θi]] / HankelH2[n1, k2 * ρ2 * Sin[θi]];
```

Ez

```
EzcylTMS[ϕ_, z_, ρ_] :=
  E0 * Sin[θi] e^(i*k2*z*Cos[θi]) * (Sum[an[ϕ, z, n1] * HankelH2[n1, k2 * ρ * Sin[θi]] * e^(i*n1*ϕ) / i^n1, {n1, nmin, nmax}]);
EzcylTMI[ϕ_, z_, ρ_] :=
  E0 * Sin[θi] * e^(i*k2 z Cos[θi]) * (Sum[BesselJ[n1, k2 * ρ * Sin[θi]] * e^(i*n1*ϕ) / i^n1, {n1, nmin, nmax}]);
EzcylTMSandI[ϕi_, ϕ_, z_, ρ_] := EzcylTMS[ϕ, z, ρ] + EzcylTMI[ϕi, z, ρ];
```

Erho

```
ErhocylTMS[ϕ_, z_, ρ_] := i * E0 * Cos[θi] * e^(i*k2*z*Cos[θi]) * (Sum[an[ϕ, z, n1] *
    ((1/2 (HankelH2[-1 + n1, k2 * ρ * Sin[θi]] - HankelH2[1 + n1, k2 * ρ * Sin[θi]])) * e^(i*n1*ϕ)) / i^n1, {n1, nmin, nmax}]
  );
ErhocylTMI[ϕ_, z_, ρ_] :=
  E0 * Cos[ϕ] * Cos[θi] * e^(i*k2 z Cos[θi]) * (Sum[BesselJ[n1, k2 * ρ * Sin[θi]] * e^(i*n1*ϕ) / i^n1, {n1, nmin, nmax}]);
```

```
ErhocylTMSandI[ϕi_, ϕ_, z_, ρ_] := ErhocylTMS[ϕ, z, ρ] + ErhocylTMI[ϕi, z, ρ];
```

Ephi

```
EphicylTMS[ϕ_, z_, ρ_] := i * E0 * (Cot[θi] / (k2 * ρ)) * e^(i*k2*z*Cos[θi]) *
  ( Sum_{n1=nmin}^{nmax} n1 * i^(-n1+1) * an[ϕ, z, n1] * HankelH2[n1, k2 * ρ * Sin[θi]] * e^(i*n1*ϕ) );

EphicylTMI[ϕ_, z_, ρ_] :=
  -E0 * Sin[ϕ] * Cos[θi] * e^(i*k2 z Cos[θi]) * ( Sum_{n1=nmin}^{nmax} (BesselJ[n1, k2 * ρ * Sin[θi]] * e^(i*n1*ϕ)) / i^n1 );

EphicylTMSandI[ϕi_, ϕ_, z_, ρ_] := EphicylTMS[ϕ, z, ρ] + EphicylTMI[ϕi, z, ρ];
```

Smooth Cylinder Equations TE-Mode

```
bn[ϕ_, z_, n1_] :=
  -(BesselJ[-1 + n1, k2 * ρ2 * Sin[θi]] - BesselJ[1 + n1, k2 * ρ2 * Sin[θi]]) /
   (HankelH2[-1 + n1, k2 * ρ2 * Sin[θi]] - HankelH2[1 + n1, k2 * ρ2 * Sin[θi]]);
  (*-D[BesselJ[n1,k2*ρ2*Sin[θi]],k2*ρ2*Sin[θi]] / D[HankelH2[n1,k2*ρ2*Sin[θi]],k2*ρ2*Sin[θi]];*)
```

Ez

```
EzcylTES[ϕ_, z_, ρ_] := 0;

EzcylTEI[ϕ_, z_, ρ_] := 0;

EzcylTESandI[ϕi_, ϕ_, z_, ρ_] := EzcylTES[ϕ, z, ρ] + EzcylTEI[ϕi, z, ρ];
```

Erho

```
ErhocylTES[ϕ_, z_, ρ_] := -i (H0 / (ω0 ε2 ρ)) * (e^(i k2 z Cos[θi]) / Sin[θi]) *
  ( Sum_{n1=nmin}^{nmax} n1 * i^(-n1+1) * bn[ϕ, z, n1] * HankelH2[n1, k2 * ρ * Sin[θi]] * e^(i n1 ϕ) );

ErhocylTEI[ϕ_, z_, ρ_] := Z2 H0 Sin[ϕ] e^(i k2 z Cos[θi]) Sum_{n1=nmin}^{nmax} (BesselJ[n1, k2 ρ Sin[θi]] e^(i n1 ϕ)) / i^n1;

ErhocylTESandI[ϕi_, ϕ_, z_, ρ_] := ErhocylTES[ϕ, z, ρ] + ErhocylTEI[ϕi, z, ρ];
```


Ephi

```
EphicylTES[ϕ_, z_, ρ_] := i * H0 * Sqrt[μ2 / ε2] * e^(i k2 z Cos[θi]) * Sum_{n1=nmin}^{nmax} (1 / i^n1) bn[ϕ, z, n1] *
  ( (1/2) (HankelH2[-1 + n1, k2 * ρ * Sin[θi]] - HankelH2[1 + n1, k2 * ρ * Sin[θi]]) ) * e^(i n1 ϕ);

EphicylTEI[ϕ_, z_, ρ_] :=
  Z2 * H0 * Cos[ϕ] * e^(i*k2*z*Cos[θi]) * Sum_{n1=nmin}^{nmax} (BesselJ[n1, k2 ρ Sin[θi]] * e^(i n1 ϕ)) / i^n1;

EphicylTESandI[ϕi_, ϕ_, z_, ρ_] := EphicylTES[ϕ, z, ρ] + EphicylTEI[ϕi, z, ρ];
```

Geometric Parameters and Other Inputs

User Inputs - Geometric Parameters

```
aorb = "b"; (* change to "a" or "b"*)

dratio = .1;
apercent = .001;

b = dratio * lambda0;          (*% Thickness of corrugations*)

a = b * apercent;          (*% Distance between corrugations*)

ρ2 = dratio * lambda0;          (*% Max.radius of cylinder (outer radius)*)
ρ1 = ρ2 * .999;          (*% Min.radius of cylinder (inner radius)*)

ρfar = ρ2 * 10;          (*Observation ρ for use in polar plots*)

ϕa = 37 * (Pi / 180);
(* Arbitrarilty assigned value for solving coefficients*)

(*-------------------------------------------------------------------*)
(*An interpretation of Balanis' work in Adv Eng EM Book, chapter 11.5*)
(*has ϕ for scattered and ϕ for incident as separate values. If truly *)
(*considered separate, then ϕi later on is set to ϕivalue. If not, and *)
(*they are considered the same, then ϕi=ϕm in the point matching section *)
(*ϕi=ϕ in the rest of the equations*)
ϕivalue = ϕa;          (* Radial incidence angle*)
(*ϕivalue=37*(Pi/180);*)          (* Radial incidence angle*)
(*ϕivalue=-Pi+.01;*)          (* Radial incidence angle*)
(*-------------------------------------------------------------------*)
```



```
(* Check for max value of 'm' and 'l' the prevent k_rho
 from going imaginary in k_rho_m and k_rho_l respectively *)
mmaxcheck = a * k1 / Pi;
lmaxcheck = (a + b) * k2 * (1 - Cos[θi]) / (2 * Pi);
```

User Inputs - Other Parameters

```
nmax = 3; (*Odd only*)

mmax = 0;

rhosteps = 550;          (*% Qty of steps in range for rho*)

phisteps = 550;          (* Qty of steps in range for phi*)
```

Filenames

```
RhoTableName = StringForm["RhoTable_``_``_``", dratio, apercent * 100, mmax];

PhiTableName = StringForm["PhiTable_``_``_``", dratio, apercent * 100, mmax];

RhoXYPlotsName = StringForm["RhoXYPlots_``_``_``", dratio, apercent * 100, mmax];

PhiPolarPlotsName =
  StringForm["PhiPolarPlots_``_``_``", dratio, apercent * 100, mmax];

PhiRCSPolarPlotsName =
  StringForm["PhiRCSPolarPlots_``_``_``", dratio, apercent * 100, mmax];

PhiRCSdBPolarPlotsName =
  StringForm["PhiRCSdBPolarPlots_``_``_``", dratio, apercent * 100, mmax];

PhiRCSdBXYPlotsName =
  StringForm["PhiRCSdBXYPlots_``_``_``", dratio, apercent * 100, mmax];

RCSdBdataSheetName =
  StringForm["PhiRCSdBXYData_``_``_``", dratio, apercent * 100, mmax];

Rhofilename = StringForm["rho_out``", aorb];

Phifilename = StringForm["phi_out``", aorb];
```


Calculated Values Based on Geometric Parameters and Inputs

n, m, l quantity and Ranges, and Matching Point j

```
mmin = -mmax;
lmin = mmin;
lmax = mmax;
nmin = -nmax;

mp1 = 1; (*overdetermined matching point factor*)

nqty = nmax - nmin + 1;
lqty = lmax - lmin + 1;          (*Also used in place of 'm' in many places*)
jqty = mp1 * nqty * lqty;          (*Number of matching points*)

nrange = N[Range[nmin, nmax, 1] ];
lrange = N[Range[lmin, lmax, 1] ];
mrange = N[Range[mmin, mmax, 1] ];
```

Z Point

```
zpointa = a / 4;
(*z point for plotting and point matching, in boundary 'a'*)
zpointb = (a / 2) + (b / 2);
(*z point for plotting and point matching, in boundary 'b'*)

If[aorb == "a",          (* Selected z point*)
  zpoint = zpointa; ,
  zpoint = zpointb;
 ];
(*zpoints are only used in point matching
 when enabled in the point matching section*)

z = zpoint;
```

Rho Range

```
(*Observation ρ, ρfar, in section 'user inputs - geometric parameters*)

rhomin = ρ1 - (.1 * ρ1);          (*% Min value in coordinate range for rho*)

rhomax = (10 * lambda0) + ρ2;          (*% Max value in coordinate range for rho*)

rhodelta = (rhomax - rhomin) / (rhosteps - 1);

rhorange = N[Range[rhomin, rhomax, rhodelta] ];
```

Phi Value and Range

```
ϕ = ϕa;
phidelta = (2 * Pi) / phisteps;
phirange = N[Range[(0.5 * phidelta), (2 * Pi) - (0.5 * phidelta), phidelta] ] ;
```

Matching Points - z's and ϕ's

```
zsteps = jqty;
jsteps = jqty;

(*3 matching point options*)
(*A.) Changing z's with fixed ϕ*)
(*B.) Changing ϕ's with fixed z*)
(*C.) Changing z's with fixed half ϕ=0 and half ϕ=Pi/2*)
(*D.) Requires overdetermined matrix form,
more j's matching points than unknowns*)
(*    Changing ϕ's with for multiple z
 points based on matching point multiplier, mp *)
(*    D. is not yet implemented  *)
(*-----------------------------------------------------------------------*)
(* Boundary 'b' z matching points*)
(* Boundary 'b' z matching points*)
(* Boundary 'b' z matching points*)
(*zjblength=b/2;*)
(*zjbstep=zjblength/zsteps;*)
(*zjb=N[Range[(a/2)+(0.5*zjbstep),(b/2)+(a/2)-(0.5*zjbstep),zjbstep]];*)

zjbstart = (a / 2) + ((b / zsteps));
zjbstop = ((a / 2) + b) - ((b / zsteps));
zjbstep = (zjbstop - zjbstart) / (zsteps - 1);
zjb = N[Range[zjbstart, zjbstop, zjbstep]] ;

(* updated matching points to negate prior matching points -
 Fixed zpoint matching method for varying phi point matches *)
zjb = ConstantArray[zpointb, jsteps];
(*-----------------------------------------------------------------------*)

(*-----------------------------------------------------------------------*)
(* Boundary 'a' z matching points*)
(* Boundary 'a' z matching points*)
```



```
(* Boundary 'a' z matching points*)
(*zjalength=a/2;*)
(*zjastep=zjalength/zsteps;*)
(*zja=N[Range[(0.5*zjastep),(a/2)-(0.5*zjastep),zjastep] ] ; *)

zjastart = (-a / 2) + ((a / zsteps));
zjastop = (+a / 2) - ((a / zsteps));
zjastep = (zjastop - zjastart) / (zsteps - 1);
zja = N[Range[zjastart, zjastop, zjastep]];

(* updated matching points to negate prior matching points -
 Fixed zpoint matching method for varying phi point matches *)
zja = ConstantArray[zpointa, jsteps];

(*-----------------------------------------------------------------------*)

(*-----------------------------------------------------------------------*)
(* ϕ matching points, 50% ϕm=0 and 50% ϕm=Pi/2*)
ϕm = ConstantArray[0, jsteps];
count = 1;
For[count = 1, count ≤ zsteps, count++,
  If[count < .5 * zsteps,
    ϕm[[count]] = 0;,
    ϕm[[count]] = Pi / 2;
   ];
 ];
(* updated matching points to negate prior matching points*)
(*ϕm=ConstantArray[ϕa,jsteps]; *)

(* updated matching points to negate prior matching points*)
(*matching points of 0 to 2Pi*)
ϕstart = (0) + ((.5 * 2 * Pi / jsteps));
ϕstop = (2 * Pi) - ((.5 * 2 * Pi / jsteps));
ϕstep = (ϕstop - ϕstart) / (jsteps - 1);
ϕm = N[Range[ϕstart, ϕstop, ϕstep] ];

(*ϕm=ConstantArray[ϕa,jsteps];*) (*for z matching points*)

(*-----------------------------------------------------------------------*)
```

Numerical Computation of Coefficients and Fields

Setting Empty Matrices to Solve for Coefficients and Field Components

```
(*-------------------------------------------------------------------------------*)
(*----------------------     EMPTY MATRICES FOR ALL EQUATIONS     ---------------------*)
(*-------------------------------------------------------------------------------*)
(*Empty Matrices for Eq1 set, Boundary 'b'*)
Cnlbprematrix = Array[Cnlb, {nqty, lqty}];
anljbprematrix = Array[anljb, {nqty, lqty, jqty}];
anljbarray = ConstantArray[0, jqty];
bnjbprematrix = Array[bnjb, {jqty}];

(*Empty Matrices for Eq2 set, Boundary 'a'*)
Cnlaprematrix = Array[Cnla, {nqty, lqty}];
anljaprematrix = Array[anlja, {nqty, lqty, jqty}];
anljaarray = ConstantArray[0, jqty];
bnjaprematrix = Array[bnja, {jqty}];
Anmprematrix = Array[Anm, {nqty, lqty}];
hnmjprematrix = Array[hnmj, {nqty, lqty, jqty}];
hnmjarray = ConstantArray[0, jqty];

(*Empty Matrices for Eq3 set, Boundary 'b'*)
Dnlbprematrix = Array[Dnlb, {nqty, lqty}];        (*Cnlb already defined*)
cnljbprematrix = Array[cnljb, {nqty, lqty, jqty}];
cnljbarray = ConstantArray[0, jqty];
enljbprematrix = Array[enljb, {nqty, lqty, jqty}];
enljbarray = ConstantArray[0, jqty];
dnjbprematrix = Array[dnjb, {jqty}];

(*Empty Matrices for Eq4 set, Boundary 'a'*)
Dnlaprematrix = Array[Dnla, {nqty, lqty}];
(*Cnla and Anm already defined*)
cnljaprematrix = Array[cnlja, {nqty, lqty, jqty}];
cnljaarray = ConstantArray[0, jqty];
enljaprematrix = Array[enlja, {nqty, lqty, jqty}];
enljaarray = ConstantArray[0, jqty];
dnjaprematrix = Array[dnja, {jqty}];
knmjprematrix = Array[knmj, {nqty, lqty, jqty}];
knmjarray = ConstantArray[0, jqty];
lnmjprematrix = Array[knmj, {nqty, lqty, jqty}];
```



```
lnmjarray = ConstantArray[0, jqty];

(*Empty Matrices for Eq5 set, Boundary 'b'*)
fnjbprematrix = Array[fnjb, {jqty}];
gnljbprematrix = Array[gnljb, {nqty, lqty, jqty}];
gnljbarray = ConstantArray[0, jqty];
Fnlprematrix = Array[Fnl, {nqty, lqty}];
δ2nljprematrix = Array[δ2nlj, {nqty, lqty, jqty}];
δ2nljarray = ConstantArray[0, jqty];

(*Empty Matrices for Eq6 set, Boundary 'a'*)
fnjaprematrix = Array[fnja, {jqty}];
gnljaprematrix = Array[gnlja, {nqty, lqty, jqty}];
gnljaarray = ConstantArray[0, jqty];
Bnmprematrix = Array[Bnm, {nqty, lqty}];
inmjprematrix = Array[inmj, {nqty, lqty, jqty}];
inmjarray = ConstantArray[0, jqty];

(*Empty Matrices for Eq7 set, Boundary 'b'*)
mnjbprematrix = Array[mnjb, {jqty}];
nnljbprematrix = Array[nnljb, {nqty, lqty, jqty}];
nnljbarray = ConstantArray[0, jqty];
qnljbprematrix = Array[qnljb, {nqty, lqty, jqty}];
qnljbarray = ConstantArray[0, jqty];
Enlprematrix = Array[Enl, {nqty, lqty}];
δ3nljprematrix = Array[δ3nlj, {nqty, lqty, jqty}];
δ3nljarray = ConstantArray[0, jqty];
δ4nljprematrix = Array[δ4nlj, {nqty, lqty, jqty}];
δ4nljarray = ConstantArray[0, jqty];

(*Empty Matrices for Eq8 set, Boundary 'a'*)
mnjaprematrix = Array[mnja, {jqty}];
nnljaprematrix = Array[nnlja, {nqty, lqty, jqty}];
nnljaarray = ConstantArray[0, jqty];
qnljaprematrix = Array[qnlja, {nqty, lqty, jqty}];
qnljaarray = ConstantArray[0, jqty];
onmjprematrix = Array[onmj, {nqty, lqty, jqty}];
onmjarray = ConstantArray[0, jqty];
pnmjprematrix = Array[pnmj, {nqty, lqty, jqty}];
pnmjarray = ConstantArray[0, jqty];
```

Numerically Solving for Field Components

```
In[44960]:= (*Boundary Conditions p=p2,
  a/2≤z≤a/2+b (For Boundary'b') and-a/2≤z≤a/2 (For Boundary'a')*)

In[44961]:= ncount = 1;
  lcount = 1;
  jcount = 1;

In[44964]:= For[jcount = 1, jcount ≤ jqty, jcount++,
    ϕ = ϕm[[jcount]]; (*ϕ matching points*)
    (*ϕ=ϕa*); (*ϕ fixed for z point matching*)
    ϕi = ϕ; (*ϕ matching points*)
    (*ϕi=ϕivalue;*)
    (*Only if considered unique and distinct from scattered field*)

    (*Region II Equations, Incident Field Components, Boundary 'b'*)
    bnjb[jcount] = bnjeq[ϕi, zjb[[jcount]], ρ2];
    fnjb[jcount] = fnjeq[ϕi, zjb[[jcount]], ρ2];
    dnjb[jcount] = dnjeq[ϕi, zjb[[jcount]], ρ2];
    mnjb[jcount] = mnjeq[ϕi, zjb[[jcount]], ρ2];

    (*Region II Equations, Incident Field Components, Boundary 'a'*)
    bnja[jcount] = bnjeq[ϕi, zja[[jcount]], ρ2];
    fnja[jcount] = fnjeq[ϕi, zja[[jcount]], ρ2];
    dnja[jcount] = dnjeq[ϕi, zja[[jcount]], ρ2];
    mnja[jcount] = mnjeq[ϕi, zja[[jcount]], ρ2];

    For[ncount = 1, ncount ≤ nqty, ncount++,
     n = nrange[[ncount]];

     For[lcount = 1, lcount ≤ lqty, lcount++,
      l = lrange[[lcount]];
      m = lrange[[lcount]];
      (*Region II Equations, Scattered Field Components, Boundary 'b'*)
      anljb[ncount, lcount, jcount] = anljeq[ϕ, zjb[[jcount]], ρ2];
      gnljb[ncount, lcount, jcount] = gnljeq[ϕ, zjb[[jcount]], ρ2];
      cnljb[ncount, lcount, jcount] = cnljeq[ϕ, zjb[[jcount]], ρ2];
      enljb[ncount, lcount, jcount] = enljeq[ϕ, zjb[[jcount]], ρ2];
      nnljb[ncount, lcount, jcount] = nnljeq[ϕ, zjb[[jcount]], ρ2];
      qnljb[ncount, lcount, jcount] = qnljeq[ϕ, zjb[[jcount]], ρ2];

      (*Boundary 'b'Only*)
      δ2nlj[ncount, lcount, jcount] =
       δ2nljeq[ϕ, zjb[[jcount]], ρ2]; (* boundary 'b'*)
      δ3nlj[ncount, lcount, jcount] = δ3nljeq[ϕ, zjb[[jcount]], ρ2];
```



```
      (* boundary 'b'*)
      δ4nlj[ncount, lcount, jcount] = δ4nljeq[ϕ, zjb[[jcount]], ρ2];
      (* boundary 'b'*)

      (*Region II Equations, Scattered Field Components, Boundary 'a'*)
      anlja[ncount, lcount, jcount] = anljeq[ϕ, zja[[jcount]], ρ2];
      gnlja[ncount, lcount, jcount] = gnljeq[ϕ, zja[[jcount]], ρ2];
      cnlja[ncount, lcount, jcount] = cnljeq[ϕ, zja[[jcount]], ρ2];
      enlja[ncount, lcount, jcount] = enljeq[ϕ, zja[[jcount]], ρ2];
      nnlja[ncount, lcount, jcount] = nnljeq[ϕ, zja[[jcount]], ρ2];
      qnlja[ncount, lcount, jcount] = qnljeq[ϕ, zja[[jcount]], ρ2];

      (*Region I Equations, Boundary 'a'*)
      hnmj[ncount, lcount, jcount] =
       hnmjeq[ϕ, zja[[jcount]], ρ2]; (* boundary 'a'*)
      inmj[ncount, lcount, jcount] = inmjeq[ϕ, zja[[jcount]], ρ2]; (* boundary 'a'*)
      knmj[ncount, lcount, jcount] = knmjeq[ϕ, zja[[jcount]], ρ2]; (* boundary 'a'*)
      lnmj[ncount, lcount, jcount] = lnmjeq[ϕ, zja[[jcount]], ρ2]; (* boundary 'a'*)
      onmj[ncount, lcount, jcount] = onmjeq[ϕ, zja[[jcount]], ρ2]; (* boundary 'a'*)
      pnmj[ncount, lcount, jcount] = pnmjeq[ϕ, zja[[jcount]], ρ2]; (* boundary 'a'*)
     ];
    ];
   ];
```



```
In[44965]:= For[jcount = 1, jcount ≤ jqty, jcount++,
    anljbarray[[jcount]] = Flatten[Part[anljbprematrix, All, All, jcount]];
    anljbarray[[jcount]] = Flatten[Part[anljbprematrix, All, All, jcount]];
    cnljbarray[[jcount]] = Flatten[Part[cnljbprematrix, All, All, jcount]];
    enljbarray[[jcount]] = Flatten[Part[enljbprematrix, All, All, jcount]];
    gnljbarray[[jcount]] = Flatten[Part[gnljbprematrix, All, All, jcount]];
    nnljbarray[[jcount]] = Flatten[Part[nnljbprematrix, All, All, jcount]];
    qnljbarray[[jcount]] = Flatten[Part[qnljbprematrix, All, All, jcount]];
    anljaarray[[jcount]] = Flatten[Part[anljaprematrix, All, All, jcount]];
    cnljaarray[[jcount]] = Flatten[Part[cnljaprematrix, All, All, jcount]];
    enljaarray[[jcount]] = Flatten[Part[enljaprematrix, All, All, jcount]];
    gnljaarray[[jcount]] = Flatten[Part[gnljaprematrix, All, All, jcount]];
    nnljaarray[[jcount]] = Flatten[Part[nnljaprematrix, All, All, jcount]];
    qnljaarray[[jcount]] = Flatten[Part[qnljaprematrix, All, All, jcount]];
    δ2nljarray[[jcount]] = Flatten[Part[δ2nljprematrix, All, All, jcount]];
    δ3nljarray[[jcount]] = Flatten[Part[δ3nljprematrix, All, All, jcount]];
    δ4nljarray[[jcount]] = Flatten[Part[δ4nljprematrix, All, All, jcount]];
    hnmjarray[[jcount]] = Flatten[Part[hnmjprematrix, All, All, jcount]];
    knmjarray[[jcount]] = Flatten[Part[knmjprematrix, All, All, jcount]];
    lnmjarray[[jcount]] = Flatten[Part[lnmjprematrix, All, All, jcount]];
    onmjarray[[jcount]] = Flatten[Part[onmjprematrix, All, All, jcount]];
    pnmjarray[[jcount]] = Flatten[Part[pnmjprematrix, All, All, jcount]];
    inmjarray[[jcount]] = Flatten[Part[inmjprematrix, All, All, jcount]];
   ];
```

Solve for Coefficients Cnlb, Dnlb, Fnl, and Enl

```
In[44966]:= Cnlb = Flatten[LeastSquares[anljbarray, -bnjbprematrix]];

In[44967]:= Dnlb = Flatten[LeastSquares[enljbarray, -(dnjbprematrix + cnljbarray.Cnlb)]];

In[44968]:= Fnl = Flatten[LeastSquares[δ2nljarray, fnjbprematrix + gnljbarray.Dnlb]];

In[44969]:= Enl = Flatten[LeastSquares[δ3nljarray,
      (mnjbprematrix + nnljbarray.Cnlb + qnljbarray.Dnlb) - δ4nljarray.Fnl]];
```

Solve for Coefficients Cnla, Dnla, Anm, and Bnm - Loop Method

```
In[44970]:= Anmtemp = FullSimplify[Flatten[PseudoInverse[hnmjarray].
      (bnjaprematrix + (anljaarray.Flatten[Cnlaprematrix]))]];

In[44971]:= Bnmtemp = FullSimplify[Flatten[PseudoInverse[inmjarray].
      (fnjaprematrix + (gnljaarray.Flatten[Dnlaprematrix]))]];
```



```
In[44972]:= Cnlatemp = FullSimplify[
    Flatten[PseudoInverse[cnljaarray].(((knmjarray.Anmtemp) + (lnmjarray.Bnmtemp)) -
       dnjaprematrix - (enljaarray.Flatten[Dnlaprematrix]))]];

In[44973]:= Cnlatemp2 = Solve[Flatten[FullSimplify[Reduce[Cnlatemp == Flatten[Cnlaprematrix],
      Flatten[Cnlaprematrix]]]], Flatten[Cnlaprematrix]];

In[44974]:= Anmtemp2 = Flatten[FullSimplify[Anmtemp /. Cnlatemp2]];

In[44975]:= Cnlatem3 = Flatten[FullSimplify[Cnlatemp /. Cnlatemp2]];

In[44976]:= Dnlatemp = FullSimplify[Flatten[
     PseudoInverse[qnljaarray].(((onmjarray.Anmtemp2) + (pnmjarray.Bnmtemp)) -
        mnjaprematrix - (nnljaarray.Cnlatem3))]];

In[44977]:= ncount = 1;
  lcount = 1;
  Dnlatemp2 = ConstantArray[0, jqty];
  Dnlasol = ConstantArray[0, jqty];
  For[ncount = 1, ncount ≤ nqty, ncount++,
    For[lcount = 1, lcount ≤ lqty, lcount++,
     index = ((ncount - 1) * (lqty)) + lcount;
     If[index == 1,
      (*TRUE*)
      Dnlatemp2[[index]] = Solve[Dnlatemp[[index]] ==
         Flatten[Dnlaprematrix][[index]], Flatten[Dnlaprematrix][[index]]];
      Dnlaeq = Flatten[Dnlatemp2[[index]]];,
      (*FALSE*)
      Dnlatemp = Dnlatemp /. Dnlaeq;
      Dnlatemp2[[index]] =
       Solve[(Dnlatemp[[index]] == Flatten[Dnlaprematrix][[index]]) /. Dnlaeq,
        Flatten[Dnlaprematrix][[index]]];
      Dnlaeq = Flatten[Append[Flatten[Dnlaeq], Dnlatemp2[[index]]]];
     ];
    ];
   ];
```

```
In[44982]:= For[ncount = nqty, ncount ≥ 1, ncount--,
   For[lcount = lqty, lcount ≥ 1, lcount--,
    index = ((ncount - 1) * (lqty)) + lcount;
    If[index == jqty,
     (*TRUE*)
     Dnlasol[[index]] = Flatten[Dnlatemp2[[index]]];
     Dnlaeq2 = Flatten[Dnlasol[[index]]];
     ,
     (*FALSE*)
     Dnlasol[[index]] = Flatten[Dnlatemp2[[index]] /. Dnlaeq2];
     Dnlaeq2 = Flatten[Append[Flatten[Dnlaeq2], Dnlasol[[index]]]];
    ];
   ];
  ];

In[44983]:= Dnlasol = Flatten[Dnlasol];

In[44984]:= Dnla = Dnlasol[[All, 2]];

In[44985]:= Cnla = FullSimplify[Cnlatem3 /. Dnlasol];

In[44986]:= Anm = FullSimplify[Anmtemp2 /. Dnlasol];

In[44987]:= Bnm = FullSimplify[Bnmtemp /. Dnlasol];
```

Solve for Coefficients Cnla, Dnla, Anm, and Bnm - NCAlgebra Method - NOT USED

```
In[44991]:= (* NCSolve Only Method *)

In[44992]:= (*Dnla =-PseudoInverse[
     qnljaarray-pnmjarray.PseudoInverse[inmjarray].gnljaarray].mnjaprematrix+
   PseudoInverse[qnljaarray-pnmjarray.PseudoInverse[inmjarray].gnljaarray].
    onmjarray.PseudoInverse[hnmjarray].bnjaprematrix+
   PseudoInverse[qnljaarray-pnmjarray.PseudoInverse[inmjarray].gnljaarray].
    pnmjarray.PseudoInverse[inmjarray].fnjaprematrix; *)

In[44993]:= (*Cnla=-PseudoInverse[cnljaarray-knmjarray.PseudoInverse[hnmjarray].anljaarray].
     dnjaprematrix-PseudoInverse[
     cnljaarray-knmjarray.PseudoInverse[hnmjarray].anljaarray].enljaarray.Dnla+
   PseudoInverse[cnljaarray-knmjarray.PseudoInverse[hnmjarray].anljaarray].
    knmjarray.PseudoInverse[hnmjarray].bnjaprematrix+
   PseudoInverse[cnljaarray-knmjarray.PseudoInverse[hnmjarray].anljaarray].
    lnmjarray.PseudoInverse[inmjarray].fnjaprematrix+
   PseudoInverse[cnljaarray-knmjarray.PseudoInverse[hnmjarray].anljaarray].
    lnmjarray.PseudoInverse[inmjarray].gnljaarray.Dnla; *)
```



```
In[44994]:= (*Anm= LeastSquares[hnmjarray, (bnjaprematrix+anljaarray.Cnla)]; *)

In[44995]:= (*Bnm=LeastSquares[inmjarray, (fnjaprematrix+gnljaarray.Dnla)]; *)
```

Empty Arrays for Storing Field Solutions - Changing Rho

Corrugated Cylinder Empty Arrays

Scattered + Incident Arrays

```
In[44996]:= Eztemp = ConstantArray[0, Length[rhorange]];
  Ephitemp = ConstantArray[0, Length[rhorange]];
  Erhotemp = ConstantArray[0, Length[rhorange]];
```

Scattered Only Arrays

```
In[44999]:= EztempS = ConstantArray[0, Length[rhorange]];
  EphitempS = ConstantArray[0, Length[rhorange]];
  ErhotempS = ConstantArray[0, Length[rhorange]];
```

Incident Only Arrays

```
In[45002]:= EztempI = ConstantArray[0, Length[rhorange]];
  EphitempI = ConstantArray[0, Length[rhorange]];
  ErhotempI = ConstantArray[0, Length[rhorange]];
```

Regular ("Smooth") Cylinder Empty Arrays

Scattered + Incident Arrays (TM and TE)

```
In[45005]:= EzregcylTMSandI = ConstantArray[0, Length[rhorange]];
  ErhoregcylTMSandI = ConstantArray[0, Length[rhorange]];
  EphiregcylTMSandI = ConstantArray[0, Length[rhorange]];
  EzregcylTESandI = ConstantArray[0, Length[rhorange]];
  ErhoregcylTESandI = ConstantArray[0, Length[rhorange]];
  EphiregcylTESandI = ConstantArray[0, Length[rhorange]];
```


Scattered Only Arrays (TM and TE)

```
In[45011]:= EzregcylTMS = ConstantArray[0, Length[rhorange]];
  ErhoregcylTMS = ConstantArray[0, Length[rhorange]];
  EphiregcylTMS = ConstantArray[0, Length[rhorange]];
  EzregcylTES = ConstantArray[0, Length[rhorange]];
  ErhoregcylTES = ConstantArray[0, Length[rhorange]];
  EphiregcylTES = ConstantArray[0, Length[rhorange]];
```

Incident Only Arrays (TM and TE)

```
In[45017]:= EzregcylTMI = ConstantArray[0, Length[rhorange]];
  ErhoregcylTMI = ConstantArray[0, Length[rhorange]];
  EphiregcylTMI = ConstantArray[0, Length[rhorange]];
  EzregcylTEI = ConstantArray[0, Length[rhorange]];
  ErhoregcylTEI = ConstantArray[0, Length[rhorange]];
  EphiregcylTEI = ConstantArray[0, Length[rhorange]];
```

Changing Rho Do Loop Calculation

Do Loop

```
In[45023]:= count = 1;

  ncount = 1;
  lcount = 1;
  jcount = 1;

  Do[
   ϕ = ϕa; (*scattered field ϕ, possibly incident ϕ as well*)
   (*ϕi=ϕ;*)
   ϕi = ϕivalue; (*Only if considered unique and distinct from scattered field*)

   (*Region II temp variables, Scattered Field Component*)
   EIIzstemp = 0;
   EIIphistemp = 0;
   HIIzstemp = 0;
   HIIphistemp = 0;
   EIIrhostemp = 0;
   HIIrhostemp = 0;
```



```
   (*Region II temp variables, Total Field*)
   EIIztemp = 0;
   EIIphitemp = 0;
   HIIztemp = 0;
   HIIphitemp = 0;
   EIIrhotemp = 0;
   HIIrhotemp = 0;

   (*Region I temp variables*)
   EIztemp = 0;
   EIphitemp = 0;
   HIztemp = 0;
   HIphitemp = 0;
   EIrhotemp = 0;
   HIrhotemp = 0;

   (*Smooth Cylinder*)
   If[ρ < ρ2,
    (*TRUE*)
    EzregcylTMSandI[[count]] = 0;
    EzregcylTMS[[count]] = 0;
    ErhoregcylTMSandI[[count]] = 0;
    ErhoregcylTMS[[count]] = 0;
    EphiregcylTMSandI[[count]] = 0;
    EphiregcylTMS[[count]] = 0;
    EzregcylTESandI[[count]] = 0;
    EzregcylTES[[count]] = 0;
    ErhoregcylTESandI[[count]] = 0;
    ErhoregcylTES[[count]] = 0;
    EphiregcylTESandI[[count]] = 0;
    EphiregcylTES[[count]] = 0;
    ,
    (*FALSE*)
    EzregcylTMSandI[[count]] = EzcylTMSandI[ϕi, ϕ, z, ρ];
    EzregcylTMS[[count]] = EzcylTMS[ϕ, z, ρ];
    ErhoregcylTMSandI[[count]] = ErhocylTMSandI[ϕi, ϕ, z, ρ];
    ErhoregcylTMS[[count]] = ErhocylTMS[ϕ, z, ρ];
    EphiregcylTMSandI[[count]] = EphicylTMSandI[ϕi, ϕ, z, ρ];
    EphiregcylTMS[[count]] = EphicylTMS[ϕ, z, ρ];
    EzregcylTESandI[[count]] = EzcylTESandI[ϕi, ϕ, z, ρ];
    EzregcylTES[[count]] = EzcylTES[ϕ, z, ρ];
    ErhoregcylTESandI[[count]] = ErhocylTESandI[ϕi, ϕ, z, ρ];
```

```
ErhoregcylTES[[count]] = ErhocylTES[ϕ, z, ρ];
EphiregcylTESandI[[count]] = EphicylTESandI[ϕi, ϕ, z, ρ];
EphiregcylTES[[count]] = EphicylTES[ϕ, z, ρ];

];

If[ρ ≥ ρ1, (*Check to see if inside conductor or not*)
 If[-a/2 ≤ z ≤ a/2, (*Check to see if in boundary 'a' or boundary 'b'*)
   (*--------------IF TRUE--------------*)
   (* In boundary 'a' but still need to determine if in Region I or II*)
   boundarycheck = "Boundary a";
   If[ρ < ρ2, (*Region I fields, boundary 'a'*)
    (*Region I fields, boundary 'a'*)
    For[ncount = 1, ncount ≤ nqty, ncount++,
      n = nrange[[ncount]];
      For[lcount = 1, lcount ≤ lqty, lcount++,
       l = lrange[[lcount]];
       m = lrange[[lcount]];
       index = ((ncount - 1) * (lqty)) + lcount;
       EIztemp = EIztemp + (hnmjeq[ϕ, z, ρ] * Anm[[index]]);
       EIphitemp =
        EIphitemp + (knmjeq[ϕ, z, ρ] * Anm[[index]] + lnmjeq[ϕ, z, ρ] * Bnm[[index]]);
       HIztemp = HIztemp + (inmjeq[ϕ, z, ρ] * Bnm[[index]]);
       HIphitemp =
        HIphitemp + (onmjeq[ϕ, z, ρ] * Anm[[index]] + pnmjeq[ϕ, z, ρ] * Bnm[[index]]);
       EIrhotemp = EIrhotemp + (rnmjeq[ϕ, z, ρ] * Anm[[index]] +
           snmjeq[ϕ, z, ρ] * Bnm[[index]]);
       HIrhotemp = HIrhotemp + (tnmjeq[ϕ, z, ρ] * Anm[[index]] +
           unmjeq[ϕ, z, ρ] * Bnm[[index]]);
      ];
     ];
    ,
    (*Region II fields, boundary 'a'*)
    For[ncount = 1, ncount ≤ nqty, ncount++,
     n = nrange[[ncount]];
     For[lcount = 1, lcount ≤ lqty, lcount++,
      l = lrange[[lcount]];
      m = lrange[[lcount]];
      index = ((ncount - 1) * (lqty)) + lcount;
      (*Scattered Field Portion First*)
      EIIzstemp = EIIzstemp + anljeq[ϕ, z, ρ] * Cnla[[index]];
      EIIphistemp = EIIphistemp +
        (cnljeq[ϕ, z, ρ] * Cnla[[index]] + enljeq[ϕ, z, ρ] * Dnla[[index]]);
```



```
      HIIzstemp = HIIzstemp + gnljeq[ϕ, z, ρ] * Dnla[[index]];
      HIIphistemp = HIIphistemp +
        (nnljeq[ϕ, z, ρ] * Cnla[[index]] + qnljeq[ϕ, z, ρ] * Dnla[[index]]);
      EIIrhostemp = EIIrhostemp + (vnljeq[ϕ, z, ρ] * Cnla[[index]] +
          wnljeq[ϕ, z, ρ] * Dnla[[index]]);
      HIIrhostemp = HIIrhostemp + (xnljeq[ϕ, z, ρ] * Cnla[[index]] +
          ynljeq[ϕ, z, ρ] * Dnla[[index]]);
     ];
    ];
    (*Add Incident and Scattered Field*)
    EIIztemp = bnjeq[ϕi, z, ρ] + EIIzstemp;
    EIIphitemp = dnjeq[ϕi, z, ρ] + EIIphistemp;
    HIIztemp = fnjeq[ϕi, z, ρ] + HIIzstemp;
    HIIphitemp = mnjeq[ϕi, z, ρ] + HIIphistemp;
    EIIrhotemp = γnjeq[ϕi, z, ρ] + EIIrhostemp;
    HIIrhotemp = δnjeq[ϕi, z, ρ] + HIIrhostemp;

    (*Scattered Field Only*)
    EIIzstemp = EIIzstemp;
    EIIphistemp = EIIphistemp;
    EIIrhostemp = EIIrhostemp;

   ];
   ,
   (*--------------IF FALSE--------------*)
   (* In boundary 'b' but still need to determine if in Cylinder or Region II*)
   boundarycheck = "Boundary b";
   If[ρ < ρ2, (*In Cylinder, boundary 'b'*)
    Eztemp[[count]] = Eztemp[[count]] + 0;
    Ephitemp[[count]] = Ephitemp[[count]] + 0;
    Erhotemp[[count]] = Erhotemp[[count]] + 0;

    EztempS[[count]] = EztempS[[count]] + 0;
    EphitempS[[count]] = EphitempS[[count]] + 0;
    ErhotempS[[count]] = ErhotempS[[count]] + 0;

    Hztemp[[count]] = Hztemp[[count]] + 0;
    Hphitemp[[count]] = Hphitemp[[count]] + 0;
    Hrhotemp[[count]] = Hrhotemp[[count]] + 0;,
    (*Region II fields, boundary 'b'*)
    For[ncount = 1, ncount ≤ nqty, ncount++,
     n = nrange[[ncount]];
     For[lcount = 1, lcount ≤ lqty, lcount++,
```



```
      l = lrange[[lcount]];
      m = lrange[[lcount]];
      index = ((ncount - 1) * (lqty)) + lcount;
      (*Scattered Field Portion First*)
      EIIzstemp = EIIzstemp + anljeq[ϕ, z, ρ] * Cnlb[[index]];
      EIIphistemp = EIIphistemp +
        (cnljeq[ϕ, z, ρ] * Cnlb[[index]] + enljeq[ϕ, z, ρ] * Dnlb[[index]]);
      HIIzstemp = HIIzstemp + gnljeq[ϕ, z, ρ] * Dnlb[[index]];
      HIIphistemp = HIIphistemp +
        (nnljeq[ϕ, z, ρ] * Cnlb[[index]] + qnljeq[ϕ, z, ρ] * Dnlb[[index]]);
      EIIrhostemp = EIIrhostemp + (vnljeq[ϕ, z, ρ] * Cnlb[[index]] +
          wnljeq[ϕ, z, ρ] * Dnlb[[index]]);
      HIIrhostemp = HIIrhostemp + (xnljeq[ϕ, z, ρ] * Cnlb[[index]] +
          ynljeq[ϕ, z, ρ] * Dnlb[[index]]);
     ];
    ];
    (*Add Incident and Scattered Field*)

    EIIztemp = bnjeq[ϕi, z, ρ] + EIIzstemp;
    EIIphitemp = dnjeq[ϕi, z, ρ] + EIIphistemp;
    HIIztemp = fnjeq[ϕi, z, ρ] + HIIzstemp;
    HIIphitemp = mnjeq[ϕi, z, ρ] + HIIphistemp;
    EIIrhotemp = γnjeq[ϕi, z, ρ] + EIIrhostemp;
    HIIrhotemp = δnjeq[ϕi, z, ρ] + HIIrhostemp;

    (*Scattered Field Only*)
    EIIzstemp = EIIzstemp;
    EIIphistemp = EIIphistemp;
    EIIrhostemp = EIIrhostemp;

   ];
  ] (*END IF STATEMENT*);,
 Eztemp[[count]] = Eztemp[[count]] + 0;
 Ephitemp[[count]] = Ephitemp[[count]] + 0;
 Erhotemp[[count]] = Erhotemp[[count]] + 0;

 EztempS[[count]] = EztempS[[count]] + 0;
 EphitempS[[count]] = EphitempS[[count]] + 0;
 ErhotempS[[count]] = ErhotempS[[count]] + 0;

 Hztemp[[count]] = Hztemp[[count]] + 0;
 Hphitemp[[count]] = Hphitemp[[count]] + 0;
 Hrhotemp[[count]] = Hrhotemp[[count]] + 0;
```



```
  (*All E-fields zeros*)
 ];

 Eztemp[[count]] = EIztemp + EIIztemp;
 Ephitemp[[count]] = EIphitemp + EIIphitemp;
 Erhotemp[[count]] = EIrhotemp + EIIrhotemp;

 EztempS[[count]] = EztempS[[count]] + EIIzstemp;
 EphitempS[[count]] = EphitempS[[count]] + EIIphistemp;
 ErhotempS[[count]] = ErhotempS[[count]] + EIIrhostemp;

 Hztemp[[count]] = HIztemp + HIIztemp;
 Hphitemp[[count]] = HIphitemp + HIIphitemp;
 Hrhotemp[[count]] = HIrhotemp + HIIrhotemp;

 count = count + 1;
 , {ρ, rhomin, rhomax, rhodelta}];
(*Close of 'Do loop'*)
```

Solution for Plots - Corrugated Cylinder

In[45028]:=

```
ϕ = ϕa; (*scattered field ϕ, possibly incident ϕ as well*)
(*ϕi=ϕ;*)
ϕi = ϕivalue; (*Only if considered unique and distinct from scattered field*)
```

```
In[45030]:= (*Incident + Scattered Solution*)
SolEz = N[Abs[Eztemp]];
dataEz = Transpose[{rhorange / lambda0, SolEz}];

SolEphi = N[Abs[Ephitemp]];
dataEphi = Transpose[{rhorange / lambda0, SolEphi}];

SolErho = N[Abs[Erhotemp]];
dataErho = Transpose[{rhorange / lambda0, SolErho}];

SolEAllSandI = N[Abs[Sqrt[((Erhotemp) * Cos[ϕ] - (Ephitemp) * Sin[ϕ]) ^2 +
      ((Erhotemp) * Sin[ϕ] + (Ephitemp) * Cos[ϕ]) ^2 + (Eztemp) ^2]]];
dataEAllSandI = Transpose[{rhorange / lambda0, SolEAllSandI}];

(*Scattered Only Solution*)
SolEzS = N[Abs[EztempS]];
dataEzS = Transpose[{rhorange / lambda0, SolEzS}];

SolEphiS = N[Abs[EphitempS]];
dataEphiS = Transpose[{rhorange / lambda0, SolEphiS}];

SolErhoS = N[Abs[ErhotempS]];
dataErhoS = Transpose[{rhorange / lambda0, SolErhoS}];

SolEAllS = N[Abs[Sqrt[(ErhotempS * Cos[ϕ] - EphitempS * Sin[ϕ]) ^2 +
      (ErhotempS * Sin[ϕ] + EphitempS * Cos[ϕ]) ^2 + (EztempS) ^2]]];
dataEAllS = Transpose[{rhorange / lambda0, SolEAllS}];
```

Solution for Plots - Regular "Smooth" Cylinder

```
In[45046]:=
ϕ = ϕa; (*scattered field ϕ, possibly incident ϕ as well*)
(*ϕi=ϕ;*)
ϕi = ϕivalue; (*Only if considered unique and distinct from scattered field*)
```


TM Mode Solutions

```
In[45048]:= (*Incident + Scattered Solution*)
SolEzregcylTMSandI = N[Abs[EzregcylTMSandI]];
dataEzregcylTMSandI = Transpose[{rhorange / lambda0, SolEzregcylTMSandI}];

SolErhoregcylTMSandI = N[Abs[ErhoregcylTMSandI]];
dataErhoregcylTMSandI = Transpose[{rhorange / lambda0, SolErhoregcylTMSandI}];

SolEphiregcylTMSandI = N[Abs[EphiregcylTMSandI]];
dataEphiregcylTMSandI = Transpose[{rhorange / lambda0, SolEphiregcylTMSandI}];

SolEAllTMSandI =
  N[Abs[Sqrt[((ErhoregcylTMSandI) * Cos[ϕ] - (EphiregcylTMSandI) * Sin[ϕ]) ^2 +
      ((ErhoregcylTMSandI) * Sin[ϕ] + (EphiregcylTMSandI) * Cos[ϕ]) ^2 +
      (EzregcylTMSandI) ^2]]];
dataEAllTMSandI = Transpose[{rhorange / lambda0, SolEAllTMSandI}];

(*Scattered Only Solution*)
SolEzregcylTMS = N[Abs[EzregcylTMS]];
dataEzregcylTMS = Transpose[{rhorange / lambda0, SolEzregcylTMS}];

SolErhoregcylTMS = N[Abs[ErhoregcylTMS]];
dataErhoregcylTMS = Transpose[{rhorange / lambda0, SolErhoregcylTMS}];

SolEphiregcylTMS = N[Abs[EphiregcylTMS]];
dataEphiregcylTMS = Transpose[{rhorange / lambda0, SolEphiregcylTMS}];

SolEAllTMS = N[Abs[Sqrt[(ErhoregcylTMS * Cos[ϕ] - EphiregcylTMS * Sin[ϕ]) ^2 +
      (ErhoregcylTMS * Sin[ϕ] + EphiregcylTMS * Cos[ϕ]) ^2 + (EzregcylTMS) ^2]]];
dataEAllTMS = Transpose[{rhorange / lambda0, SolEAllTMS}];
```


TE Mode Solutions

```
In[45064]:= (*Incident + Scattered Solution*)
SolEzregcylTESandI = N[Abs[EzregcylTESandI]];
dataEzregcylTESandI = Transpose[{rhorange / lambda0, SolEzregcylTESandI}];

SolErhoregcylTESandI = N[Abs[ErhoregcylTESandI]];
dataErhoregcylTESandI = Transpose[{rhorange / lambda0, SolErhoregcylTESandI}];

SolEphiregcylTESandI = N[Abs[EphiregcylTESandI]];
dataEphiregcylTESandI = Transpose[{rhorange / lambda0, SolEphiregcylTESandI}];

SolEAllTESandI =
  N[Abs[Sqrt[((ErhoregcylTESandI) * Cos[ϕ] - (EphiregcylTESandI) * Sin[ϕ]) ^2 +
      ((ErhoregcylTESandI) * Sin[ϕ] + (EphiregcylTESandI) * Cos[ϕ]) ^2 +
      (EzregcylTESandI) ^2]]];
dataEAllTESandI = Transpose[{rhorange / lambda0, SolEAllTESandI}];

(*Scattered Only Solution*)
SolEzregcylTES = N[Abs[EzregcylTES]];
dataEzregcylTES = Transpose[{rhorange / lambda0, SolEzregcylTES}];

SolErhoregcylTES = N[Abs[ErhoregcylTES]];
dataErhoregcylTES = Transpose[{rhorange / lambda0, SolErhoregcylTES}];

SolEphiregcylTES = N[Abs[EphiregcylTES]];
dataEphiregcylTES = Transpose[{rhorange / lambda0, SolEphiregcylTES}];

SolEAllTES = N[Abs[Sqrt[(ErhoregcylTES * Cos[ϕ] - EphiregcylTES * Sin[ϕ]) ^2 +
      (ErhoregcylTES * Sin[ϕ] + EphiregcylTES * Cos[ϕ]) ^2 + (EzregcylTES) ^2]]];
dataEAllTES = Transpose[{rhorange / lambda0, SolEAllTES}];
```


TM + TE Mode Solutions

```
In[45080]:= (*Incident + Scattered Solution*)
SolEzregcylTMplusTESandI = N[Abs[EzregcylTESandI + EzregcylTMSandI]];
dataEzregcylTMplusTESandI =
  Transpose[{rhorange / lambda0, SolEzregcylTMplusTESandI}];

SolErhoregcylTMplusTESandI = N[Abs[ErhoregcylTESandI + ErhoregcylTMSandI]];
dataErhoregcylTMplusTESandI =
  Transpose[{rhorange / lambda0, SolErhoregcylTMplusTESandI}];

SolEphiregcylTMplusTESandI = N[Abs[EphiregcylTESandI + EphiregcylTMSandI]];
dataEphiregcylTMplusTESandI =
  Transpose[{rhorange / lambda0, SolEphiregcylTMplusTESandI}];

SolEAllTMplusTESandI = N[Abs[Sqrt[((ErhoregcylTESandI + ErhoregcylTMSandI) * Cos[ϕ] -
          (EphiregcylTESandI + EphiregcylTMSandI) * Sin[ϕ]) ^2 +
      ((ErhoregcylTESandI + ErhoregcylTMSandI) * Sin[ϕ] +
          (EphiregcylTESandI + EphiregcylTMSandI) * Cos[ϕ]) ^2 +
      (EzregcylTESandI + EzregcylTMSandI) ^2]]];
dataEAllTMplusTESandI = Transpose[{rhorange / lambda0, SolEAllTMplusTESandI}];

(*Scattered Only Solution*)
SolEzregcylTMplusTES = N[Abs[EzregcylTES + EzregcylTMS]];
dataEzregcylTMplusTES = Transpose[{rhorange / lambda0, SolEzregcylTMplusTES}];

SolErhoregcylTMplusTES = N[Abs[ErhoregcylTES + ErhoregcylTMS]];
dataErhoregcylTMplusTES = Transpose[{rhorange / lambda0, SolErhoregcylTMplusTES}];

SolEphiregcylTMplusTES = N[Abs[EphiregcylTES + EphiregcylTMS]];
dataEphiregcylTMplusTES = Transpose[{rhorange / lambda0, SolEphiregcylTMplusTES}];

SolEAllTMplusTES = N[Abs[Sqrt[((ErhoregcylTES + ErhoregcylTMS) * Cos[ϕ] -
          (EphiregcylTES + EphiregcylTMS) * Sin[ϕ]) ^2 +
      ((ErhoregcylTES + ErhoregcylTMS) * Sin[ϕ] + (EphiregcylTES + EphiregcylTMS) *
           Cos[ϕ]) ^2 + (EzregcylTES + EzregcylTMS) ^2]]];
dataEAllTMplusTES = Transpose[{rhorange / lambda0, SolEAllTMplusTES}];
```

Changing Rho XY Plots

Table of Parameters

```
In[45098]:= tablevalues = {{N[E0] "V/m", "-", "-", "-", "-"}, {N[H0] "A/m", "-", "-", "-", "-"},
    {lambda0 "m", "-", "-", "-", "-"}, {f0 "Hz", "-", "-", "-", "-"},
    {a / lambda0 "λ", "-", "-", "-", "-"}, {b / lambda0 "λ", "-", "-", "-", "-"},
    {ρ1 / lambda0 "λ", "-", "-", "-", "-"}, {ρ2 / lambda0 "λ", "-", "-", "-", "-"},
    {"-", rhomin / lambda0 "λ", rhomax / lambda0 "λ", rhodelta / lambda0 "λ",
     Length[rhorange]}, {N[ϕ * (180 / Pi)] "Deg", "-", "-", "-", "-"},
    {zpoint / a "a" && zpoint / lambda0 "λ" , "-", "-", "-", "-"},
    {"-" , "-", "-", "-", jqty}, {N[θi * (180 / Pi)] "Deg", "-", "-", "-", "-"},
    {N[ϕi * (180 / Pi)] "Deg", "-", "-", "-", "-"},
    {"-", nmin, nmax, "1", nqty}, {"-", lmin, lmax, "1", lqty},
    {"-", mmin, mmax, "1", lqty}, {mmaxcheck, "-", "-", "-", "-"},
    {lmaxcheck, "-", "-", "-", "-"}, {boundarycheck, "-", "-", "-", "-"}};
  tablerowheading = {"E0", "H0", "λ", "Frequency", "a", "b", "ρ1", "ρ2",
    "ρ range", "ϕ (observed)", "z (observed)", "Matching Points", "θi",
    "ϕi", "n", "l", "m", "max allowable m", "max allowable l", "Boundary"};
  tablecolheading = {"value", "min", "max", "delta", "Qty of Points"};

In[45099]:= (*ChangingRhoXYTable =
   TableForm[tablevalues, TableHeadings→{tablerowheading, tablecolheading}]*)

In[45100]:= ChangingRhoTable = Grid[ArrayFlatten[{{{{" "}}, {tablecolheading}},
     {List /@ tablerowheading, ArrayFlatten[tablevalues]}}],
   ItemStyle → {Bold, 20}, Frame → All, Background → {{LightGray}, {LightGray}}]

In[45101]:= Export[ToString[StringForm["``.jpg", RhoTableName]], ChangingRhoTable]
```


Corrugated Cylinder Plots

Plot Ez (Scattered)

```
In[45102]:= EzCorrS = ListLinePlot[dataEzS,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolEzS]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Ez Corrugated Cylinder (Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot Ez (Inc + Scattered)

```
In[45103]:= EzCorrSandI = ListLinePlot[dataEz,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolEz]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2], PlotMarkers ->
     None, PlotLabel -> "Ez Corrugated Cylinder (Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot Erho (Scattered)

```
In[45104]:= ErhoCorrS = ListLinePlot[dataErhoS,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolErhoS]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Eρ Corrugated Cylinder (Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Plot Erho (Inc + Scattered)

```
In[45105]:= ErhoCorrSandI = ListLinePlot[dataErho,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolErho]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2], PlotMarkers ->
     None, PlotLabel -> "Eρ Corrugated Cylinder (Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot Ephi (Scattered)

```
In[45106]:= EphiCorrS = ListLinePlot[dataEphiS,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolEphiS]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Eϕ Corrugated Cylinder (Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot Ephi (Inc + Scattered)

```
In[45107]:= EphiCorrSandI = ListLinePlot[dataEphi,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolEphi]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2], PlotMarkers ->
     None, PlotLabel -> "Eϕ Corrugated Cylinder (Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Plot EAll (Scattered)

```
In[45108]:= EAllCorrS = ListLinePlot[dataEAllS,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolEAllS]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2], PlotMarkers ->
     None, PlotLabel -> "ETotal Corrugated Cylinder (Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot EAll (Inc + Scattered)

```
In[45109]:= EAllCorrSandI = ListLinePlot[dataEAllSandI,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolEAllSandI]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2], PlotMarkers ->
     None, PlotLabel -> "ETotal Corrugated Cylinder (Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary TM Plots

```
In[45110]:= GraphicsGrid[{{EzCorrS, EzCorrSandI}, {ErhoCorrS, ErhoCorrSandI},
    {EphiCorrS, EphiCorrSandI}, {EAllCorrS, EAllCorrSandI}},
   Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TM Plots

Ez (TM Scattered)

```
In[45111]:= EzRegCylTMS = ListLinePlot[dataEzregcylTMS,
     PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTMS]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Ez Smooth Cylinder (TM Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ez (TM Inc + Scattered)

```
In[45112]:= EzRegCylTMSandI = ListLinePlot[dataEzregcylTMSandI, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTMSandI]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Ez Smooth Cylinder (TM Incident + Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TM Scattered)

```
In[45113]:= ErhoRegCylTMS = ListLinePlot[dataErhoregcylTMS,
     PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolErhoregcylTMS]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eρ Smooth Cylinder (TM Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Erho (TM Inc + Scattered)

```
In[45114]:= ErhoRegCylTMSandI = ListLinePlot[dataErhoregcylTMSandI, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolErhoregcylTMSandI]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eρ Smooth Cylinder (TM Incident + Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TM Scattered)

```
In[45115]:= EphiRegCylTMS = ListLinePlot[dataEphiregcylTMS,
     PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEphiregcylTMS]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eϕ Smooth Cylinder (TM Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TM Inc + Scattered)

```
In[45116]:= EphiRegCylTMSandI = ListLinePlot[dataEphiregcylTMSandI, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEphiregcylTMSandI]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eϕ Smooth Cylinder (TM Incident + Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


EAll (TM Scattered)

```
In[45117]:= EAllRegCylTMS = ListLinePlot[dataEAllTMS,
     PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllTMS]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "ETotal Smooth Cylinder (TM Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TM Inc + Scattered)

```
In[45118]:= EAllRegCylTMSandI = ListLinePlot[dataEAllTMSandI,
     PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllTMSandI]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "ETotal Smooth Cylinder (TM Incident + Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary TM Plots

```
GraphicsGrid[{{EzRegCylTMS, EzRegCylTMSandI}, {ErhoRegCylTMS, ErhoRegCylTMSandI},
   {EphiRegCylTMS, EphiRegCylTMSandI}, {EAllRegCylTMS, EAllRegCylTMSandI}},
  Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TE Plots

Ez (TE Scattered)

```
In[45120]:= EzRegCylTES = ListLinePlot[dataEzregcylTES,
     PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTES]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Ez Smooth Cylinder (TE Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Ez (TE Inc + Scattered)

```
In[45121]:= EzRegCylTESandI = ListLinePlot[dataEzregcylTESandI, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTESandI]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Ez Smooth Cylinder (TE Incident + Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TE Scattered)

```
In[45122]:= ErhoRegCylTES = ListLinePlot[dataErhoregcylTES,
     PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolErhoregcylTES]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eρ Smooth Cylinder (TE Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TE Inc + Scattered)

```
In[45123]:= ErhoRegCylTESandI = ListLinePlot[dataErhoregcylTESandI, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolErhoregcylTESandI]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eρ Smooth Cylinder (TE Incident + Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TE Scattered)

```
In[45124]:= EphiRegCylTES = ListLinePlot[dataEphiregcylTES,
    PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEphiregcylTES]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eφ Smooth Cylinder (TE Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TE Inc + Scattered)

```
In[45125]:= EphiRegCylTESandI = ListLinePlot[dataEphiregcylTESandI, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEphiregcylTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eφ Smooth Cylinder (TE Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TE Scattered)

```
In[45126]:= EAllRegCylTES = ListLinePlot[dataEAllTES,
    PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllTES]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder ((TE Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


EAll (TE Inc + Scattered)

```
In[45127]:= EAllRegCylTESandI = ListLinePlot[dataEAllTESandI,
    PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TE Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary TE Plots

```
In[45128]:= GraphicsGrid[{{EzRegCylTES, EzRegCylTESandI},
    {ErhoRegCylTES, ErhoRegCylTESandI}, {EphiRegCylTES, EphiRegCylTESandI},
    {EAllRegCylTES, EAllRegCylTESandI}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TM + TE Plots

Ez (TM+TE Scattered)

```
In[45129]:= EzRegCylTMplusTES = ListLinePlot[dataEzregcylTMplusTES, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTMplusTES]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM + TE Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ez (TM+TE Inc + Scattered)

```
In[45130]:= EzRegCylTMplusTESandI = ListLinePlot[dataEzregcylTMplusTESandI, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTMplusTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM + TE Inc and Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Erho (TM+TE Scattered)

```
In[45131]:= ErhoRegCylTMplusTES = ListLinePlot[dataErhoregcylTMplusTES, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolErhoregcylTMplusTES]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM + TE Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TM+TE Inc + Scattered)

```
In[45132]:= ErhoRegCylTMplusTESandI = ListLinePlot[dataErhoregcylTMplusTESandI, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolErhoregcylTMplusTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM + TE Inc and Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TM+TE Scattered)

```
In[45133]:= EphiRegCylTMplusTES = ListLinePlot[dataEphiregcylTMplusTES, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEphiregcylTMplusTES]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eφ Smooth Cylinder (TM + TE Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Ephi (TM+TE Inc + Scattered)

```
In[45134]:= EphiRegCylTMplusTESandI = ListLinePlot[dataEphiregcylTMplusTESandI, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEphiregcylTMplusTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eφ Smooth Cylinder (TM + TE Inc and Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TM+TE Scattered)

```
In[45135]:= EAllRegCylTMplusTES = ListLinePlot[dataEAllTMplusTES,
    PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllTMplusTES]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM + TE Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TM+TE Inc + Scattered)

```
In[45136]:= EAllRegCylTMplusTESandI = ListLinePlot[dataEAllTMplusTESandI, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllTMplusTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM + TE Inc and Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary TM+TE Plots

```
GraphicsGrid[{{EzRegCylTMplusTES, EzRegCylTMplusTESandI},
   {ErhoRegCylTMplusTES, ErhoRegCylTMplusTESandI},
   {EphiRegCylTMplusTES, EphiRegCylTMplusTESandI},
   {EAllRegCylTMplusTES, EAllRegCylTMplusTESandI}},
  Spacings -> {Scaled[0], Scaled[0]}];
```

Compare (Smooth vs Corrugated) Plots

TM Compare

```
In[45138]:= GraphicsGrid[{{Show[EzRegCylTMS, EzCorrS, PlotRange ->
        {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTMS, SolEzS]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Ez TM Scattered Smooth Cylinder (Blue)
         vs Corrugated Cylinder (Red, --)"],
    Show[EzRegCylTMSandI, EzCorrSandI, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolEzregcylTMSandI, SolEz]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Ez TM Inc + Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"]},
   {Show[ErhoRegCylTMS, ErhoCorrS, PlotRange -> {{rhomin/lambda0, rhomax/lambda0},
        {0, Max[SolErhoregcylTMS, SolErhoS]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Eρ TM Scattered Smooth Cylinder (Blue)
         vs Corrugated Cylinder (Red, --)"],
    Show[ErhoRegCylTMSandI, ErhoCorrSandI, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolErhoregcylTMSandI, SolErho]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Eρ TM Inc + Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"]},
   {Show[EphiRegCylTMS, EphiCorrS, PlotRange -> {{rhomin/lambda0, rhomax/lambda0},
        {0, Max[SolEphiregcylTMS, SolEphiS]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Eϕ TM Scattered Smooth Cylinder (Blue)
         vs Corrugated Cylinder (Red, --)"],
    Show[EphiRegCylTMSandI, EphiCorrSandI, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolEphiregcylTMSandI, SolEphi]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Eϕ TM Inc + Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"]},
    {Show[EAllRegCylTMS, EAllCorrS, PlotRange -> {{rhomin/lambda0,
```



```
         rhomax/lambda0}, {0, Max[SolEAllS, SolEAllTMS]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "ETotal TM Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"],
    Show[EAllRegCylTMSandI, EAllCorrSandI, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolEAllTMSandI, SolEAllSandI]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "ETotal TM Inc + Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"]}
  }];
```

TE Compare

```
In[45139]:= GraphicsGrid[{{Show[EzRegCylTES, EzCorrS, PlotRange ->
        {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTES, SolEzS]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Ez TE Scattered Smooth Cylinder (Blue)
         vs Corrugated Cylinder (Red, --)"],
    Show[EzRegCylTESandI, EzCorrSandI, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolEzregcylTESandI, SolEz]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Ez TE Inc + Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"]},
   {Show[ErhoRegCylTES, ErhoCorrS, PlotRange -> {{rhomin/lambda0, rhomax/lambda0},
        {0, Max[SolErhoregcylTES, SolErhoS]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Eρ TE Scattered Smooth Cylinder (Blue)
         vs Corrugated Cylinder (Red, --)"],
    Show[ErhoRegCylTESandI, ErhoCorrSandI, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolErhoregcylTESandI, SolErho]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Eρ TE Inc + Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"]},
   {Show[EphiRegCylTES, EphiCorrS, PlotRange -> {{rhomin/lambda0, rhomax/lambda0},
        {0, Max[SolEphiregcylTES, SolEphiS]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
```



```
      PlotLabel -> "Eϕ TE Scattered Smooth Cylinder (Blue)
         vs Corrugated Cylinder (Red, --)"],
    Show[EphiRegCylTESandI, EphiCorrSandI, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolEphiregcylTESandI, SolEphi]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Eϕ TE Inc + Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"]},
    {Show[EAllRegCylTES, EAllCorrS, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolEAllTES, SolEAllS]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "ETotal TE Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"],
    Show[EAllRegCylTESandI, EAllCorrSandI, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolEAllTESandI, SolEAllSandI]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "ETotal TE Inc + Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"]}
  }];
```

TM+TE Compare

```
In[45140]:= GraphicsGrid[
   {{Show[EzRegCylTMplusTES, EzCorrS, PlotRange -> {{rhomin/lambda0, rhomax/
          lambda0}, {0, Max[SolEzregcylTMplusTES, SolEzS]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, Frame -> True,
      FrameLabel -> {"ρ/λ", "V/m"}, PlotLabel -> "Ez TM+TE Scattered Smooth
         Cylinder (Blue) vs Corrugated Cylinder (Red, --)"],
    Show[EzRegCylTMplusTESandI, EzCorrSandI, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolEzregcylTMplusTESandI, SolEz]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Ez TM+TE Inc + Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"]},
   {Show[ErhoRegCylTMplusTES, ErhoCorrS, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolErhoregcylTMplusTES, SolErhoS]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Eρ TM+TE Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"],
    Show[ErhoRegCylTMplusTESandI, ErhoCorrSandI, PlotRange -> {{rhomin/lambda0,
```



```
         rhomax/lambda0}, {0, Max[SolErhoregcylTMplusTESandI, SolErho]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Eρ TM+TE Inc + Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"]},
   {Show[EphiRegCylTMplusTES, EphiCorrS, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolEphiregcylTMplusTES, SolEphiS]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Eϕ TM+TE Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"],
    Show[EphiRegCylTMplusTESandI, EphiCorrSandI, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolEphiregcylTMplusTESandI, SolEphi]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Eϕ TM+TE Inc + Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"]},
    {Show[EAllRegCylTES, EAllCorrS, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolEAllS, SolEAllTMplusTES]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "ETotal TM+TE Scattered Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)"],
    Show[EAllRegCylTMplusTESandI, EAllCorrSandI, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolEAllTMplusTESandI, SolEAllSandI]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "ETotal TM+TE Inc + Scattered Smooth
         Cylinder (Blue) vs Corrugated Cylinder (Red, --)"]}
  }];
```

TM, TE, TM+TE Compare

```
In[45141]:= RhoXYPlots = GraphicsGrid[{{Show[EzRegCylTMS, EzCorrS, PlotRange ->
        {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTMS, SolEzS]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
      PlotLabel -> "Ez TM Scattered Smooth Cylinder (Blue)
         vs Corrugated Cylinder (Red, --)"],
     "Ez TE Does Not Exist",
     Show[EzRegCylTMplusTES, EzCorrS, PlotRange -> {{rhomin/lambda0,
         rhomax/lambda0}, {0, Max[SolEzregcylTMplusTES, SolEzS]}},
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
```

```
 Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
 PlotLabel -> "Ez TM+TE Scattered Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)"]},

{Show[EzRegCylTMSandI, EzCorrSandI, PlotRange ->
   {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTMSandI, SolEz]}},
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
 PlotLabel -> "Ez TM Inc + Scattered Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)"],
 "Ez TE Does Not Exist",
 Show[EzRegCylTMplusTESandI,
 EzCorrSandI, PlotRange -> {{rhomin/lambda0, rhomax/lambda0},
   {0, Max[SolEzregcylTMplusTESandI, SolEz]}},
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
 PlotLabel -> "Ez TM+TE Inc + Scattered Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)"]},

{Show[ErhoRegCylTMS, ErhoCorrS, PlotRange ->
   {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolErhoregcylTMS, SolErhoS]}},
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
 PlotLabel -> "Eρ TM Scattered Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)"],
 Show[ErhoRegCylTES, ErhoCorrS, PlotRange -> {{rhomin/lambda0,
    rhomax/lambda0}, {0, Max[SolErhoregcylTES, SolErhoS]}},
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
 PlotLabel -> "Eρ TE Scattered Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)"],
 Show[ErhoRegCylTMplusTES, ErhoCorrS, PlotRange -> {{rhomin/lambda0,
    rhomax/lambda0}, {0, Max[SolErhoregcylTMplusTES, SolErhoS]}},
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
 PlotLabel -> "Eρ TM+TE Scattered Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)"]},

{Show[ErhoRegCylTMSandI, ErhoCorrSandI, PlotRange -> {{rhomin/lambda0,
    rhomax/lambda0}, {0, Max[SolErhoregcylTMSandI, SolErho]}},
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
 PlotLabel -> "Eρ TM Inc + Scattered Smooth Cylinder
```



```
    (Blue) vs Corrugated Cylinder (Red, --)"],
 Show[ErhoRegCylTESandI, ErhoCorrSandI, PlotRange -> {{rhomin/lambda0,
    rhomax/lambda0}, {0, Max[SolErhoregcylTESandI, SolErho]}},
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
  PlotLabel -> "Eρ TE Inc + Scattered Smooth Cylinder (Blue) vs Corrugated
    Cylinder (Red, --)"], Show[ErhoRegCylTMplusTESandI,
  ErhoCorrSandI, PlotRange -> {{rhomin/lambda0, rhomax/lambda0},
   {0, Max[SolErhoregcylTMplusTESandI, SolErho]}},
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
  PlotLabel -> "Eρ TM+TE Inc + Scattered Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)"]},

{Show[EphiRegCylTMS, EphiCorrS, PlotRange ->
   {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEphiregcylTMS, SolEphiS]}},
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
  PlotLabel -> "Eφ TM Scattered Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)"],
 Show[EphiRegCylTES, EphiCorrS, PlotRange -> {{rhomin/lambda0,
    rhomax/lambda0}, {0, Max[SolEphiregcylTES, SolEphiS]}},
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
  PlotLabel -> "Eφ TE Scattered Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)"],
 Show[EphiRegCylTMplusTES, EphiCorrS, PlotRange -> {{rhomin/lambda0,
    rhomax/lambda0}, {0, Max[SolEphiregcylTMplusTES, SolEphiS]}},
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
  PlotLabel -> "Eφ TM+TE Scattered Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)"]},

{Show[EphiRegCylTMSandI, EphiCorrSandI, PlotRange -> {{rhomin/lambda0,
    rhomax/lambda0}, {0, Max[SolEphiregcylTMSandI, SolEphi]}},
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
  PlotLabel -> "Eφ TM Inc + Scattered Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)"],
 Show[EphiRegCylTESandI, EphiCorrSandI, PlotRange -> {{rhomin/lambda0,
    rhomax/lambda0}, {0, Max[SolEphiregcylTESandI, SolEphi]}},
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
```



```
 PlotLabel -> "Eφ TE Inc + Scattered Smooth Cylinder (Blue) vs Corrugated
    Cylinder (Red, --)"], Show[EphiRegCylTMplusTESandI,
 EphiCorrSandI, PlotRange -> {{rhomin/lambda0, rhomax/lambda0},
   {0, Max[SolEphiregcylTMplusTESandI, SolEphi]}},
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
 PlotLabel -> "Eφ TM+TE Inc + Scattered Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)"]},

{Show[EAllRegCylTMS, EAllCorrS, PlotRange ->
   {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllS, SolEAllTMS]}},
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
 PlotLabel -> "ETotal TM Scattered Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)"],
 Show[EAllRegCylTES, EAllCorrS, PlotRange -> {{rhomin/lambda0,
    rhomax/lambda0}, {0, Max[SolEAllS, SolEAllTES]}},
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
 PlotLabel -> "ETotal TE Scattered Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)"],
 Show[EAllRegCylTMplusTES, EAllCorrS, PlotRange ->
   {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllS, SolEAllTMplusTES]}},
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
 PlotLabel -> "ETotal TM+TE Scattered Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)"]},

{Show[EAllRegCylTMSandI, EAllCorrSandI, PlotRange -> {{rhomin/lambda0,
    rhomax/lambda0}, {0, Max[SolEAllTMSandI, SolEAllSandI]}},
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
 PlotLabel -> "ETotal TM Inc + Scattered Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)"],
 Show[EAllRegCylTMplusTESandI, EAllCorrSandI, PlotRange -> {{rhomin/lambda0,
    rhomax/lambda0}, {0, Max[SolEAllTMplusTESandI, SolEAllSandI]}},
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
 PlotLabel -> "ETotal TM+TE Inc + Scattered Smooth
    Cylinder (Blue) vs Corrugated Cylinder (Red, --)"],
 Show[EAllRegCylTMplusTESandI, EAllCorrSandI, PlotRange -> {{rhomin/lambda0,
    rhomax/lambda0}, {0, Max[SolEAllTMplusTESandI, SolEAllSandI]}},
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
```



```
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "ETotal TM+TE Inc + Scattered Smooth
       Cylinder (Blue) vs Corrugated Cylinder (Red, --)"]}
   }]
```

```
In[45142]:= Export[ToString[StringForm["``.jpg", RhoXYPlotsName]], RhoXYPlots];
Speak["The rho plots are done"]
```

Empty Arrays for Storing Field Solutions - Changing Phi

Corrugated Cylinder Empty Arrays

Scattered + Incident Arrays

```
In[45144]:= Eztemp = ConstantArray[0, Length[phirange]];
Ephitemp = ConstantArray[0, Length[phirange]];
Erhotemp = ConstantArray[0, Length[phirange]];
```

Scattered Only Arrays

```
In[45147]:= EztempS = ConstantArray[0, Length[phirange]];
EphitempS = ConstantArray[0, Length[phirange]];
ErhotempS = ConstantArray[0, Length[phirange]];
```

Incident Only Arrays

```
In[45150]:= EztempI = ConstantArray[0, Length[phirange]];
EphitempI = ConstantArray[0, Length[phirange]];
ErhotempI = ConstantArray[0, Length[phirange]];
```

Regular (“Smooth”) Cylinder Empty Arrays

Scattered + Incident Arrays (TM and TE)

```
In[45153]:= EzregcylTMSandI = ConstantArray[0, Length[phirange]];
ErhoregcylTMSandI = ConstantArray[0, Length[phirange]];
EphiregcylTMSandI = ConstantArray[0, Length[phirange]];
EzregcylTESandI = ConstantArray[0, Length[phirange]];
ErhoregcylTESandI = ConstantArray[0, Length[phirange]];
EphiregcylTESandI = ConstantArray[0, Length[phirange]];
```

Scattered Only Arrays (TM and TE)

```
In[45158]:= EzregcylTMS = ConstantArray[0, Length[phirange]];
ErhoregcylTMS = ConstantArray[0, Length[phirange]];
EphiregcylTMS = ConstantArray[0, Length[phirange]];
EzregcylTES = ConstantArray[0, Length[phirange]];
ErhoregcylTES = ConstantArray[0, Length[phirange]];
EphiregcylTES = ConstantArray[0, Length[phirange]];
```

Incident Only Arrays (TM and TE)

```
In[45165]:= EzregcylTMI = ConstantArray[0, Length[phirange]];
ErhoregcylTMI = ConstantArray[0, Length[phirange]];
EphiregcylTMI = ConstantArray[0, Length[phirange]];
EzregcylTEI = ConstantArray[0, Length[phirange]];
ErhoregcylTEI = ConstantArray[0, Length[phirange]];
EphiregcylTEI = ConstantArray[0, Length[phirange]];
```

Changing Phi Do Loop Calculation

Do Loop

```
In[45171]:= count = 1;

In[45172]:= ncount = 1;
lcount = 1;
jcount = 1;

ρ = ρfar; (*Observation ρ for use in polar plots*)

Do[
  (*ϕi=ϕ;*)
  ϕi = ϕivalue; (*Only if considered unique and distinct from scattered field*)

  (*Region II temp variables, Scattered Field Component*)
  EIIzstemp = 0;
  EIIphistemp = 0;
  EIIrhostemp = 0;

  HIIzstemp = 0;
```



```
  HIIphistemp = 0;
  HIIrhostemp = 0;

  (*Region II temp variables, Incident Field Component*)
  EIIzItemp = 0;
  EIIphiItemp = 0;
  EIIrhoItemp = 0;

  (*Region II temp variables, Total Field*)
  EIIztemp = 0;
  EIIphitemp = 0;
  HIIztemp = 0;
  HIIphitemp = 0;
  EIIrhotemp = 0;
  HIIrhotemp = 0;

  (*Region I temp variables*)
  EIztemp = 0;
  EIphitemp = 0;
  HIztemp = 0;
  HIphitemp = 0;
  EIrhotemp = 0;
  HIrhotemp = 0;

  (*Smooth Cylinder*)
  If[ρ < ρ2,
   (*TRUE*)
   EzregcylTMSandI[[count]] = 0;
   EzregcylTMS[[count]] = 0;
   EzregcylTMI[[count]] = 0;
   ErhoregcylTMSandI[[count]] = 0;
   ErhoregcylTMS[[count]] = 0;
   ErhoregcylTMI[[count]] = 0;
   EphiregcylTMSandI[[count]] = 0;
   EphiregcylTMS[[count]] = 0;
   EphiregcylTMI[[count]] = 0;
   EzregcylTESandI[[count]] = 0;
   EzregcylTES[[count]] = 0;
   EzregcylTEI[[count]] = 0;
   ErhoregcylTESandI[[count]] = 0;
   ErhoregcylTES[[count]] = 0;
   ErhoregcylTEI[[count]] = 0;
   EphiregcylTESandI[[count]] = 0;
```



```
   EphiregcylTES[[count]] = 0;
   EphiregcylTEI[[count]] = 0;
   ,
   (*FALSE*)
   EzregcylTMSandI[[count]] = EzcylTMSandI[ϕi, ϕ, z, ρ];
   EzregcylTMS[[count]] = EzcylTMS[ϕ, z, ρ];
   EzregcylTMI[[count]] = EzcylTMI[ϕi, z, ρ];

   ErhoregcylTMSandI[[count]] = ErhocylTMSandI[ϕi, ϕ, z, ρ];
   ErhoregcylTMS[[count]] = ErhocylTMS[ϕ, z, ρ];
   ErhoregcylTMI[[count]] = ErhocylTMI[ϕi, z, ρ];

   EphiregcylTMSandI[[count]] = EphicylTMSandI[ϕi, ϕ, z, ρ];
   EphiregcylTMS[[count]] = EphicylTMS[ϕ, z, ρ];
   EphiregcylTMI[[count]] = EphicylTMI[ϕi, z, ρ];

   EzregcylTESandI[[count]] = EzcylTESandI[ϕi, ϕ, z, ρ];
   EzregcylTES[[count]] = EzcylTES[ϕ, z, ρ];
   EzregcylTEI[[count]] = EzcylTEI[ϕi, z, ρ];

   ErhoregcylTESandI[[count]] = ErhocylTESandI[ϕi, ϕ, z, ρ];
   ErhoregcylTES[[count]] = ErhocylTES[ϕ, z, ρ];
   ErhoregcylTEI[[count]] = ErhocylTEI[ϕi, z, ρ];

   EphiregcylTESandI[[count]] = EphicylTESandI[ϕi, ϕ, z, ρ];
   EphiregcylTES[[count]] = EphicylTES[ϕ, z, ρ];
   EphiregcylTEI[[count]] = EphicylTEI[ϕi, z, ρ];


  ];

  If[ρ ≥ ρ1, (*Check to see if inside conductor or not*)
   If[-a/2 ≤ z ≤ a/2, (*Check to see if in boundary 'a' or boundary 'b'*)
     (*--------------IF TRUE--------------*)
     (* In boundary 'a' but still need to determine if in Region I or II*)
     boundarycheck = "Boundary a";
     If[ρ < ρ2, (*Region I fields, boundary 'a'*)
      (*Region I fields, boundary 'a'*)
      For[ncount = 1, ncount ≤ nqty, ncount++,
        n = nrange[[ncount]];
        For[lcount = 1, lcount ≤ lqty, lcount++,
         l = lrange[[lcount]];
         m = lrange[[lcount]];
```



```
         index = ((ncount - 1) * (lqty)) + lcount;
         EIztemp = EIztemp + (hnmjeq[ϕ, z, ρ] * Anm[[index]]);
         EIphitemp = EIphitemp +
           (knmjeq[ϕ, z, ρ] * Anm[[index]] + lnmjeq[ϕ, z, ρ] * Bnm[[index]]);
         HIztemp = HIztemp + (inmjeq[ϕ, z, ρ] * Bnm[[index]]);
         HIphitemp = HIphitemp +
           (onmjeq[ϕ, z, ρ] * Anm[[index]] + pnmjeq[ϕ, z, ρ] * Bnm[[index]]);
         EIrhotemp = EIrhotemp + (rnmjeq[ϕ, z, ρ] * Anm[[index]] +
             snmjeq[ϕ, z, ρ] * Bnm[[index]]);
         HIrhotemp = HIrhotemp + (tnmjeq[ϕ, z, ρ] * Anm[[index]] +
             unmjeq[ϕ, z, ρ] * Bnm[[index]]);
        ];
      ];
      ,
      (*Region II fields, boundary 'a'*)
      For[ncount = 1, ncount ≤ nqty, ncount++,
       n = nrange[[ncount]];
       For[lcount = 1, lcount ≤ lqty, lcount++,
        l = lrange[[lcount]];
        m = lrange[[lcount]];
        index = ((ncount - 1) * (lqty)) + lcount;
        (*Scattered Field Portion First*)
        EIIzstemp = EIIzstemp + anljeq[ϕ, z, ρ] * Cnla[[index]];
        EIIphistemp = EIIphistemp +
          (cnljeq[ϕ, z, ρ] * Cnla[[index]] + enljeq[ϕ, z, ρ] * Dnla[[index]]);
        HIIzstemp = HIIzstemp + gnljeq[ϕ, z, ρ] * Dnla[[index]];
        HIIphistemp = HIIphistemp +
          (nnljeq[ϕ, z, ρ] * Cnla[[index]] + qnljeq[ϕ, z, ρ] * Dnla[[index]]);
        EIIrhostemp = EIIrhostemp + (vnljeq[ϕ, z, ρ] * Cnla[[index]] +
            wnljeq[ϕ, z, ρ] * Dnla[[index]]);
        HIIrhostemp = HIIrhostemp + (xnljeq[ϕ, z, ρ] * Cnla[[index]] +
            ynljeq[ϕ, z, ρ] * Dnla[[index]]);
       ];
      ];
      (*Add Incident and Scattered Field*)
      EIIztemp = bnjeq[ϕi, z, ρ] + EIIzstemp;
      EIIphitemp = dnjeq[ϕi, z, ρ] + EIIphistemp;
      HIIztemp = fnjeq[ϕi, z, ρ] + HIIzstemp;
      HIIphitemp = mnjeq[ϕi, z, ρ] + HIIphistemp;
      EIIrhotemp = γnjeq[ϕi, z, ρ] + EIIrhostemp;
      HIIrhotemp = δnjeq[ϕi, z, ρ] + HIIrhostemp;

      (*Incident Fields Only*)
```

```
 EztempI[[count]] = bnjeq[ϕi, z, ρ];
 EphitempI[[count]] = dnjeq[ϕi, z, ρ];
 ErhotempI[[count]] = γnjeq[ϕi, z, ρ];

 (*Scattered Field Only*)
 EIIzstemp = EIIzstemp;
 EIIphistemp = EIIphistemp;
 EIIrhostemp = EIIrhostemp;

];
,
(*--------------IF FALSE--------------*) (* In boundary '
 b' but still need to determine if in Cylinder or Region II*)
boundarycheck = "Boundary b";
If[ρ < ρ2, (*In Cylinder, boundary 'b'*)
 Eztemp[[count]] = Eztemp[[count]] + 0;
 Ephitemp[[count]] = Ephitemp[[count]] + 0;
 Erhotemp[[count]] = Erhotemp[[count]] + 0;

 EztempS[[count]] = EztempS[[count]] + 0;
 EphitempS[[count]] = EphitempS[[count]] + 0;
 ErhotempS[[count]] = ErhotempS[[count]] + 0;

 Hztemp[[count]] = Hztemp[[count]] + 0;
 Hphitemp[[count]] = Hphitemp[[count]] + 0;
 Hrhotemp[[count]] = Hrhotemp[[count]] + 0;,
 (*Region II fields, boundary 'b'*)
 For[ncount = 1, ncount ≤ nqty, ncount++,
  n = nrange[[ncount]];
  For[lcount = 1, lcount ≤ lqty, lcount++,
   l = lrange[[lcount]];
   m = lrange[[lcount]];
   index = ((ncount - 1) * (lqty)) + lcount;
   (*Scattered Field Portion First*)
   EIIzstemp = EIIzstemp + anljeq[ϕ, z, ρ] * Cnlb[[index]];
   EIIphistemp = EIIphistemp +
     (cnljeq[ϕ, z, ρ] * Cnlb[[index]] + enljeq[ϕ, z, ρ] * Dnlb[[index]]);
   HIIzstemp = HIIzstemp + gnljeq[ϕ, z, ρ] * Dnlb[[index]];
   HIIphistemp = HIIphistemp +
     (nnljeq[ϕ, z, ρ] * Cnlb[[index]] + qnljeq[ϕ, z, ρ] * Dnlb[[index]]);
   EIIrhostemp = EIIrhostemp + (vnljeq[ϕ, z, ρ] * Cnlb[[index]] +
       wnljeq[ϕ, z, ρ] * Dnlb[[index]]);
   HIIrhostemp = HIIrhostemp + (xnljeq[ϕ, z, ρ] * Cnlb[[index]] +
```



```
        ynljeq[ϕ, z, ρ] * Dnlb[[index]]);
    ];
   ];
   (*Add Incident and Scattered Field*)

   EIIztemp = bnjeq[ϕi, z, ρ] + EIIzstemp;
   EIIphitemp = dnjeq[ϕi, z, ρ] + EIIphistemp;
   HIIztemp = fnjeq[ϕi, z, ρ] + HIIzstemp;
   HIIphitemp = mnjeq[ϕi, z, ρ] + HIIphistemp;
   EIIrhotemp = γnjeq[ϕi, z, ρ] + EIIrhostemp;
   HIIrhotemp = δnjeq[ϕi, z, ρ] + HIIrhostemp;

   (*Incident Fields Only*)
   EztempI[[count]] = bnjeq[ϕi, z, ρ];
   EphitempI[[count]] = dnjeq[ϕi, z, ρ];
   ErhotempI[[count]] = γnjeq[ϕi, z, ρ];

   (*Scattered Field Only*)
   EIIzstemp = EIIzstemp;
   EIIphistemp = EIIphistemp;
   EIIrhostemp = EIIrhostemp;

  ];
 ] (*END IF STATEMENT*);,
 Eztemp[[count]] = Eztemp[[count]] + 0;
 Ephitemp[[count]] = Ephitemp[[count]] + 0;
 Erhotemp[[count]] = Erhotemp[[count]] + 0;

 EztempS[[count]] = EztempS[[count]] + 0;
 EphitempS[[count]] = EphitempS[[count]] + 0;
 ErhotempS[[count]] = ErhotempS[[count]] + 0;

 Hztemp[[count]] = Hztemp[[count]] + 0;
 Hphitemp[[count]] = Hphitemp[[count]] + 0;
 Hrhotemp[[count]] = Hrhotemp[[count]] + 0;

 (*All E-fields zeros*)
];

Eztemp[[count]] = EIztemp + EIIztemp;
Ephitemp[[count]] = EIphitemp + EIIphitemp;
Erhotemp[[count]] = EIrhotemp + EIIrhotemp;
```



```
  EztempS[[count]] = EztempS[[count]] + EIIzstemp;
  EphitempS[[count]] = EphitempS[[count]] + EIIphistemp;
  ErhotempS[[count]] = ErhotempS[[count]] + EIIrhostemp;

  Hztemp[[count]] = HIztemp + HIIztemp;
  Hphitemp[[count]] = HIphitemp + HIIphitemp;
  Hrhotemp[[count]] = HIrhotemp + HIIrhotemp;

  count = count + 1;
  , {ϕ, phirange[[1]], phirange[[Length[phirange]]], phidelta}];
 (*Close of 'Do loop'*)
```

Solution for Plots - Corrugated Cylinder

```
In[45178]:= (*Incident + Scattered Solution*)
SolEz = N[Abs[Eztemp]];
dataEz = Transpose[{phirange, SolEz}];

SolEphi = N[Abs[Ephitemp]];
dataEphi = Transpose[{phirange, SolEphi}];

SolErho = N[Abs[Erhotemp]];
dataErho = Transpose[{phirange, SolErho}];

SolEAllSandI =
  N[Abs[Sqrt[((Erhotemp) * Cos[phirange] - (Ephitemp) * Sin[phirange]) ^2 +
      ((Erhotemp) * Sin[phirange] + (Ephitemp) * Cos[phirange]) ^2 + (Eztemp) ^2]]];
dataEAllSandI = Transpose[{phirange, SolEAllSandI}];

(*Scattered Only Solution*)
SolEzS = N[Abs[EztempS]];
dataEzS = Transpose[{phirange, SolEzS}];

SolEphiS = N[Abs[EphitempS]];
dataEphiS = Transpose[{phirange, SolEphiS}];

SolErhoS = N[Abs[ErhotempS]];
dataErhoS = Transpose[{phirange, SolErhoS}];

SolEAllS = N[Abs[Sqrt[(ErhotempS * Cos[phirange] - EphitempS * Sin[phirange]) ^2 +
      (ErhotempS * Sin[phirange] + EphitempS * Cos[phirange]) ^2 + (EztempS) ^2]]];
dataEAllS = Transpose[{phirange, SolEAllS}];
```



```
(*RCS Solution*)
RCSSolEz = (4 * Pi * (ρ^2)) * (((Abs[EztempS]) ^2) / ((Abs[EztempI]) ^2));
RCSdataEz = Transpose[{phirange, RCSSolEz}];

RCSSolErho = (4 * Pi * (ρ^2)) * (((Abs[ErhotempS]) ^2) / ((Abs[ErhotempI]) ^2));
RCSdataErho = Transpose[{phirange, RCSSolErho}];

RCSSolEphi = (4 * Pi * (ρ^2)) * (((Abs[EphitempS]) ^2) / ((Abs[EphitempI]) ^2));
RCSdataEphi = Transpose[{phirange, RCSSolEphi}];

RCSSolEAllS = Sqrt[(ErhotempS * Cos[phirange] - EphitempS * Sin[phirange]) ^2 +
    (ErhotempS * Sin[phirange] + EphitempS * Cos[phirange]) ^2 + (EztempS) ^2];
RCSSolEAllI = Sqrt[(ErhotempI * Cos[phirange] - EphitempI * Sin[phirange]) ^2 +
    (ErhotempI * Sin[phirange] + EphitempI * Cos[phirange]) ^2 + (EztempI) ^2];
RCSSolEAll = (4 * Pi * (ρ^2)) * (((Abs[RCSSolEAllS]) ^2) / ((Abs[RCSSolEAllI]) ^2));
RCSdataEAll = Transpose[{phirange, RCSSolEAll}];

(*RCS Solution - in dB*)
RCSdataEzdB = Transpose[{phirange, 10 * Log10[RCSSolEz]}];
RCSdataErhodB = Transpose[{phirange, 10 * Log10[RCSSolErho]}];
RCSdataEphidB = Transpose[{phirange, 10 * Log10[RCSSolEphi]}];
RCSdataEAlldB = Transpose[{phirange, 10 * Log10[RCSSolEAll]}];
```

Solution for Plots - Regular “Smooth” Cylinder

TM Mode Solutions

```
In[45209]:= (*Incident + Scattered Solution*)
SolEzregcylTMSandI = N[Abs[EzregcylTMSandI]];
dataEzregcylTMSandI = Transpose[{phirange, SolEzregcylTMSandI}];

SolErhoregcylTMSandI = N[Abs[ErhoregcylTMSandI]];
dataErhoregcylTMSandI = Transpose[{phirange, SolErhoregcylTMSandI}];

SolEphiregcylTMSandI = N[Abs[EphiregcylTMSandI]];
dataEphiregcylTMSandI = Transpose[{phirange, SolEphiregcylTMSandI}];

SolEAllTMS =
  N[Abs[Sqrt[(ErhoregcylTMS * Cos[phirange] - EphiregcylTMS * Sin[phirange]) ^2 +
      (ErhoregcylTMS * Sin[phirange] + EphiregcylTMS * Cos[phirange]) ^2 +
      (EzregcylTMS) ^2]]];
dataEAllTMS = Transpose[{phirange, SolEAllTMS}];
```

PhD Research Garcia_Final_v4.1.nb | 61

```
(*Scattered Only Solution*)
SolEzregcylTMS = N[Abs[EzregcylTMS]];
dataEzregcylTMS = Transpose[{phirange, SolEzregcylTMS}];

SolErhoregcylTMS = N[Abs[ErhoregcylTMS]];
dataErhoregcylTMS = Transpose[{phirange, SolErhoregcylTMS}];

SolEphiregcylTMS = N[Abs[EphiregcylTMS]];
dataEphiregcylTMS = Transpose[{phirange, SolEphiregcylTMS}];

SolEAllTMSandI = N[Abs[Sqrt[
     ((ErhoregcylTMSandI) * Cos[phirange] - (EphiregcylTMSandI) * Sin[phirange]) ^2 +
      ((ErhoregcylTMSandI) * Sin[phirange] + (EphiregcylTMSandI) * Cos[phirange]) ^2 +
      (EzregcylTMSandI) ^2]]];
dataEAllTMSandI = Transpose[{phirange, SolEAllTMSandI}];

(*RCS Solution*)
RCSSolEzregcylTM =
  (4 * Pi * (ρ^2)) * (((Abs[EzregcylTMS]) ^2) / ((Abs[EzregcylTMI]) ^2));
RCSdataEzregcylTM = Transpose[{phirange, RCSSolEzregcylTM}];

RCSSolErhoregcylTM =
  (4 * Pi * (ρ^2)) * (((Abs[ErhoregcylTMS]) ^2) / ((Abs[ErhoregcylTMI]) ^2));
RCSdataErhoregcylTM = Transpose[{phirange, RCSSolErhoregcylTM}];

RCSSolEphiregcylTM =
  (4 * Pi * (ρ^2)) * (((Abs[EphiregcylTMS]) ^2) / ((Abs[EphiregcylTMI]) ^2));
RCSdataEphiregcylTM = Transpose[{phirange, RCSSolEphiregcylTM}];

RCSSolEregcylTMAllS =
  Sqrt[(ErhoregcylTMS * Cos[phirange] - EphiregcylTMS * Sin[phirange]) ^2 +
    (ErhoregcylTMS * Sin[phirange] + EphiregcylTMS * Cos[phirange]) ^2 +
    (EzregcylTMS) ^2];
RCSSolEregcylTMAllI = Sqrt[
   (ErhoregcylTMI * Cos[phirange] - EphiregcylTMI * Sin[phirange]) ^2 +
    (ErhoregcylTMI * Sin[phirange] + EphiregcylTMI * Cos[phirange]) ^2 +
    (EzregcylTMI) ^2];
RCSSolEregcylTMAll = (4 * Pi * (ρ^2)) *
   (((Abs[RCSSolEregcylTMAllS]) ^2) / ((Abs[RCSSolEregcylTMAllI]) ^2));
RCSdataEregcylTMAll = Transpose[{phirange, RCSSolEregcylTMAll}];

(*RCS Solution - in dB*)
```

Printed by Wolfram Mathematica Student Edition

62 | PhD Research Garcia_Final_v4.1.nb

```
RCSdataEzregcylTMdB = Transpose[{phirange, 10 * Log10[RCSSolEzregcylTM]}];
RCSdataErhoregcylTMdB = Transpose[{phirange, 10 * Log10[RCSSolErhoregcylTM]}];
RCSdataEphiregcylTMdB = Transpose[{phirange, 10 * Log10[RCSSolEphiregcylTM]}];
RCSdataEregcylTMAlldB = Transpose[{phirange, 10 * Log10[RCSSolEregcylTMAll]}];
```

TE Mode Solutions

```
In[45240]:= (*Incident + Scattered Solution*)
SolEzregcylTESandI = N[Abs[EzregcylTESandI]];
dataEzregcylTESandI = Transpose[{phirange, SolEzregcylTESandI}];

SolErhoregcylTESandI = N[Abs[ErhoregcylTESandI]];
dataErhoregcylTESandI = Transpose[{phirange, SolErhoregcylTESandI}];

SolEphiregcylTESandI = N[Abs[EphiregcylTESandI]];
dataEphiregcylTESandI = Transpose[{phirange, SolEphiregcylTESandI}];

SolEAllTESandI = N[Abs[Sqrt[
     ((ErhoregcylTESandI) * Cos[phirange] - (EphiregcylTESandI) * Sin[phirange]) ^2 +
      ((ErhoregcylTESandI) * Sin[phirange] + (EphiregcylTESandI) * Cos[phirange]) ^2 +
      (EzregcylTESandI) ^2]]];
dataEAllTESandI = Transpose[{phirange, SolEAllTESandI}];

(*Scattered Only Solution*)
SolEzregcylTES = N[Abs[EzregcylTES]];
dataEzregcylTES = Transpose[{phirange, SolEzregcylTES}];

SolErhoregcylTES = N[Abs[ErhoregcylTES]];
dataErhoregcylTES = Transpose[{phirange, SolErhoregcylTES}];

SolEphiregcylTES = N[Abs[EphiregcylTES]];
dataEphiregcylTES = Transpose[{phirange, SolEphiregcylTES}];

SolEAllTES =
  N[Abs[Sqrt[(ErhoregcylTES * Cos[phirange] - EphiregcylTES * Sin[phirange]) ^2 +
      (ErhoregcylTES * Sin[phirange] + EphiregcylTES * Cos[phirange]) ^2 +
      (EzregcylTES) ^2]]];
dataEAllTES = Transpose[{phirange, SolEAllTES}];

(*RCS Solution*)
(*RCSSolEzregcylTE = DNE;*)
(*RCSdataEzregcylTE =DNE;*)
RCSSolErhoregcylTE =
```

Printed by Wolfram Mathematica Student Edition

PhD Research Garcia_Final_v4.1.nb | 63

```
  (4 * Pi * (ρ^2)) * (((Abs[ErhoregcylTES]) ^2) / ((Abs[ErhoregcylTEI]) ^2));
RCSdataErhoregcylTE = Transpose[{phirange, RCSSolErhoregcylTE}];

RCSSolEphiregcylTE =
  (4 * Pi * (ρ^2)) * (((Abs[EphiregcylTES]) ^2) / ((Abs[EphiregcylTEI]) ^2));
RCSdataEphiregcylTE = Transpose[{phirange, RCSSolEphiregcylTE}];

RCSSolEregcylTEAllS =
  Sqrt[(ErhoregcylTES * Cos[phirange] - EphiregcylTES * Sin[phirange]) ^2 +
    (ErhoregcylTES * Sin[phirange] + EphiregcylTES * Cos[phirange]) ^2 +
    (EzregcylTES) ^2];
RCSSolEregcylTEAllI = Sqrt[
   (ErhoregcylTEI * Cos[phirange] - EphiregcylTEI * Sin[phirange]) ^2 +
    (ErhoregcylTEI * Sin[phirange] + EphiregcylTEI * Cos[phirange]) ^2 +
    (EzregcylTEI) ^2];
RCSSolEregcylTEAll = (4 * Pi * (ρ^2)) *
   (((Abs[RCSSolEregcylTEAllS]) ^2) / ((Abs[RCSSolEregcylTEAllI]) ^2));
RCSdataEregcylTEAll = Transpose[{phirange, RCSSolEregcylTEAll}];

(*RCS Solution - in dB*)
RCSdataErhoregcylTEdB = Transpose[{phirange, 10 * Log10[ RCSSolErhoregcylTE]}];
RCSdataEphiregcylTEdB = Transpose[{phirange, 10 * Log10[RCSSolEphiregcylTE]}];
RCSdataEregcylTEAlldB = Transpose[{phirange, 10 * Log10[RCSSolEregcylTEAll]}];
```

TM + TE Mode Solutions

```
In[45268]:= (*Incident + Scattered Solution*)
SolEzregcylTMplusTESandI = N[Abs[EzregcylTESandI + EzregcylTMSandI]];
dataEzregcylTMplusTESandI = Transpose[{phirange, SolEzregcylTMplusTESandI}];

SolErhoregcylTMplusTESandI = N[Abs[ErhoregcylTESandI + ErhoregcylTMSandI]];
dataErhoregcylTMplusTESandI = Transpose[{phirange, SolErhoregcylTMplusTESandI}];

SolEphiregcylTMplusTESandI = N[Abs[EphiregcylTESandI + EphiregcylTMSandI]];
dataEphiregcylTMplusTESandI = Transpose[{phirange, SolEphiregcylTMplusTESandI}];

SolEAllTMplusTESandI =
  N[Abs[Sqrt[((ErhoregcylTESandI + ErhoregcylTMSandI) * Cos[phirange] -
         (EphiregcylTESandI + EphiregcylTMSandI) * Sin[phirange]) ^2 +
      ((ErhoregcylTESandI + ErhoregcylTMSandI) * Sin[phirange] +
         (EphiregcylTESandI + EphiregcylTMSandI) * Cos[phirange]) ^2 +
      (EzregcylTESandI + EzregcylTMSandI) ^2]]];
```

Printed by Wolfram Mathematica Student Edition

64 | PhD Research Garcia_Final_v4.1.nb

```
dataEAllTMplusTESandI = Transpose[{phirange, SolEAllTMplusTESandI}];

(*Scattered Only Solution*)
SolEzregcylTMplusTES = N[Abs[EzregcylTES + EzregcylTMS]];
dataEzregcylTMplusTES = Transpose[{phirange, SolEzregcylTMplusTES}];

SolErhoregcylTMplusTES = N[Abs[ErhoregcylTES + ErhoregcylTMS]];
dataErhoregcylTMplusTES = Transpose[{phirange, SolErhoregcylTMplusTES}];

SolEphiregcylTMplusTES = N[Abs[EphiregcylTES + EphiregcylTMS]];
dataEphiregcylTMplusTES = Transpose[{phirange, SolEphiregcylTMplusTES}];

SolEAllTMplusTES = N[Abs[Sqrt[((ErhoregcylTES + ErhoregcylTMS) * Cos[phirange] -
          (EphiregcylTES + EphiregcylTMS) * Sin[phirange]) ^2 +
       ((ErhoregcylTES + ErhoregcylTMS) * Sin[phirange] + (EphiregcylTES +
            EphiregcylTMS) * Cos[phirange]) ^2 + (EzregcylTES + EzregcylTMS) ^2]]];
dataEAllTMplusTES = Transpose[{phirange, SolEAllTMplusTES}];

(*RCS Solution*)
RCSSolEzregcylTMplusTE = (4 * Pi * (ρ^2)) *
   (((Abs[EzregcylTES + EzregcylTMS]) ^2) / ((Abs[EzregcylTEI + EzregcylTMI]) ^2));
RCSdataEzregcylTMplusTE = Transpose[{phirange, RCSSolEzregcylTMplusTE}];

RCSSolErhoregcylTMplusTE =
  (4 * Pi * (ρ^2)) * (((Abs[ErhoregcylTES + ErhoregcylTMS]) ^2) /
     ((Abs[ErhoregcylTEI + ErhoregcylTMI]) ^2));
RCSdataErhoregcylTMplusTE = Transpose[{phirange, RCSSolErhoregcylTMplusTE}];

RCSSolEphiregcylTMplusTE =
  (4 * Pi * (ρ^2)) * (((Abs[EphiregcylTES + EphiregcylTMS]) ^2) /
     ((Abs[EphiregcylTEI + EphiregcylTMI]) ^2));
RCSdataEphiregcylTMplusTE = Transpose[{phirange, RCSSolEphiregcylTMplusTE}];

RCSSolEregcylTMplusTEAllS = Sqrt[
   ((ErhoregcylTES + ErhoregcylTMS) * Cos[phirange] - (EphiregcylTES + EphiregcylTMS) *
        Sin[phirange]) ^2 + ((ErhoregcylTES + ErhoregcylTMS) * Sin[phirange] +
       (EphiregcylTES + EphiregcylTMS) * Cos[phirange]) ^
     2 + (EzregcylTES + EzregcylTMS) ^2];
RCSSolEregcylTMplusTEAllI = Sqrt[((ErhoregcylTEI + ErhoregcylTMI) * Cos[phirange] -
       (EphiregcylTEI + EphiregcylTMI) * Sin[phirange]) ^2 +
    ((ErhoregcylTEI + ErhoregcylTMI) * Sin[phirange] +
       (EphiregcylTEI + EphiregcylTMI) * Cos[phirange]) ^
     2 + (EzregcylTEI + EzregcylTMI) ^2];
```

Printed by Wolfram Mathematica Student Edition

```
RCSSolEregcylTMplusTEAll = (4 * Pi * (ρ^2)) *
    (((Abs[RCSSolEregcylTMplusTEAllS])^2) / ((Abs[RCSSolEregcylTMplusTEAllI])^2));
RCSdataEregcylTMplusTEAll = Transpose[{phirange, RCSSolEregcylTMplusTEAll}];

(*RCS Solution - in dB*)
RCSdataEzregcylTMplusTEdB =
  Transpose[{phirange, 10 * Log10[RCSSolEzregcylTMplusTE]}];
RCSdataErhoregcylTMplusTEdB =
  Transpose[{phirange, 10 * Log10[RCSSolErhoregcylTMplusTE]}];
RCSdataEphiregcylTMplusTEdB =
  Transpose[{phirange, 10 * Log10[RCSSolEphiregcylTMplusTE]}];
RCSdataEregcylTMplusTEAlldB =
  Transpose[{phirange, 10 * Log10[RCSSolEregcylTMplusTEAll]}];
```

Export Data [Changing Rho Data]

```
data2 = {a, b, ρ2, ρ1, EztempS, EztempI, ErhotempS, ErhotempI, EphitempS, EphitempI,
   EzregcylTES, EzregcylTMS, EzregcylTEI, EzregcylTMI, ErhoregcylTES, ErhoregcylTMS,
   ErhoregcylTEI, ErhoregcylTMI, EphiregcylTES, EphiregcylTMS, EphiregcylTEI,
   EphiregcylTMI, rhorange, ρ, lambda0, θi, ϕ, E0, H0, f0, rhomin, rhomax,
   rhodelta, zpoint, jqty, ϕi, nmin, nmax, nqty, lmin, lmax, lqty, mmin, mmax, lqty,
   mmaxcheck, lmaxcheck, boundarycheck, Eztemp, Ephitemp, Erhotemp, ϕa, ϕivalue};

Export[ToString[StringForm["``.mx", Rhofilename]], data2]
```


Changing Phi Polar Plots

Table of Parameters

```
tablevalues = {{N[E0] "V/m", "-", "-", "-", "-"},
    {N[H0] "A/m", "-", "-", "-", "-"}, {f0 "Hz", "-", "-", "-", "-"},
    {lambda0 "m", "-", "-", "-", "-"}, {a / lambda0 "λ", "-", "-", "-", "-"},
    {b / lambda0 "λ", "-", "-", "-", "-"}, {ρ1 / lambda0 "λ", "-", "-", "-", "-"},
    {ρ2 / lambda0 "λ", "-", "-", "-", "-"}, {"-", N[(phirange[[1]]) * (180 / Pi)] "Deg",
     N[(phirange[[Length[phirange]]]) * (180 / Pi)] "Deg",
     N[(phidelta) * (180 / Pi)] "Deg", Length[phirange]}, {ρ / lambda0 "λ", "-",
     "-", "-", "-"}, {zpoint / a "a" && zpoint / lambda0 "λ", "-", "-", "-", "-"},
    {"-", "-", "-", "-", jqty}, {N[θi * (180 / Pi)] "Deg", "-", "-", "-", "-"},
    {N[ϕi * (180 / Pi)] "Deg", "-", "-", "-", "-"},
    {"-", nmin, nmax, "1", nqty}, {"-", lmin, lmax, "1", lqty},
    {"-", mmin, mmax, "1", lqty}, {mmaxcheck, "-", "-", "-", "-"},
    {lmaxcheck, "-", "-", "-", "-"}, {boundarycheck, "-", "-", "-", "-"}};
tablerowheading = {"E0", "H0", "λ", "Frequency", "a", "b", "ρ1", "ρ2",
   "ϕ range", "ρ (observed)", "z (observed)", "Matching Points", "θi",
   "ϕi", "n", "l", "m", "max allowable m", "max allowable l", "Boundary"};
tablecolheading = {"value", "min", "max", "delta", "Qty of Points"};

ChangingPhiTable = Grid[ArrayFlatten[{{{{" "}}, {tablecolheading}},
     {List /@ tablerowheading, ArrayFlatten[tablevalues]}}],
   ItemStyle → {Bold, 20}, Frame → All, Background → {{LightGray}, {LightGray}}]

Export[ToString[StringForm["``.jpg", PhiTableName]], ChangingPhiTable]
```

Corrugated Cylinder Plots

Plot Ez (Scattered)

```
EzCorrS = ListPolarPlot[dataEzS, Joined → True, PolarGridLines → Automatic,
   PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
   BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
   PlotLabel -> "Ez Corrugated Cylinder (Scattered Field)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzS]}];
```


Plot Ez (Inc + Scattered)

```
EzCorrSandI = ListPolarPlot[dataEz, Joined → True, PolarGridLines → Automatic,
   PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
   BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
   PlotLabel -> "Ez Corrugated Cylinder (Incident + Scattered Field)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[SolEz]}];
```

Plot Erho (Scattered)

```
ErhoCorrS = ListPolarPlot[dataErhoS, Joined → True, PolarGridLines → Automatic,
   PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
   BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
   PlotLabel -> "Eρ Corrugated Cylinder (Scattered Field)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoS]}];
```

Plot Erho (Inc + Scattered)

```
ErhoCorrSandI = ListPolarPlot[dataErho, Joined → True, PolarGridLines → Automatic,
   PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
   BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
   PlotLabel -> "Eρ Corrugated Cylinder (Incident + Scattered Field)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[SolErho]}];
```

Plot Ephi (Scattered)

```
EphiCorrS = ListPolarPlot[dataEphiS, Joined → True, PolarGridLines → Automatic,
   PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
   BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
   PlotLabel -> "Eϕ Corrugated Cylinder (Scattered Field)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiS]}];
```


Plot Ephi (Inc + Scattered)

```
EphiCorrSandI = ListPolarPlot[dataEphi, Joined → True, PolarGridLines → Automatic,
   PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
   BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
   PlotLabel -> "Eϕ Corrugated Cylinder (Incident + Scattered Field)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphi]}];
```

Plot EAll (Scattered)

```
EAllCorrS = ListPolarPlot[dataEAllS, Joined → True, PolarGridLines → Automatic,
   PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
   BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
   PlotLabel -> "ETotal Corrugated Cylinder (Scattered Field)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllS]}];
```

Plot EAll (Inc + Scattered)

```
EAllCorrSandI =
  ListPolarPlot[dataEAllSandI, Joined → True, PolarGridLines → Automatic,
   PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
   BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
   PlotLabel -> "ETotal Corrugated Cylinder (Incident + Scattered Field)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllSandI]}];
```

Summary Corrugated Cylinder Plots

```
GraphicsGrid[
  {{EzCorrS, EzCorrSandI}, {ErhoCorrS, ErhoCorrSandI}, {EphiCorrS, EphiCorrSandI},
   {EAllCorrS, EAllCorrSandI}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TM Plots

Ez (TM Scattered)

```
In[45315]:= EzRegCylTMS = ListPolarPlot[dataEzregcylTMS, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTMS]}];
```

Ez (TM Inc + Scattered)

```
In[45316]:= EzRegCylTMSandI = ListPolarPlot[dataEzregcylTMSandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTMSandI]}];
```

Erho (TM Scattered)

```
In[45317]:= ErhoRegCylTMS = ListPolarPlot[dataErhoregcylTMS, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTMS]}];
```

Erho (TM Inc + Scattered)

```
In[45318]:= ErhoRegCylTMSandI = ListPolarPlot[dataErhoregcylTMSandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTMSandI]}];
```


Ephi (TM Scattered)

```
In[45319]:= EphiRegCylTMS = ListPolarPlot[dataEphiregcylTMS, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TM Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTMS]}];
```

Ephi (TM Inc + Scattered)

```
In[45320]:= EphiRegCylTMSandI = ListPolarPlot[dataEphiregcylTMSandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TM Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTMSandI]}];
```

EAll(TM Scattered)

```
In[45321]:= EAllRegCylTMS = ListPolarPlot[dataEAllTMS, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTMS]}];
```

EAll (TM Inc + Scattered)

```
In[45322]:= EAllRegCylTMSandI = ListPolarPlot[dataEAllTMSandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTMSandI]}];
```

Summary TM Plots

```
GraphicsGrid[{{EzRegCylTMS, EzRegCylTMSandI}, {ErhoRegCylTMS, ErhoRegCylTMSandI},
   {EphiRegCylTMS, EphiRegCylTMSandI}, {EAllRegCylTMS, EAllRegCylTMSandI}},
  Spacings -> {Scaled[0], Scaled[0]}];
```


Smooth Cylinder Plots - TE Plots

Ez (TE Scattered)

```
In[45324]:= EzRegCylTES = ListPolarPlot[dataEzregcylTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTES]}];
```

Ez (TE Inc + Scattered)

```
In[45325]:= EzRegCylTESandI = ListPolarPlot[dataEzregcylTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTESandI]}];
```

Erho (TE Scattered)

```
In[45326]:= ErhoRegCylTES = ListPolarPlot[dataErhoregcylTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTES]}];
```

Erho (TE Inc + Scattered)

```
In[45327]:= ErhoRegCylTESandI = ListPolarPlot[dataErhoregcylTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTESandI]}];
```


Ephi (TE Scattered)

```
In[45328]:= EphiRegCylTES = ListPolarPlot[dataEphiregcylTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTES]}];
```

Ephi (TE Inc + Scattered)

```
In[45329]:= EphiRegCylTESandI = ListPolarPlot[dataEphiregcylTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTESandI]}];
```

EAll (TE Scattered)

```
In[45330]:= EAllRegCylTES = ListPolarPlot[dataEAllTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTES]}];
```

EAll (TE Inc + Scattered)

```
In[45331]:= EAllRegCylTESandI = ListPolarPlot[dataEAllTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTESandI]}];
```

Summary TE Plots

```
GraphicsGrid[{{EzRegCylTES, EzRegCylTESandI},
   {ErhoRegCylTES, ErhoRegCylTESandI}, {EphiRegCylTES, EphiRegCylTESandI},
   {EAllRegCylTES, EAllRegCylTESandI}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TM + TE Plots

Ez (TM+TE Scattered)

```
In[45333]:= EzRegCylTMplusTES = ListPolarPlot[dataEzregcylTMplusTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM + TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTMplusTES]}];
```

Ez (TM+TE Inc + Scattered)

```
In[45334]:= EzRegCylTMplusTESandI = ListPolarPlot[dataEzregcylTMplusTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM + TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTMplusTESandI]}];
```

Erho (TM+TE Scattered)

```
In[45335]:= ErhoRegCylTMplusTES = ListPolarPlot[dataErhoregcylTMplusTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM + TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTMplusTES]}];
```

Erho (TM+TE Inc + Scattered)

```
In[45336]:= ErhoRegCylTMplusTESandI = ListPolarPlot[dataErhoregcylTMplusTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM + TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTMplusTESandI]}];
```


Ephi (TM+TE Scattered)

```
In[45337]:= EphiRegCylTMplusTES = ListPolarPlot[dataEphiregcylTMplusTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TM + TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTMplusTES]}];
```

Ephi (TM+TE Inc + Scattered)

```
In[45338]:= EphiRegCylTMplusTESandI = ListPolarPlot[dataEphiregcylTMplusTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TM + TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTMplusTESandI]}];
```

EAll (TM+TE Scattered)

```
In[45339]:= EAllRegCylTMplusTES = ListPolarPlot[dataEAllTMplusTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM + TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTMplusTES]}];
```

EAll (TM+TE Inc + Scattered)

```
In[45340]:= EAllRegCylTMplusTESandI = ListPolarPlot[dataEAllTMplusTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM + TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTMplusTESandI]}];
```


Summary TM+TE Plots

```
GraphicsGrid[{{EzRegCylTMplusTES, EzRegCylTMplusTESandI},
   {ErhoRegCylTMplusTES, ErhoRegCylTMplusTESandI},
   {EphiRegCylTMplusTES, EphiRegCylTMplusTESandI},
   {EAllRegCylTMplusTES, EAllRegCylTMplusTESandI}},
  Spacings -> {Scaled[0], Scaled[0]}];
```

Compare (Regular vs Corrugated) Plots

TM Compare

```
GraphicsGrid[{{ListPolarPlot[
     {dataEzregcylTMS, dataEzS}, Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
      "Ez TM Scattered Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTMS, SolEzS]}],

    ListPolarPlot[{dataEzregcylTMSandI, dataEz},
     Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Ez TM Incident + Scattered Smooth Cylinder (Blue)
        vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
     PolarAxesOrigin → {0, Max[SolEzregcylTMSandI, SolEz]}]},

   {ListPolarPlot[{dataErhoregcylTMS, dataErhoS},
     Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
      "Eρ TM Scattered Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTMS, SolErhoS]}],

    ListPolarPlot[{dataErhoregcylTMSandI, dataErho},
     Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
```



```
     PlotLabel -> "Eρ TM Incident + Scattered Smooth Cylinder (Blue)
        vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
     PolarAxesOrigin → {0, Max[SolErhoregcylTMSandI, SolErho]}]},

   {ListPolarPlot[{dataEphiregcylTMS, dataEphiS},
     Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
      "Eϕ TM Scattered Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTMS, SolEphiS]}],

    ListPolarPlot[{dataEphiregcylTMSandI, dataEphi},
     Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eϕ TM Incident + Scattered Smooth Cylinder (Blue)
        vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
     PolarAxesOrigin → {0, Max[SolEphiregcylTMSandI, SolEphi]}]},

   {ListPolarPlot[{dataEAllTMS, dataEAllS}, Joined → True,
     PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
     PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "ETotal TM Scattered Smooth
        Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTMS, SolEAllS]}],

    ListPolarPlot[{dataEAllTMSandI, dataEAllSandI},
     Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "ETotal TM Incident + Scattered Smooth Cylinder (Blue)
        vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
     PolarAxesOrigin → {0, Max[SolEAllTMSandI, SolEAllSandI]}]}
  }];
```

TE Compare

```
In[45343]:= GraphicsGrid[{{ListPolarPlot[
     {dataEzregcylTES, dataEzS}, Joined → True, PolarGridLines → Automatic,
```

```
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
   "E_z TE Scattered Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTES, SolEzS]}],

 ListPolarPlot[{dataEzregcylTESandI, dataEz},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "E_z TE Incident + Scattered Smooth Cylinder (Blue)
     vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEzregcylTESandI, SolEz]}]},

{ListPolarPlot[{dataErhoregcylTES, dataErhoS},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
   "E_ρ TE Scattered Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTES, SolErhoS]}],

 ListPolarPlot[{dataErhoregcylTESandI, dataErho},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "E_ρ TE Incident + Scattered Smooth Cylinder (Blue)
     vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolErhoregcylTESandI, SolErho]}]},

{ListPolarPlot[{dataEphiregcylTES, dataEphiS},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
   "E_ϕ TE Scattered Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTES, SolEphiS]}],

 ListPolarPlot[{dataEphiregcylTESandI, dataEphi},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
```



```
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "E_ϕ TE Incident + Scattered Smooth Cylinder (Blue)
     vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEphiregcylTESandI, SolEphi]}]},

 {ListPolarPlot[{dataEAllTES, dataEAllS}, Joined → True,
  PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
  PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
  PlotMarkers -> None, PlotLabel -> "E_Total TE Scattered Smooth
     Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTES, SolEAllS]}],

 ListPolarPlot[{dataEAllTESandI, dataEAllSandI},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "E_Total TE Incident + Scattered Smooth Cylinder (Blue)
     vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEAllTESandI, SolEAllSandI]}]}
}];
```

TM+TE Compare

```
In[45344]:= GraphicsGrid[{{ListPolarPlot[
     {dataEzregcylTMplusTES, dataEzS}, Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "E_z TM+TE Scattered Smooth Cylinder (Blue) vs
        Corrugated Cylinder (Red, --)", ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
     PolarAxesOrigin → {0, Max[SolEzregcylTMplusTES, SolEzS]}],

    ListPolarPlot[{dataEzregcylTMplusTESandI, dataEz},
     Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "E_z TM+TE Incident + Scattered Smooth Cylinder (Blue)
        vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
     PolarAxesOrigin → {0, Max[SolEzregcylTMplusTESandI, SolEz]}]},
```



```
{ListPolarPlot[{dataErhoregcylTMplusTES, dataErhoS},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "E_ρ TM+TE Scattered Smooth Cylinder (Blue) vs
     Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolErhoregcylTMplusTES, SolErhoS]}],

 ListPolarPlot[{dataErhoregcylTMplusTESandI, dataErho},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "E_ρ TM+TE Incident + Scattered Smooth Cylinder (Blue)
     vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolErhoregcylTMplusTESandI, SolErho]}]},

{ListPolarPlot[{dataEphiregcylTMplusTES, dataEphiS},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "E_ϕ TM+TE Scattered Smooth Cylinder (Blue) vs
     Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEphiregcylTMplusTES, SolEphiS]}],

 ListPolarPlot[{dataEphiregcylTMplusTESandI, dataEphi},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "E_ϕ TM+TE Incident + Scattered Smooth Cylinder (Blue)
     vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEphiregcylTMplusTESandI, SolEphi]}]},

{ListPolarPlot[{dataEAllTMplusTES, dataEAllS},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "E_Total TM+TE Scattered Smooth Cylinder (Blue)
     vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
```



```
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEAllTMplusTES, SolEAllS]}],

 ListPolarPlot[{dataEAllTMplusTESandI, dataEAllSandI},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "E_Total TM+TE Incident + Scattered Smooth Cylinder (Blue)
     vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEAllTMplusTESandI, SolEAllSandI]}]}
}];
```

TM, TE, and TM+TE Compare

```
In[45345]:= PhiPolarPlots =
  GraphicsGrid[{{ListPolarPlot[{dataEzregcylTMS, dataEzS}, Joined → True,
     PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
     PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "E_z TM Scattered (V/m) Smooth
        Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTMS, SolEzS]}],
    "E_z TE Does Not Exist",
    ListPolarPlot[{dataEzregcylTMplusTES, dataEzS},
     Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "E_z TM+TE (V/m) Scattered Smooth Cylinder (Blue)
        vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
     PolarAxesOrigin → {0, Max[SolEzregcylTMplusTES, SolEzS]}]},


   {ListPolarPlot[{dataEzregcylTMSandI, dataEz},
     Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "E_z TM Incident + Scattered (V/m) Smooth Cylinder (Blue)
        vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
     PolarAxesOrigin → {0, Max[SolEzregcylTMSandI, SolEz]}],
```

```
"Ez TE Does Not Exist",
ListPolarPlot[{dataEzregcylTMplusTESandI, dataEz},
 Joined → True, PolarGridLines → Automatic,
 PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
 BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
 PlotLabel -> "Ez TM+TE Incident + Scattered (V/m) Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
 PolarAxesOrigin → {0, Max[SolEzregcylTMplusTESandI, SolEz]}]},


{ListPolarPlot[{dataErhoregcylTMS, dataErhoS},
 Joined → True, PolarGridLines → Automatic,
 PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
 BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
 PlotLabel -> "Eρ TM Scattered (V/m) Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
 PolarAxesOrigin → {0, Max[SolErhoregcylTMS, SolErhoS]}],
ListPolarPlot[{dataErhoregcylTES, dataErhoS}, Joined → True,
 PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
 PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
 PlotMarkers -> None, PlotLabel -> "Eρ TE Scattered (V/m) Smooth
    Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
 ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTES, SolErhoS]}],
ListPolarPlot[{dataErhoregcylTMplusTES, dataErhoS}, Joined → True,
 PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
 PlotStyle -> {Blue, {Red, Dashed}},
 BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
 PlotLabel -> "Eρ TM+TE Scattered (V/m) Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
 PolarAxesOrigin → {0, Max[SolErhoregcylTMplusTES, SolErhoS]}]},

{ListPolarPlot[{dataErhoregcylTMSandI, dataErho},
 Joined → True, PolarGridLines → Automatic,
 PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
 BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
 PlotLabel -> "Eρ TM Incident + Scattered (V/m) Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
 PolarAxesOrigin → {0, Max[SolErhoregcylTMSandI, SolErho]}],
```



```
ListPolarPlot[{dataErhoregcylTESandI, dataErho}, Joined → True,
 PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
 PlotStyle -> {Blue, {Red, Dashed}},
 BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
 PlotLabel -> "Eρ TE Incident + Scattered (V/m) Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
 PolarAxesOrigin → {0, Max[SolErhoregcylTESandI, SolErho]}],
ListPolarPlot[{dataErhoregcylTMplusTESandI, dataErho},
 Joined → True, PolarGridLines → Automatic,
 PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
 BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
 PlotLabel -> "Eρ TM+TE Incident + Scattered (V/m) Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
 PolarAxesOrigin → {0, Max[SolErhoregcylTMplusTESandI, SolErho]}]},


{ListPolarPlot[{dataEphiregcylTMS, dataEphiS},
 Joined → True, PolarGridLines → Automatic,
 PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
 BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
 PlotLabel -> "Eφ TM Scattered (V/m) Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
 PolarAxesOrigin → {0, Max[SolEphiregcylTMS, SolEphiS]}],
ListPolarPlot[{dataEphiregcylTES, dataEphiS}, Joined → True,
 PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
 PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
 PlotMarkers -> None, PlotLabel -> "Eφ TE Scattered (V/m) Smooth
    Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
 ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTES, SolEphiS]}],
ListPolarPlot[{dataEphiregcylTMplusTES, dataEphiS}, Joined → True,
 PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
 PlotStyle -> {Blue, {Red, Dashed}},
 BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
 PlotLabel -> "Eφ TM+TE Scattered (V/m) Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
 PolarAxesOrigin → {0, Max[SolEphiregcylTMplusTES, SolEphiS]}]},
```



```
{ListPolarPlot[{dataEphiregcylTMSandI, dataEphi},
 Joined → True, PolarGridLines → Automatic,
 PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
 BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
 PlotLabel -> "Eφ TM Incident + Scattered (V/m) Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
 PolarAxesOrigin → {0, Max[SolEphiregcylTMSandI, SolEphi]}],
ListPolarPlot[{dataEphiregcylTESandI, dataEphi}, Joined → True,
 PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
 PlotStyle -> {Blue, {Red, Dashed}},
 BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
 PlotLabel -> "Eφ TE Incident + Scattered (V/m) Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
 PolarAxesOrigin → {0, Max[SolEphiregcylTESandI, SolEphi]}],
ListPolarPlot[{dataEphiregcylTMplusTESandI, dataEphi},
 Joined → True, PolarGridLines → Automatic,
 PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
 BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
 PlotLabel -> "Eφ TM+TE Incident + Scattered (V/m) Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
 PolarAxesOrigin → {0, Max[SolEphiregcylTMplusTESandI, SolEphi]}]},


{ListPolarPlot[{dataEAllTMS, dataEAllS}, Joined → True,
 PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
 PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
 PlotMarkers -> None, PlotLabel -> "ETotal TM Scattered (V/m) Smooth
    Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
 ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTMS, SolEAllS]}],
ListPolarPlot[{dataEAllTES, dataEAllS}, Joined → True,
 PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
 PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
 PlotMarkers -> None, PlotLabel -> "ETotal TE Scattered (V/m) Smooth
    Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
 ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTES, SolEAllS]}],
ListPolarPlot[{dataEAllTMplusTES, dataEAllS}, Joined → True,
 PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
 PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
```



```
 PlotMarkers -> None, PlotLabel -> "ETotal TM+TE Scattered (V/m)
    Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
 ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
 PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTMplusTES, SolEAllS]}]},


{ListPolarPlot[{dataEAllTMSandI, dataEAllSandI},
 Joined → True, PolarGridLines → Automatic,
 PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
 BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
 PlotLabel -> "ETotal TM Incident + Scattered (V/m) Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
 PolarAxesOrigin → {0, Max[SolEAllTMSandI, SolEAllSandI]}],
ListPolarPlot[{dataEAllTESandI, dataEAllSandI}, Joined → True,
 PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
 PlotStyle -> {Blue, {Red, Dashed}},
 BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
 PlotLabel -> "ETotal TE Incident + Scattered (V/m) Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
 PolarAxesOrigin → {0, Max[SolEAllTESandI, SolEAllSandI]}],

ListPolarPlot[{dataEAllTMplusTESandI, dataEAllSandI},
 Joined → True, PolarGridLines → Automatic,
 PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
 BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
 PlotLabel -> "ETotal TM+TE Incident + Scattered (V/m) Smooth Cylinder
    (Blue) vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
 BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
 PolarAxesOrigin → {0, Max[SolEAllTMplusTESandI, SolEAllSandI]}]
 }
}]
Export[ToString[StringForm["``.jpg", PhiPolarPlotsName]], PhiPolarPlots]
```

Changing Phi RCS Polar Plots

Corrugated Cylinder Plots

Plot Ez

```
RCSEzCorr = ListPolarPlot[RCSdataEz, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Ez Corrugated Cylinter (RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEz]}];
```

Plot Erho

```
RCSErhoCorr = ListPolarPlot[RCSdataErho, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Eρ Corrugated Cylinter (RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolErho]}];
```

Plot Ephi

```
RCSEphiCorr = ListPolarPlot[RCSdataEphi, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Eϕ Corrugated Cylinter (RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEphi]}];
```

Plot EAll

```
RCSEAllCorr = ListPolarPlot[RCSdataEAll, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "ETotal Corrugated Cylinter (RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEAll]}];
```

Summary Corrugated Cylinder Plots

```
GraphicsGrid[{{RCSEzCorr}, {RCSErhoCorr}, {RCSEphiCorr}, {RCSEAllCorr}},
  Spacings -> {Scaled[0], Scaled[0]}];
```


Smooth Cylinder Plots - TM Plots

Ez (TM RCS)

```
RCSEzRegCylTM = ListPolarPlot[RCSdataEzregcylTM, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Ez Smooth Cylinder (TM RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEzregcylTM]}];
```

Erho (TM RCS)

```
In[45353]:= RCSErhoRegCylTM = ListPolarPlot[RCSdataErhoregcylTM, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Eρ Smooth Cylinder (TM RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolErhoregcylTM]}];
```

Ephi (TM RCS)

```
In[45354]:= RCSEphiRegCylTM = ListPolarPlot[RCSdataEphiregcylTM, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Eϕ Smooth Cylinder (TM RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEphiregcylTM]}];
```

EAll (TM RCS)

```
In[45355]:= RCSEAllRegCylTM = ListPolarPlot[RCSdataEregcylTMAll, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "ETotal Smooth Cylinder (TM RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEregcylTMAll]}];
```

Summary TM Plots

```
In[45356]:= GraphicsGrid[{{RCSEzRegCylTM}, {RCSErhoRegCylTM}, {RCSEphiRegCylTM},
   {RCSEAllRegCylTM}}, Spacings -> {Scaled[0], Scaled[0]}];
```


Smooth Cylinder Plots - TE Plots

Ez (TE RCS) - DOES NOT EXIST

Erho (TE RCS)

```
In[45357]:= RCSErhoRegCylTE = ListPolarPlot[RCSdataErhoregcylTE, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Eρ Smooth Cylinder (TE RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolErhoregcylTE]}];
```

Ephi (TE RCS)

```
In[45358]:= RCSEphiRegCylTE = ListPolarPlot[RCSdataEphiregcylTE, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Eϕ Smooth Cylinder (TE RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEphiregcylTE]}];
```

EAll (TE RCS)

```
In[45359]:= RCSEAllRegCylTE = ListPolarPlot[RCSdataEregcylTEAll, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "ETotal Smooth Cylinder (TE RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEregcylTEAll]}];
```

Summary TE Plots

```
In[45360]:= GraphicsGrid[{{RCSErhoRegCylTE}, {RCSEphiRegCylTE}, {RCSEAllRegCylTE}},
   Spacings -> {Scaled[0], Scaled[0]}];
```


Smooth Cylinder Plots - TM + TE Plots

Ez (TM + TE RCS)

```
In[45361]:= RCSEzRegCylTMplusTE = ListPolarPlot[RCSdataEzregcylTMplusTE, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Ez Smooth Cylinder (TM + TE RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEzregcylTMplusTE]}];
```

Erho (TM + TE RCS)

```
In[45362]:= RCSErhoRegCylTMplusTE = ListPolarPlot[RCSdataErhoregcylTMplusTE, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Eρ Smooth Cylinder (TM + TE RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolErhoregcylTMplusTE]}];
```

Ephi (TM + TE RCS)

```
In[45363]:= RCSEphiRegCylTMplusTE = ListPolarPlot[RCSdataEphiregcylTMplusTE, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Eϕ Smooth Cylinder (TM + TE RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEphiregcylTMplusTE]}];
```

EAll (TM + TE RCS)

```
In[45364]:= RCSEAllRegCylTMplusTE = ListPolarPlot[RCSdataEregcylTMplusTEAll, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
   PlotLabel -> "ETotal Smooth Cylinder (TM + TE RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEregcylTMplusTEAll]}];
```

Summary TM + TE Plots

```
GraphicsGrid[
  {{RCSEzRegCylTMplusTE}, {RCSErhoRegCylTMplusTE}, {RCSEphiRegCylTMplusTE},
   {RCSEAllRegCylTMplusTE}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Compare (Regular vs Corrugated) Plots

TM, TE, and TM+TE Compare

```
PhiRCSPolarPlots =
  GraphicsGrid[{{ListPolarPlot[{RCSdataEzregcylTM, RCSdataEz}, Joined → True,
      PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
      PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
       "Ez TM RCS (m²) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEzregcylTM, RCSSolEz]}]
     ,
     "Ez TE Does Not Exist"
     ,
     ListPolarPlot[{RCSdataEzregcylTMplusTE, RCSdataEz},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "Ez TM+TE RCS (m²) Smooth Cylinder (Blue) vs
         Corrugated Cylinder (Red, --)", ImageSize -> 700,
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
      PolarAxesOrigin → {0, Max[RCSSolEzregcylTMplusTE, RCSSolEz]}]},

    {ListPolarPlot[{RCSdataErhoregcylTM, RCSdataErho},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
       "Eρ TM RCS (m²) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolErhoregcylTM, RCSSolErho]}]
     ,
     ListPolarPlot[{RCSdataErhoregcylTE, RCSdataErho},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
```



```
       "Eρ TE RCS (m²) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolErhoregcylTE, RCSSolErho]}]
     ,
     ListPolarPlot[{RCSdataErhoregcylTMplusTE, RCSdataErho},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "Eρ TM+TE RCS (m²) Smooth Cylinder (Blue) vs
         Corrugated Cylinder (Red, --)", ImageSize -> 700,
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
      PolarAxesOrigin → {0, Max[RCSSolErhoregcylTMplusTE, RCSSolErho]}]
    },

    {ListPolarPlot[{RCSdataEphiregcylTM, RCSdataEphi},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
       "Eϕ TM RCS (m²) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEphiregcylTM, RCSSolEphi]}]
     ,
     ListPolarPlot[{RCSdataEphiregcylTE, RCSdataEphi},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
       "Eϕ TE RCS (m²) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEphiregcylTE, RCSSolEphi]}]
     ,
     ListPolarPlot[{RCSdataEphiregcylTMplusTE, RCSdataEphi},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "Eϕ TM+TE RCS (m²) Smooth Cylinder (Blue) vs
         Corrugated Cylinder (Red, --)", ImageSize -> 700,
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
      PolarAxesOrigin → {0, Max[RCSSolEphiregcylTMplusTE, RCSSolEphi]}]
    },

    {ListPolarPlot[{RCSdataEregcylTMAll, RCSdataEAll},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
```



```
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "ETotal TM RCS (m²) Smooth Cylinder (Blue) vs
         Corrugated Cylinder (Red, --)", ImageSize -> 700,
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
      PolarAxesOrigin → {0, Max[RCSSolEregcylTMAll, RCSSolEAll]}]
     ,
     ListPolarPlot[{RCSdataEregcylTEAll, RCSdataEAll},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "ETotal TE RCS (m²) Smooth Cylinder (Blue) vs
         Corrugated Cylinder (Red, --)", ImageSize -> 700,
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
      PolarAxesOrigin → {0, Max[RCSSolEregcylTEAll, RCSSolEAll]}]
     ,
     ListPolarPlot[{RCSdataEregcylTMplusTEAll, RCSdataEAll},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "ETotal TM+TE RCS (m²) Smooth Cylinder (Blue)
         vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
      PolarAxesOrigin → {0, Max[RCSSolEregcylTMplusTEAll, RCSSolEAll]}]
    }

   }]
```

```
Export[ToString[StringForm["``.jpg", PhiRCSPolarPlotsName]], PhiRCSPolarPlots];
```

Changing Phi RCS dB Polar Plots

Corrugated Cylinder Plots

Plot Ez

```
RCSEzCorrdB = ListPolarPlot[RCSdataEzdB, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Ez Corrugated Cylinder (RCS dBsm)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEz]]]}];
```


Plot Erho

```
RCSErhoCorrdB = ListPolarPlot[RCSdataErhodB, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Eρ Corrugated Cylinder (RCS dBsm)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolErho]]]}];
```

Plot Ephi

```
RCSEphiCorrdB = ListPolarPlot[RCSdataEphidB, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Eϕ Corrugated Cylinder (RCS dBsm)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEphi]]]}];
```

Plot EAll

```
RCSEAllCorrdB = ListPolarPlot[RCSdataEAlldB, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "ETotal Corrugated Cylinder (RCS dBsm)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEAll]]]}];
```

Summary Corrugated Cylinder Plots

```
GraphicsGrid[{{RCSEzCorrdB}, {RCSErhoCorrdB}, {RCSEphiCorrdB}, {RCSEAllCorrdB}},
  Spacings -> {Scaled[0], Scaled[0]}];
```

PhD Research Garcia_Final_v4.1.nb | 93

Smooth Cylinder Plots - TM Plots

Ez (TM RCS)

```
In[45375]:= RCSEzRegCylTMdB = ListPolarPlot[RCSdataEzregcylTMdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEzregcylTM]]]}];
```

Erho (TM RCS)

```
In[45376]:= RCSErhoRegCylTMdB = ListPolarPlot[RCSdataErhoregcylTMdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolErhoregcylTM]]]}];
```

Ephi (TM RCS)

```
In[45377]:= RCSEphiRegCylTMdB = ListPolarPlot[RCSdataEphiregcylTMdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TM RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEphiregcylTM]]]}];
```

EAll (TM RCS)

```
In[45378]:= RCSEAllRegCylTMdB = ListPolarPlot[RCSdataEregcylTMAlldB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEregcylTMAll]]]}];
```

Summary TM Plots

```
In[45379]:= GraphicsGrid[{{RCSEzRegCylTMdB}, {RCSErhoRegCylTMdB}, {RCSEphiRegCylTMdB},
    {RCSEAllRegCylTMdB}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Printed by Wolfram Mathematica Student Edition

94 | PhD Research Garcia_Final_v4.1.nb

Smooth Cylinder Plots - TE Plots

Ez (TE RCS) - DOES NOT EXIST

Erho (TE RCS)

```
In[45380]:= RCSErhoRegCylTEdB = ListPolarPlot[RCSdataErhoregcylTEdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TE RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolErhoregcylTE]]]}];
```

Ephi (TE RCS)

```
In[45381]:= RCSEphiRegCylTEdB = ListPolarPlot[RCSdataEphiregcylTEdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TE RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEphiregcylTE]]]}];
```

EAll (TE RCS)

```
In[45382]:= RCSEAllRegCylTEdB = ListPolarPlot[RCSdataEregcylTEAlldB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TE RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEregcylTEAll]]]}];
```

Summary TE Plots

```
In[45383]:= GraphicsGrid[{{RCSErhoRegCylTEdB}, {RCSEphiRegCylTEdB}, {RCSEAllRegCylTEdB}},
   Spacings -> {Scaled[0], Scaled[0]}];
```

Printed by Wolfram Mathematica Student Edition

PhD Research Garcia_Final_v4.1.nb | 95

Smooth Cylinder Plots - TM + TE Plots

Ez (TM + TE RCS)

```
In[45384]:= RCSEzRegCylTMplusTEdB = ListPolarPlot[RCSdataEzregcylTMplusTEdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM + TE RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEzregcylTMplusTE]]]}];
```

Erho (TM + TE RCS)

```
In[45385]:= RCSErhoRegCylTMplusTEdB = ListPolarPlot[RCSdataErhoregcylTMplusTEdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM + TE RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolErhoregcylTMplusTE]]]}];
```

Ephi (TM + TE RCS)

```
In[45386]:= RCSEphiRegCylTMplusTEdB = ListPolarPlot[RCSdataEphiregcylTMplusTEdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TM + TE RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEphiregcylTMplusTE]]]}];
```

EAll (TM + TE RCS)

```
In[45387]:= RCSEAllRegCylTMplusTEdB = ListPolarPlot[RCSdataEregcylTMplusTEAlldB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM + TE RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEregcylTMplusTEAll]]]}];
```

Printed by Wolfram Mathematica Student Edition

96 | PhD Research Garcia_Final_v4.1.nb

Summary TM + TE Plots

```
In[45388]:= GraphicsGrid[
   {{RCSEzRegCylTMplusTEdB}, {RCSErhoRegCylTMplusTEdB}, {RCSEphiRegCylTMplusTEdB},
    {RCSEAllRegCylTMplusTEdB}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Compare (Regular vs Corrugated) Plots

TM, TE, and TM+TE Compare

```
In[45389]:= PhiRCSdBPolarPlots =
  GraphicsGrid[{{ListPolarPlot[{RCSdataEzregcylTMdB, RCSdataEzdB}, Joined → True,
      PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
      PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
      PlotMarkers -> None, PlotLabel -> "Ez TM RCS (dBsm) Smooth
         Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin →
       {0, Max[Abs[10 * Log10[RCSSolEzregcylTM]], Abs[10 * Log10[RCSSolEz]]]}]
     ,
     "Ez TE Does Not Exist"
     ,
     ListPolarPlot[{RCSdataEzregcylTMplusTEdB, RCSdataEzdB}, Joined → True,
      PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
      PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
      PlotMarkers -> None, PlotLabel -> "Ez TM+TE RCS (dBsm) Smooth
         Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0,
        Max[Abs[10 * Log10[RCSSolEzregcylTMplusTE]], Abs[10 * Log10[RCSSolEz]]]}]},

    {ListPolarPlot[{RCSdataErhoregcylTMdB, RCSdataErhodB}, Joined → True,
      PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
      PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
      PlotMarkers -> None, PlotLabel -> "Eρ TM RCS (dBsm) Smooth
         Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin →
       {0, Max[Abs[10 * Log10[RCSSolErhoregcylTM]], Abs[10 * Log10[RCSSolErho]]]}]
     ,
     ListPolarPlot[{RCSdataErhoregcylTEdB, RCSdataErhodB}, Joined → True,
      PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
```

Printed by Wolfram Mathematica Student Edition

```
  PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
  PlotMarkers -> None, PlotLabel -> "Eρ TE RCS (dBsm) Smooth
     Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  PolarAxes → True, PolarAxesOrigin →
   {0, Max[Abs[10 * Log10[RCSSolErhoregcylTE]], Abs[10 * Log10[RCSSolErho]]]}]
 ,
 ListPolarPlot[{RCSdataErhoregcylTMplusTEdB, RCSdataErhodB}, Joined → True,
  PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
  PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
  PlotMarkers -> None, PlotLabel -> "Eρ TM+TE RCS (dBsm) Smooth
     Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  PolarAxes → True, PolarAxesOrigin → {0, Max[
     Abs[10 * Log10[RCSSolErhoregcylTMplusTE]], Abs[10 * Log10[RCSSolErho]]]}]
},

{ListPolarPlot[{RCSdataEphiregcylTMdB, RCSdataEphidB}, Joined → True,
  PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
  PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
  PlotMarkers -> None, PlotLabel -> "Eϕ TM RCS (dBsm) Smooth
     Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  PolarAxes → True, PolarAxesOrigin →
   {0, Max[Abs[10 * Log10[RCSSolEphiregcylTM]], Abs[10 * Log10[RCSSolEphi]]]}]
 ,
 ListPolarPlot[{RCSdataEphiregcylTEdB, RCSdataEphidB}, Joined → True,
  PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
  PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
  PlotMarkers -> None, PlotLabel -> "Eϕ TE RCS (dBsm) Smooth
     Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  PolarAxes → True, PolarAxesOrigin →
   {0, Max[Abs[10 * Log10[RCSSolEphiregcylTE]], Abs[10 * Log10[RCSSolEphi]]]}]
 ,
 ListPolarPlot[{RCSdataEphiregcylTMplusTEdB, RCSdataEphidB}, Joined → True,
  PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
  PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
  PlotMarkers -> None, PlotLabel -> "Eϕ TM+TE RCS (dBsm) Smooth
     Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  PolarAxes → True, PolarAxesOrigin → {0, Max[
     Abs[10 * Log10[RCSSolEphiregcylTMplusTE]], Abs[10 * Log10[RCSSolEphi]]]}]
```



```
 },

 {ListPolarPlot[{RCSdataEregcylTMAlldB, RCSdataEAlldB}, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "ETotal TM RCS (dBsm) Smooth
      Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin →
    {0, Max[Abs[10 * Log10[RCSSolEregcylTMAll]], Abs[10 * Log10[RCSSolEAll]]]}]
  ,
  ListPolarPlot[{RCSdataEregcylTEAlldB, RCSdataEAlldB}, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "ETotal TE RCS (dBsm) Smooth
      Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin →
    {0, Max[Abs[10 * Log10[RCSSolEregcylTEAll]], Abs[10 * Log10[RCSSolEAll]]]}]
  ,
  ListPolarPlot[{RCSdataEregcylTMplusTEAlldB, RCSdataEAlldB}, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "ETotal TM+TE RCS (dBsm) Smooth
      Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[
      Abs[10 * Log10[RCSSolEregcylTMplusTEAll]], Abs[10 * Log10[RCSSolEAll]]]}]
 }

}]
Export[ToString[StringForm["``.jpg", PhiRCSdBPolarPlotsName]], PhiRCSdBPolarPlots]
```

Changing Phi RCS dB XY Plots

Corrugated Cylinder Plots

Plot Ez


```
In[45391]:= RCSEzCorrdBxy = ListLinePlot[RCSdataEzdB,
     PlotRange -> {{0, 2 Pi}, {Min[10 * Log10[RCSSolEz]], Max[10 * Log10[RCSSolEz]]}},
     PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Ez Corrugated Cylinder (RCS dBsm)",
     Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
     GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot Erho

```
In[45392]:= RCSErhoCorrdBxy = ListLinePlot[RCSdataErhodB, PlotRange ->
      {{0, 2 Pi}, {Min[10 * Log10[RCSSolErho]], Max[10 * Log10[RCSSolErho]]}},
     PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Eρ Corrugated Cylinder (RCS dBsm)",
     Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
     GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot Ephi

```
In[45393]:= RCSEphiCorrdBxy = ListLinePlot[RCSdataEphidB, PlotRange ->
      {{0, 2 Pi}, {Min[10 * Log10[RCSSolEphi]], Max[10 * Log10[RCSSolEphi]]}},
     PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Eϕ Corrugated Cylinder (RCS dBsm)",
     Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
     GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot EAll

```
In[45394]:= RCSEAllCorrdBxy = ListLinePlot[RCSdataEAlldB, PlotRange ->
      {{0, 2 Pi}, {Min[10 * Log10[RCSSolEAll]], Max[10 * Log10[RCSSolEAll]]}},
     PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "ETotal Corrugated Cylinder (RCS dBsm)",
     Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
     GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary Corrugated Cylinder Plots

```
In[45395]:= GraphicsGrid[{{RCSEzCorrdBxy}, {RCSErhoCorrdBxy}, {RCSEphiCorrdBxy},
     {RCSEAllCorrdBxy}}, Spacings -> {Scaled[0], Scaled[0]}];
```


Smooth Cylinder Plots - TM Plots

Ez (TM RCS)

```
In[45396]:= RCSEzRegCylTMdBxy = ListLinePlot[RCSdataEzregcylTMdB, PlotRange -> {{0, 2 Pi},
       {Min[10 * Log10[RCSSolEzregcylTM]], Max[10 * Log10[RCSSolEzregcylTM]]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Ez Smooth Cylinder (TM RCS dBsm)",
     Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
     GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TM RCS)

```
In[45397]:= RCSErhoRegCylTMdBxy = ListLinePlot[RCSdataErhoregcylTMdB, PlotRange -> {{0, 2 Pi},
       {Min[10 * Log10[RCSSolErhoregcylTM]], Max[10 * Log10[RCSSolErhoregcylTM]]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Eρ Smooth Cylinder (TM RCS dBsm)",
     Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
     GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TM RCS)

```
In[45398]:= RCSEphiRegCylTMdBxy = ListLinePlot[RCSdataEphiregcylTMdB, PlotRange -> {{0, 2 Pi},
       {Min[10 * Log10[RCSSolEphiregcylTM]], Max[10 * Log10[RCSSolEphiregcylTM]]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Eϕ Smooth Cylinder (TM RCS dBsm)",
     Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
     GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TM RCS)

```
In[45399]:= RCSEAllRegCylTMdBxy = ListLinePlot[RCSdataEregcylTMAlldB, PlotRange -> {{0, 2 Pi},
       {Min[10 * Log10[RCSSolEregcylTMAll]], Max[10 * Log10[RCSSolEregcylTMAll]]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "ETotal Smooth Cylinder (TM RCS dBsm)",
     Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
     GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary TM Plots

```
In[45400]:= GraphicsGrid[{{RCSEzRegCylTMdBxy}, {RCSErhoRegCylTMdBxy}, {RCSEphiRegCylTMdBxy},
    {RCSEAllRegCylTMdBxy}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TE Plots

Ez (TE RCS) - DOES NOT EXIST

Erho (TE RCS)

```
In[45401]:= RCSErhoRegCylTEdBxy = ListLinePlot[RCSdataErhoregcylTEdB, PlotRange -> {{0, 2 Pi},
      {Min[10 * Log10[RCSSolErhoregcylTE]], Max[10 * Log10[RCSSolErhoregcylTE]]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Eρ Smooth Cylinder (TE RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TE RCS)

```
In[45402]:= RCSEphiRegCylTEdBxy = ListLinePlot[RCSdataEphiregcylTEdB, PlotRange -> {{0, 2 Pi},
      {Min[10 * Log10[RCSSolEphiregcylTE]], Max[10 * Log10[RCSSolEphiregcylTE]]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Eϕ Smooth Cylinder (TE RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TE RCS)

```
In[45403]:= RCSEAllRegCylTEdBxy = ListLinePlot[RCSdataEregcylTEAlldB, PlotRange -> {{0, 2 Pi},
      {Min[10 * Log10[RCSSolEregcylTEAll]], Max[10 * Log10[RCSSolEregcylTEAll]]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TE RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Summary TE Plots

```
In[45404]:= GraphicsGrid[{{RCSErhoRegCylTEdBxy}, {RCSEphiRegCylTEdBxy},
    {RCSEAllRegCylTEdBxy}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TM + TE Plots

Ez (TM + TE RCS)

```
In[45405]:= RCSEzRegCylTMplusTEdBxy = ListLinePlot[RCSdataEzregcylTMplusTEdB,
    PlotRange -> {{0, 2 Pi}, {Min[10 * Log10[RCSSolEzregcylTMplusTE]],
       Max[10 * Log10[RCSSolEzregcylTMplusTE]]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM+TE RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TM + TE RCS)

```
In[45406]:= RCSErhoRegCylTMplusTEdBxy = ListLinePlot[RCSdataErhoregcylTMplusTEdB,
    PlotRange -> {{0, 2 Pi}, {Min[10 * Log10[RCSSolErhoregcylTMplusTE]],
       Max[10 * Log10[RCSSolErhoregcylTMplusTE]]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM+TE RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TM + TE RCS)

```
In[45407]:= RCSEphiRegCylTMplusTEdBxy = ListLinePlot[RCSdataEphiregcylTMplusTEdB,
    PlotRange -> {{0, 2 Pi}, {Min[10 * Log10[RCSSolEphiregcylTMplusTE]],
       Max[10 * Log10[RCSSolEphiregcylTMplusTE]]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TM+TE RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


EAll (TM + TE RCS)

```
In[45408]:= RCSEAllRegCylTMplusTEdBxy = ListLinePlot[RCSdataEregcylTMplusTEAlldB,
    PlotRange -> {{0, 2 Pi}, {Min[10 * Log10[RCSSolEregcylTMplusTEAll]],
       Max[10 * Log10[RCSSolEregcylTMplusTEAll]]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM+TE RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary TM + TE Plots

```
In[45409]:= GraphicsGrid[{{RCSEzRegCylTMplusTEdBxy}, {RCSErhoRegCylTMplusTEdBxy},
    {RCSEphiRegCylTMplusTEdBxy}, {RCSEAllRegCylTMplusTEdBxy}},
   Spacings -> {Scaled[0], Scaled[0]}];
```

Compare (Regular vs Corrugated) Plots

TM, TE, and TM+TE Compare

```
PhiRCSdBXYPlots =
 GraphicsGrid[{{Show[{RCSEzRegCylTMdBxy, RCSEzCorrdBxy}, PlotRange ->
      {{0, 2 Pi}, {Min[10 * Log10[RCSSolEzregcylTM], 10 * Log10[RCSSolEz]],
        Max[10 * Log10[RCSSolEzregcylTM], 10 * Log10[RCSSolEz]]}}, PlotLabel ->
      "Ez TM RCS (dBsm) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
    ,
    "Ez TE RCS dB Does Not Exist"
    ,
    Show[{RCSEzRegCylTMplusTEdBxy, RCSEzCorrdBxy}, PlotRange ->
      {{0, 2 Pi}, {Min[10 * Log10[RCSSolEzregcylTMplusTE], 10 * Log10[RCSSolEz]],
        Max[10 * Log10[RCSSolEzregcylTMplusTE], 10 * Log10[RCSSolEz]]}},
     PlotLabel -> "Ez TM+TE RCS (dBsm) Smooth Cylinder (Blue)
        vs Corrugated Cylinder (Red, --)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
   },
   {Show[{RCSErhoRegCylTMdBxy, RCSErhoCorrdBxy}, PlotRange ->
      {{0, 2 Pi}, {Min[10 * Log10[RCSSolErhoregcylTM], 10 * Log10[RCSSolErho]],
        Max[10 * Log10[RCSSolErhoregcylTM], 10 * Log10[RCSSolErho]]}}, PlotLabel ->
      "Eρ TM RCS (dBsm)Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
```



```
    ,
    Show[{RCSErhoRegCylTEdBxy, RCSErhoCorrdBxy}, PlotRange ->
      {{0, 2 Pi}, {Min[10 * Log10[RCSSolErhoregcylTE], 10 * Log10[RCSSolErho]],
        Max[10 * Log10[RCSSolErhoregcylTE], 10 * Log10[RCSSolErho]]}}, PlotLabel ->
      "Eρ TE RCS Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
    ,
    Show[{RCSErhoRegCylTMplusTEdBxy, RCSErhoCorrdBxy}, PlotRange -> {{0, 2 Pi},
       {Min[10 * Log10[RCSSolErhoregcylTMplusTE], 10 * Log10[RCSSolErho]],
        Max[10 * Log10[RCSSolErhoregcylTMplusTE], 10 * Log10[RCSSolErho]]}},
     PlotLabel -> "Eρ TM+TE RCS (dBsm) Smooth Cylinder (Blue)
        vs Corrugated Cylinder (Red, --)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
   },

   {Show[{RCSEphiRegCylTMdBxy, RCSEphiCorrdBxy}, PlotRange ->
      {{0, 2 Pi}, {Min[10 * Log10[RCSSolEphiregcylTM], 10 * Log10[RCSSolEphi]],
        Max[10 * Log10[RCSSolEphiregcylTM], 10 * Log10[RCSSolEphi]]}}, PlotLabel ->
      "Eϕ TM RCS (dBsm) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
    ,
    Show[{RCSEphiRegCylTEdBxy, RCSEphiCorrdBxy}, PlotRange ->
      {{0, 2 Pi}, {Min[10 * Log10[RCSSolEphiregcylTE], 10 * Log10[RCSSolEphi]],
        Max[10 * Log10[RCSSolEphiregcylTE], 10 * Log10[RCSSolEphi]]}}, PlotLabel ->
      "Eϕ TE RCS (dBsm) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
    ,
    Show[{RCSEphiRegCylTMplusTEdBxy, RCSEphiCorrdBxy},  PlotRange -> {{0, 2 Pi},
       {Min[10 * Log10[RCSSolEphiregcylTMplusTE], 10 * Log10[RCSSolEphi]],
        Max[10 * Log10[RCSSolEphiregcylTMplusTE], 10 * Log10[RCSSolEphi]]}},
     PlotLabel -> "Eϕ TM+TE RCS (dBsm) Smooth Cylinder (Blue)
        vs Corrugated Cylinder (Red, --)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
   },

   {Show[{RCSEAllRegCylTMdBxy, RCSEAllCorrdBxy}, PlotRange ->
      {{0, 2 Pi}, {Min[10 * Log10[RCSSolEregcylTMAll], 10 * Log10[RCSSolEAll]],
        Max[10 * Log10[RCSSolEregcylTMAll], 10 * Log10[RCSSolEAll]]}},
     PlotLabel -> "ETotal TM RCS (dBsm) Smooth Cylinder (Blue)
        vs Corrugated Cylinder (Red, --)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
    ,
    Show[{RCSEAllRegCylTEdBxy, RCSEAllCorrdBxy}, PlotRange ->
```

```
    {{0, 2 Pi}, {Min[10 * Log10[RCSSolEregcylTEAll], 10 * Log10[RCSSolEAll]],
      Max[10 * Log10[RCSSolEregcylTEAll], 10 * Log10[RCSSolEAll]]}},
   PlotLabel -> "ETotal TE RCS (dBsm) Smooth Cylinder (Blue)
      vs Corrugated Cylinder (Red, --)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
  ,
  Show[{RCSEAllRegCylTMplusTEdBxy, RCSEAllCorrdBxy}, PlotRange -> {{0, 2 Pi},
     {Min[10 * Log10[RCSSolEregcylTMplusTEAll], 10 * Log10[RCSSolEAll]],
      Max[10 * Log10[RCSSolEregcylTMplusTEAll], 10 * Log10[RCSSolEAll]]}},
   PlotLabel -> "ETotal TM+TE RCS (dBsm) Smooth Cylinder (Blue)
      vs Corrugated Cylinder (Red, --)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
  }
 }]
```

```
In[45412]:= Export[ToString[StringForm["``.jpg", PhiRCSdBXYPlotsName]], PhiRCSdBXYPlots]
```

RCS dBsm Data Export

```
In[45414]:= tablevalues = {10 * Log10[RCSSolEz], 10 * Log10[RCSSolEzregcylTMplusTE],
    10 * Log10[RCSSolErho], 10 * Log10[RCSSolErhoregcylTMplusTE],
    10 * Log10[RCSSolEphi], 10 * Log10[RCSSolEphiregcylTMplusTE],
    10 * Log10[RCSSolEAll], 10 * Log10[RCSSolEregcylTMplusTEAll]};

In[45415]:= tablecolheading =
  {{"EzCorrugCyl RMS dBsm", "EzSmoothCyl RMS dBsm", "ErhoCorrugCyl RMS dBsm",
    "ErhoSmoothCyl RMS dBsm", "EphiCorrugCyl RMS dBsm", "EphiSmoothCyl RMS dBsm",
    "EtotCorrugCyl RMS dBsm", "EtotSmoothCyl RMS dBsm"}};

In[45416]:= Export[ToString[StringForm["``.xlsx", RCSdBdataSheetName]],
  Join[tablecolheading, Transpose[tablevalues]]]

Speak["The polar plots are done"]
```

Export Data[Changing Phi Data]

```
In[45422]:= data = {a, b, ρ2, ρ1, EztempS, EztempI, ErhotempS, ErhotempI, EphitempS, EphitempI,
    EzregcylTES, EzregcylTMS, EzregcylTEI, EzregcylTMI, ErhoregcylTES,
    ErhoregcylTMS, ErhoregcylTEI, ErhoregcylTMI, EphiregcylTES, EphiregcylTMS,
    EphiregcylTEI, EphiregcylTMI, phirange, ρ, lambda0, θi, phidelta};

In[45423]:= Export[ToString[StringForm["``.mx", Phifilename]], data]
```

Plane-wave Scattering of a Periodic Corrugated Cylinder - Boundary a+b

```
In[1]:= ClearAll["Global`*"]
        SetDirectory["C:\Users\Samuel\Google Drive\Dissertation\FinalData"];
        (*Output Directory*)
```

Import Data [Changing Rho Data]

Boundary 'a' Data

```
In[3]:= dataa = Import["rho_outa.mx"];
In[4]:= a = dataa[[1]];
In[5]:= b = dataa[[2]];
In[6]:= ρ2 = dataa[[3]];
In[7]:= ρ1 = dataa[[4]];
In[8]:= EztempSa = dataa[[5]];
In[9]:= EztempIa = dataa[[6]];
In[10]:= ErhotempSa = dataa[[7]];
In[11]:= ErhotempIa = dataa[[8]];
In[12]:= EphitempSa = dataa[[9]];
In[13]:= EphitempIa = dataa[[10]];
In[14]:= EzregcylTESa = dataa[[11]];
In[15]:= EzregcylTMSa = dataa[[12]];
In[16]:= EzregcylTEIa = dataa[[13]];
In[17]:= EzregcylTMIa = dataa[[14]];
In[18]:= ErhoregcylTESa = dataa[[15]];
In[19]:= ErhoregcylTMSa = dataa[[16]];
In[20]:= ErhoregcylTEIa = dataa[[17]];
```



```
In[21]:= ErhoregcylTMIa = dataa[[18]];
In[22]:= EphiregcylTESa = dataa[[19]];
In[23]:= EphiregcylTMSa = dataa[[20]];
In[24]:= EphiregcylTEIa = dataa[[21]];
In[25]:= EphiregcylTMIa = dataa[[22]];
In[26]:= rhorange = dataa[[23]]; (*rho range*)
In[27]:= ρ = dataa[[24]];
In[28]:= lambda0 = dataa[[25]];
In[29]:= θi = dataa[[26]];
In[30]:= ϕ = dataa[[27]];
         E0 = dataa[[28]];
         H0 = dataa[[29]];
         f0 = dataa[[30]];
         rhomin = dataa[[31]];
         rhomax = dataa[[32]];
         rhodelta = dataa[[33]];
         zpoint = dataa[[34]];
         jqty = dataa[[35]];
         ϕi = dataa[[36]];
         nmin = dataa[[37]];
         nmax = dataa[[38]];
         nqty = dataa[[39]];
         lmin = dataa[[40]];
         lmax = dataa[[41]];
         lqty = dataa[[42]];
         mmin = dataa[[43]];
         mmax = dataa[[44]];
         lqty = dataa[[45]];
         mmaxcheck = dataa[[46]];
         lmaxcheck = dataa[[47]];
         boundarycheck = dataa[[48]];
         (*Eztemp=dataa[[49]];
         Ephitemp=dataa[[50]];
         Erhotemp=dataa[[51]];  *)
         ϕa = dataa[[52]];
         ϕivalue = dataa[[53]];
```


Boundary 'b' Data

```
In[54]:= datab = Import["rho_outb.mx"];
In[55]:= a = datab[[1]];
In[56]:= b = datab[[2]];
In[57]:= ρ2 = datab[[3]];
In[58]:= ρ1 = datab[[4]];
In[59]:= EztempSb = datab[[5]];
In[60]:= EztempIb = datab[[6]];
In[61]:= ErhotempSb = datab[[7]];
In[62]:= ErhotempIb = datab[[8]];
In[63]:= EphitempSb = datab[[9]];
In[64]:= EphitempIb = datab[[10]];
In[65]:= EzregcylTESb = datab[[11]];
In[66]:= EzregcylTMSb = datab[[12]];
In[67]:= EzregcylTEIb = datab[[13]];
In[68]:= EzregcylTMIb = datab[[14]];
In[69]:= ErhoregcylTESb = datab[[15]];
In[70]:= ErhoregcylTMSb = datab[[16]];
In[71]:= ErhoregcylTEIb = datab[[17]];
In[72]:= ErhoregcylTMIb = datab[[18]];
In[73]:= EphiregcylTESb = datab[[19]];
In[74]:= EphiregcylTMSb = datab[[20]];
In[75]:= EphiregcylTEIb = datab[[21]];
In[76]:= EphiregcylTMIb = datab[[22]];
         (*Eztemp=datab[[49]];
         Ephitemp=datab[[50]];
         Erhotemp=datab[[51]];  *)
```


Equations [Changing Rho Data]

```
(*Combining of Boundar 'a' and Boundar 'b' fields*)
EztempS = EztempSa + EztempSb;
EztempI = EztempIa;
ErhotempS = ErhotempSa + ErhotempSb;
ErhotempI = ErhotempIa;
EphitempS = EphitempSa + EphitempSb;
EphitempI = EphitempIa;
EzregcylTES = EzregcylTESa + EzregcylTESb;
EzregcylTMS = EzregcylTMSa + EzregcylTMSb;
EzregcylTEI = EzregcylTEIa;
EzregcylTMI = EzregcylTMIa;
ErhoregcylTES = ErhoregcylTESa + ErhoregcylTESb;
ErhoregcylTMS = ErhoregcylTMSa + ErhoregcylTMSb;
ErhoregcylTEI = ErhoregcylTEIa;
ErhoregcylTMI = ErhoregcylTMIa;
EphiregcylTES = EphiregcylTESa + EphiregcylTESb;
EphiregcylTMS = EphiregcylTMSa + EphiregcylTMSb;
EphiregcylTEI = EphiregcylTEIa;
EphiregcylTMI = EphiregcylTMIa;

(*Changing rho XY plot specific equations*)
EzregcylTMSandI = EzregcylTMI + EzregcylTMS;
ErhoregcylTMSandI = ErhoregcylTMI + ErhoregcylTMS;
EphiregcylTMSandI = EphiregcylTMI + EphiregcylTMS;

EzregcylTESandI = EzregcylTEI + EzregcylTES;
ErhoregcylTESandI = ErhoregcylTEI + ErhoregcylTES;
EphiregcylTESandI = EphiregcylTEI + EphiregcylTES;

Eztemp = EztempI + EztempS;
Erhotemp =  ErhotempI + ErhotempS;
Ephitemp = EphitempI + EphitempS;
```

Changing Rho Plot Calculations

Solution for Plots - Corrugated Cylinder

```
ϕ = ϕa; (*scattered field ϕ, possibly incident ϕ as well*)
(*ϕi=ϕ;*)
ϕi = ϕivalue; (*Only if considered unique and distinct from scattered field*)

(*Incident + Scattered Solution*)
SolEz = N[Abs[Eztemp]];
dataEz = Transpose[{rhorange / lambda0, SolEz}];

SolEphi = N[Abs[Ephitemp]];
dataEphi = Transpose[{rhorange / lambda0, SolEphi}];

SolErho = N[Abs[Erhotemp]];
dataErho = Transpose[{rhorange / lambda0, SolErho}];

SolEAllSandI = N[Abs[Sqrt[((Erhotemp) * Cos[ϕ] - (Ephitemp) * Sin[ϕ]) ^2 +
      ((Erhotemp) * Sin[ϕ] + (Ephitemp) * Cos[ϕ]) ^2 + (Eztemp) ^2]]];
dataEAllSandI = Transpose[{rhorange / lambda0, SolEAllSandI}];

(*Scattered Only Solution*)
SolEzS = N[Abs[EztempS]];
dataEzS = Transpose[{rhorange / lambda0, SolEzS}];

SolEphiS = N[Abs[EphitempS]];
dataEphiS = Transpose[{rhorange / lambda0, SolEphiS}];

SolErhoS = N[Abs[ErhotempS]];
dataErhoS = Transpose[{rhorange / lambda0, SolErhoS}];

SolEAllS = N[Abs[Sqrt[(ErhotempS * Cos[ϕ] - EphitempS * Sin[ϕ]) ^2 +
      (ErhotempS * Sin[ϕ] + EphitempS * Cos[ϕ]) ^2 + (EztempS) ^2]]];
dataEAllS = Transpose[{rhorange / lambda0, SolEAllS}];
```

Solution for Plots - Regular “Smooth” Cylinder

```
ϕ = ϕa; (*scattered field ϕ, possibly incident ϕ as well*)
(*ϕi=ϕ;*)
ϕi = ϕivalue; (*Only if considered unique and distinct from scattered field*)
```


TM Mode Solutions

```
(*Incident + Scattered Solution*)
SolEzregcylTMSandI = N[Abs[EzregcylTMSandI]];
dataEzregcylTMSandI = Transpose[{rhorange / lambda0, SolEzregcylTMSandI}];

SolErhoregcylTMSandI = N[Abs[ErhoregcylTMSandI]];
dataErhoregcylTMSandI = Transpose[{rhorange / lambda0, SolErhoregcylTMSandI}];

SolEphiregcylTMSandI = N[Abs[EphiregcylTMSandI]];
dataEphiregcylTMSandI = Transpose[{rhorange / lambda0, SolEphiregcylTMSandI}];

SolEAllTMSandI =
  N[Abs[Sqrt[((ErhoregcylTMSandI) * Cos[ϕ] - (EphiregcylTMSandI) * Sin[ϕ]) ^2 +
      ((ErhoregcylTMSandI) * Sin[ϕ] + (EphiregcylTMSandI) * Cos[ϕ]) ^2 +
      (EzregcylTMSandI) ^2]]];
dataEAllTMSandI = Transpose[{rhorange / lambda0, SolEAllTMSandI}];

(*Scattered Only Solution*)
SolEzregcylTMS = N[Abs[EzregcylTMS]];
dataEzregcylTMS = Transpose[{rhorange / lambda0, SolEzregcylTMS}];

SolErhoregcylTMS = N[Abs[ErhoregcylTMS]];
dataErhoregcylTMS = Transpose[{rhorange / lambda0, SolErhoregcylTMS}];

SolEphiregcylTMS = N[Abs[EphiregcylTMS]];
dataEphiregcylTMS = Transpose[{rhorange / lambda0, SolEphiregcylTMS}];

SolEAllTMS = N[Abs[Sqrt[(ErhoregcylTMS * Cos[ϕ] - EphiregcylTMS * Sin[ϕ]) ^2 +
      (ErhoregcylTMS * Sin[ϕ] + EphiregcylTMS * Cos[ϕ]) ^2 + (EzregcylTMS) ^2]]];
dataEAllTMS = Transpose[{rhorange / lambda0, SolEAllTMS}];
```


TE Mode Solutions

```
(*Incident + Scattered Solution*)
SolEzregcylTESandI = N[Abs[EzregcylTESandI]];
dataEzregcylTESandI = Transpose[{rhorange / lambda0, SolEzregcylTESandI}];

SolErhoregcylTESandI = N[Abs[ErhoregcylTESandI]];
dataErhoregcylTESandI = Transpose[{rhorange / lambda0, SolErhoregcylTESandI}];

SolEphiregcylTESandI = N[Abs[EphiregcylTESandI]];
dataEphiregcylTESandI = Transpose[{rhorange / lambda0, SolEphiregcylTESandI}];

SolEAllTESandI =
  N[Abs[Sqrt[((ErhoregcylTESandI) * Cos[ϕ] - (EphiregcylTESandI) * Sin[ϕ]) ^2 +
      ((ErhoregcylTESandI) * Sin[ϕ] + (EphiregcylTESandI) * Cos[ϕ]) ^2 +
      (EzregcylTESandI) ^2]]];
dataEAllTESandI = Transpose[{rhorange / lambda0, SolEAllTESandI}];

(*Scattered Only Solution*)
SolEzregcylTES = N[Abs[EzregcylTES]];
dataEzregcylTES = Transpose[{rhorange / lambda0, SolEzregcylTES}];

SolErhoregcylTES = N[Abs[ErhoregcylTES]];
dataErhoregcylTES = Transpose[{rhorange / lambda0, SolErhoregcylTES}];

SolEphiregcylTES = N[Abs[EphiregcylTES]];
dataEphiregcylTES = Transpose[{rhorange / lambda0, SolEphiregcylTES}];

SolEAllTES = N[Abs[Sqrt[(ErhoregcylTES * Cos[ϕ] - EphiregcylTES * Sin[ϕ]) ^2 +
      (ErhoregcylTES * Sin[ϕ] + EphiregcylTES * Cos[ϕ]) ^2 + (EzregcylTES) ^2]]];
dataEAllTES = Transpose[{rhorange / lambda0, SolEAllTES}];
```


TM + TE Mode Solutions

```
(*Incident + Scattered Solution*)
SolEzregcylTMplusTESandI = N[Abs[EzregcylTESandI + EzregcylTMSandI]];
dataEzregcylTMplusTESandI =
  Transpose[{rhorange / lambda0, SolEzregcylTMplusTESandI}];

SolErhoregcylTMplusTESandI = N[Abs[ErhoregcylTESandI + ErhoregcylTMSandI]];
dataErhoregcylTMplusTESandI =
  Transpose[{rhorange / lambda0, SolErhoregcylTMplusTESandI}];

SolEphiregcylTMplusTESandI = N[Abs[EphiregcylTESandI + EphiregcylTMSandI]];
dataEphiregcylTMplusTESandI =
  Transpose[{rhorange / lambda0, SolEphiregcylTMplusTESandI}];

SolEAllTMplusTESandI = N[Abs[Sqrt[((ErhoregcylTESandI + ErhoregcylTMSandI) * Cos[ϕ] -
         (EphiregcylTESandI + EphiregcylTMSandI) * Sin[ϕ]) ^2 +
      ((ErhoregcylTESandI + ErhoregcylTMSandI) * Sin[ϕ] +
         (EphiregcylTESandI + EphiregcylTMSandI) * Cos[ϕ]) ^2 +
      (EzregcylTESandI + EzregcylTMSandI) ^2]]];
dataEAllTMplusTESandI = Transpose[{rhorange / lambda0, SolEAllTMplusTESandI}];

(*Scattered Only Solution*)
SolEzregcylTMplusTES = N[Abs[EzregcylTES + EzregcylTMS]];
dataEzregcylTMplusTES = Transpose[{rhorange / lambda0, SolEzregcylTMplusTES}];

SolErhoregcylTMplusTES = N[Abs[ErhoregcylTES + ErhoregcylTMS]];
dataErhoregcylTMplusTES = Transpose[{rhorange / lambda0, SolErhoregcylTMplusTES}];

SolEphiregcylTMplusTES = N[Abs[EphiregcylTES + EphiregcylTMS]];
dataEphiregcylTMplusTES = Transpose[{rhorange / lambda0, SolEphiregcylTMplusTES}];

SolEAllTMplusTES = N[Abs[Sqrt[((ErhoregcylTES + ErhoregcylTMS) * Cos[ϕ] -
         (EphiregcylTES + EphiregcylTMS) * Sin[ϕ]) ^2 +
      ((ErhoregcylTES + ErhoregcylTMS) * Sin[ϕ] + (EphiregcylTES + EphiregcylTMS) *
          Cos[ϕ]) ^2 + (EzregcylTES + EzregcylTMS) ^2]]];
dataEAllTMplusTES = Transpose[{rhorange / lambda0, SolEAllTMplusTES}];
```

Changing Rho XY Plots

Table of Parameters

```
In[172]:= tablevalues = {{N[E0] "V/m", "-", "-", "-", "-"}, {N[H0] "A/m", "-", "-", "-", "-"},
    {lambda0 "m", "-", "-", "-", "-"}, {f0 "Hz", "-", "-", "-", "-"},
    {a / lambda0 "λ", "-", "-", "-", "-"}, {b / lambda0 "λ", "-", "-", "-", "-"},
    {ρ1 / lambda0 "λ", "-", "-", "-", "-"}, {ρ2 / lambda0 "λ", "-", "-", "-", "-"},
    {"-", rhomin / lambda0 "λ", rhomax / lambda0 "λ", rhodelta / lambda0 "λ",
     Length[rhorange]}, {N[ϕ * (180 / Pi)] "Deg", "-", "-", "-", "-"},
    {zpoint / a "a" && zpoint / lambda0 "λ" , "-", "-", "-", "-"},
    {"-" , "-", "-", "-", jqty}, {N[θi * (180 / Pi)] "Deg", "-", "-", "-", "-"},
    {N[ϕi * (180 / Pi)] "Deg", "-", "-", "-", "-"},
    {"-", nmin, nmax, "1", nqty}, {"-", lmin, lmax, "1", lqty},
    {"-", mmin, mmax, "1", lqty}, {mmaxcheck, "-", "-", "-", "-"},
    {lmaxcheck, "-", "-", "-", "-"}, {"Boundary a+b", "-", "-", "-", "-"}};
  tablerowheading = {"E0", "H0", "λ", "Frequency", "a", "b", "ρ1", "ρ2",
    "ρ range", "ϕ (observed)", "z (observed) ", "Matching Points", "θi",
    "ϕi", "n", "l", "m", "max allowable m", "max allowable l", "Boundary"};
  tablecolheading = {"value", "min", "max", "delta", "Qty of Points"};

In[175]:= ChangingRhoTable = Grid[ArrayFlatten[{{{{" "}}, {tablecolheading}},
     {List /@tablerowheading, ArrayFlatten[tablevalues]}}],
   ItemStyle → {Bold, 20}, Frame → All, Background → {{LightGray}, {LightGray}}]

In[176]:= Export[ToString[StringForm["``.jpg", RhoTableName]], ChangingRhoTable]
```

Corrugated Cylinder Plots

Plot Ez (Scattered)

```
In[177]:= EzCorrS = ListLinePlot[dataEzS,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolEzS]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Ez Corrugated Cylinder (Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Plot Ez (Inc + Scattered)

```
In[178]:= EzCorrSandI = ListLinePlot[dataEz, PlotStyle -> {Red, Dashed},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Corrugated Cylinder (Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot Erho (Scattered)

```
In[179]:= ErhoCorrS = ListLinePlot[dataErhoS,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolErhoS]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Eρ Corrugated Cylinder (Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot Erho (Inc + Scattered)

```
In[180]:= ErhoCorrSandI = ListLinePlot[dataErho,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolErho]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2], PlotMarkers ->
     None, PlotLabel -> "Eρ Corrugated Cylinder (Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot Ephi (Scattered)

```
In[181]:= EphiCorrS = ListLinePlot[dataEphiS,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolEphiS]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Eϕ Corrugated Cylinder (Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Plot Ephi (Inc + Scattered)

```
In[182]:= EphiCorrSandI = ListLinePlot[dataEphi,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolEphi]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2], PlotMarkers ->
     None, PlotLabel -> "Eϕ Corrugated Cylinder (Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot EAll (Scattered)

```
In[183]:= EAllCorrS = ListLinePlot[dataEAllS,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolEAllS]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2], PlotMarkers ->
     None, PlotLabel -> "Etotal Corrugated Cylinder (Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot EAll (Inc + Scattered)

```
In[184]:= EAllCorrSandI = ListLinePlot[dataEAllSandI,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolEAllSandI]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2], PlotMarkers ->
     None, PlotLabel -> "Etotal Corrugated Cylinder (Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary TM Plots

```
In[185]:= GraphicsGrid[{{EzCorrS, EzCorrSandI}, {ErhoCorrS, ErhoCorrSandI},
    {EphiCorrS, EphiCorrSandI}, {EAllCorrS, EAllCorrSandI}},
   Spacings -> {Scaled[0], Scaled[0]}];
```


Smooth Cylinder Plots - TM Plots

Ez (TM Scattered)

```
In[186]:= EzRegCylTMS = ListLinePlot[dataEzregcylTMS,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolEzregcylTMS]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ez (TM Inc + Scattered)

```
In[187]:= EzRegCylTMSandI = ListLinePlot[dataEzregcylTMSandI, PlotRange ->
     {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolEzregcylTMSandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TM Scattered)

```
In[188]:= ErhoRegCylTMS = ListLinePlot[dataErhoregcylTMS,
    PlotRange -> {{rhomin / lambda0, rhomax / lambda0}, {0, Max[SolErhoregcylTMS]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2 / lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TM Inc + Scattered)

```
In[189]:= ErhoRegCylTMSandI = ListLinePlot[dataErhoregcylTMSandI, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolErhoregcylTMSandI]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eρ Smooth Cylinder (TM Incident + Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TM Scattered)

```
In[190]:= EphiRegCylTMS = ListLinePlot[dataEphiregcylTMS,
     PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEphiregcylTMS]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eϕ Smooth Cylinder (TM Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TM Inc + Scattered)

```
In[191]:= EphiRegCylTMSandI = ListLinePlot[dataEphiregcylTMSandI, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEphiregcylTMSandI]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eϕ Smooth Cylinder (TM Incident + Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


EAll (TM Scattered)

```
In[192]:= EAllRegCylTMS = ListLinePlot[dataEAllTMS,
     PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllTMS]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "ETotal Smooth Cylinder (TM Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TM Inc + Scattered)

```
In[193]:= EAllRegCylTMSandI = ListLinePlot[dataEAllTMSandI,
     PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllTMSandI]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "ETotal Smooth Cylinder (TM Incident + Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary TM Plots

```
In[194]:= GraphicsGrid[{{EzRegCylTMS, EzRegCylTMSandI},
     {ErhoRegCylTMS, ErhoRegCylTMSandI}, {EphiRegCylTMS, EphiRegCylTMSandI},
     {EAllRegCylTMS, EAllRegCylTMSandI}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TE Plots

Ez (TE Scattered)

```
In[195]:= EzRegCylTES = ListLinePlot[dataEzregcylTES,
     PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTES]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Ez Smooth Cylinder (TE Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Ez (TE Inc + Scattered)

```
In[196]:= EzRegCylTESandI = ListLinePlot[dataEzregcylTESandI, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTESandI]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Ez Smooth Cylinder (TE Incident + Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TE Scattered)

```
In[197]:= ErhoRegCylTES = ListLinePlot[dataErhoregcylTES,
     PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolErhoregcylTES]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eρ Smooth Cylinder (TE Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TE Inc + Scattered)

```
In[198]:= ErhoRegCylTESandI = ListLinePlot[dataErhoregcylTESandI, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolErhoregcylTESandI]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eρ Smooth Cylinder (TE Incident + Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Ephi (TE Scattered)

```
In[199]:= EphiRegCylTES = ListLinePlot[dataEphiregcylTES,
     PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEphiregcylTES]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eϕ Smooth Cylinder (TE Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TE Inc + Scattered)

```
In[200]:= EphiRegCylTESandI = ListLinePlot[dataEphiregcylTESandI, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEphiregcylTESandI]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eϕ Smooth Cylinder (TE Incident + Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TE Scattered)

```
In[201]:= EAllRegCylTES = ListLinePlot[dataEAllTES,
     PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllTES]}},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "ETotal Smooth Cylinder ((TE Scattered Field)",
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
     Background -> White, ImageSize -> 700,
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TE Inc + Scattered)

```
In[202]:= EAllRegCylTESandI = ListLinePlot[dataEAllTESandI,
    PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_Total Smooth Cylinder (TE Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary TE Plots

```
In[203]:= GraphicsGrid[{{EzRegCylTES, EzRegCylTESandI},
    {ErhoRegCylTES, ErhoRegCylTESandI}, {EphiRegCylTES, EphiRegCylTESandI},
    {EAllRegCylTES, EAllRegCylTESandI}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TM + TE Plots

Ez (TM+TE Scattered)

```
In[204]:= EzRegCylTMplusTES = ListLinePlot[dataEzregcylTMplusTES, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTMplusTES]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_z Smooth Cylinder (TM + TE Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ez (TM+TE Inc + Scattered)

```
In[205]:= EzRegCylTMplusTESandI = ListLinePlot[dataEzregcylTMplusTESandI, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTMplusTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_z Smooth Cylinder (TM + TE Inc and Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Erho (TM+TE Scattered)

```
In[206]:= ErhoRegCylTMplusTES = ListLinePlot[dataErhoregcylTMplusTES, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolErhoregcylTMplusTES]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_ρ Smooth Cylinder (TM + TE Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TM+TE Inc + Scattered)

```
In[207]:= ErhoRegCylTMplusTESandI = ListLinePlot[dataErhoregcylTMplusTESandI, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolErhoregcylTMplusTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_ρ Smooth Cylinder (TM + TE Inc and Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TM+TE Scattered)

```
In[208]:= EphiRegCylTMplusTES = ListLinePlot[dataEphiregcylTMplusTES, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEphiregcylTMplusTES]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_ϕ Smooth Cylinder (TM + TE Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Ephi (TM+TE Inc + Scattered)

```
In[209]:= EphiRegCylTMplusTESandI = ListLinePlot[dataEphiregcylTMplusTESandI, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEphiregcylTMplusTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_ϕ Smooth Cylinder (TM + TE Inc and Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TM+TE Scattered)

```
In[210]:= EAllRegCylTMplusTES = ListLinePlot[dataEAllTMplusTES,
    PlotRange -> {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllTMplusTES]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_Total Smooth Cylinder (TM + TE Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TM+TE Inc + Scattered)

```
In[211]:= EAllRegCylTMplusTESandI = ListLinePlot[dataEAllTMplusTESandI, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllTMplusTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_Total Smooth Cylinder (TM + TE Inc and Scattered)",
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    GridLines -> {{{ρ2/lambda0, {Thick, Gray, Dashed}}}, Automatic},
    Background -> White, ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary TM+TE Plots

```
In[212]:= GraphicsGrid[{{EzRegCylTMplusTES, EzRegCylTMplusTESandI},
    {ErhoRegCylTMplusTES, ErhoRegCylTMplusTESandI},
    {EphiRegCylTMplusTES, EphiRegCylTMplusTESandI},
    {EAllRegCylTMplusTES, EAllRegCylTMplusTESandI}},
   Spacings -> {Scaled[0], Scaled[0]}];
```


Compare (Smooth vs Corrugated) Plots

TM Compare

```
In[213]:= GraphicsGrid[{{Show[EzRegCylTMS, EzCorrS, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTMS, SolEzS]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "E_z TM Scattered Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)"],
   Show[EzRegCylTMSandI, EzCorrSandI, PlotRange -> {{rhomin/lambda0,
       rhomax/lambda0}, {0, Max[SolEzregcylTMSandI, SolEz]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "E_z TM Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"]},
  {Show[ErhoRegCylTMS, ErhoCorrS, PlotRange -> {{rhomin/lambda0, rhomax/lambda0},
      {0, Max[SolErhoregcylTMS, SolErhoS]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "E_ρ TM Scattered Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)"],
   Show[ErhoRegCylTMSandI, ErhoCorrSandI, PlotRange -> {{rhomin/lambda0,
       rhomax/lambda0}, {0, Max[SolErhoregcylTMSandI, SolErho]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "E_ρ TM Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"]},
  {Show[EphiRegCylTMS, EphiCorrS, PlotRange -> {{rhomin/lambda0, rhomax/lambda0},
      {0, Max[SolEphiregcylTMS, SolEphiS]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "E_ϕ TM Scattered Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)"],
   Show[EphiRegCylTMSandI, EphiCorrSandI, PlotRange -> {{rhomin/lambda0,
       rhomax/lambda0}, {0, Max[SolEphiregcylTMSandI, SolEphi]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "E_ϕ TM Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"]},
   {Show[EAllRegCylTMS, EAllCorrS, PlotRange -> {{rhomin/lambda0,
```

```
      rhomax/lambda0}, {0, Max[SolEAllS, SolEAllTMS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_Total TM Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"],
  Show[EAllRegCylTMSandI, EAllCorrSandI, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolEAllTMSandI, SolEAllSandI]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_Total TM Inc + Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"]}
 }];
```

TE Compare

```
In[214]:= GraphicsGrid[{{Show[EzRegCylTES, EzCorrS, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTES, SolEzS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_z TE Scattered Smooth Cylinder (Blue)
      vs Corrugated Cylinder (Red, --)"],
  Show[EzRegCylTESandI, EzCorrSandI, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolEzregcylTESandI, SolEz]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_z TE Inc + Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"]},
 {Show[ErhoRegCylTES, ErhoCorrS, PlotRange -> {{rhomin/lambda0, rhomax/lambda0},
     {0, Max[SolErhoregcylTES, SolErhoS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_ρ TE Scattered Smooth Cylinder (Blue)
      vs Corrugated Cylinder (Red, --)"],
  Show[ErhoRegCylTESandI, ErhoCorrSandI, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolErhoregcylTESandI, SolErho]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_ρ TE Inc + Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"]},
 {Show[EphiRegCylTES, EphiCorrS, PlotRange -> {{rhomin/lambda0, rhomax/lambda0},
     {0, Max[SolEphiregcylTES, SolEphiS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
```



```
    PlotLabel -> "E_φ TE Scattered Smooth Cylinder (Blue)
      vs Corrugated Cylinder (Red, --)"],
  Show[EphiRegCylTESandI, EphiCorrSandI, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolEphiregcylTESandI, SolEphi]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_φ TE Inc + Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"]},
 {Show[EAllRegCylTES, EAllCorrS, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolEAllTES, SolEAllS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_Total TE Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"],
  Show[EAllRegCylTESandI, EAllCorrSandI, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolEAllTESandI, SolEAllSandI]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_Total TE Inc + Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"]}
 }];
```

TM+TE Compare

```
In[215]:= GraphicsGrid[
 {{Show[EzRegCylTMplusTES, EzCorrS, PlotRange -> {{rhomin/lambda0, rhomax/
        lambda0}, {0, Max[SolEzregcylTMplusTES, SolEzS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, Frame -> True,
    FrameLabel -> {"ρ/λ", "V/m"}, PlotLabel -> "E_z TM+TE Scattered Smooth
      Cylinder (Blue) vs Corrugated Cylinder (Red, --)"],
  Show[EzRegCylTMplusTESandI, EzCorrSandI, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolEzregcylTMplusTESandI, SolEz]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_z TM+TE Inc + Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"]},
 {Show[ErhoRegCylTMplusTES, ErhoCorrS, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolErhoregcylTMplusTES, SolErhoS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_ρ TM+TE Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"],
  Show[ErhoRegCylTMplusTESandI, ErhoCorrSandI, PlotRange -> {{rhomin/lambda0,
```



```
      rhomax/lambda0}, {0, Max[SolErhoregcylTMplusTESandI, SolErho]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_ρ TM+TE Inc + Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"]},
 {Show[EphiRegCylTMplusTES, EphiCorrS, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolEphiregcylTMplusTES, SolEphiS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_φ TM+TE Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"],
  Show[EphiRegCylTMplusTESandI, EphiCorrSandI, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolEphiregcylTMplusTESandI, SolEphi]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_φ TM+TE Inc + Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"]},
   {Show[EAllRegCylTES, EAllCorrS, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolEAllS, SolEAllTMplusTES]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_Total TM+TE Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"],
  Show[EAllRegCylTMplusTESandI, EAllCorrSandI, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolEAllTMplusTESandI, SolEAllSandI]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_Total TM+TE Inc + Scattered Smooth
      Cylinder (Blue) vs Corrugated Cylinder (Red, --)"]}
 }];
```

TM, TE, TM+TE Compare

```
In[216]:= RhoXYPlots = GraphicsGrid[{{Show[EzRegCylTMS, EzCorrS, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTMS, SolEzS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_z TM Scattered Smooth Cylinder (Blue)
      vs Corrugated Cylinder (Red, --)"],
   "E_z TE Does Not Exist",
   Show[EzRegCylTMplusTES, EzCorrS, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolEzregcylTMplusTES, SolEzS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
```



```
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_z TM+TE Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"]},

  {Show[EzRegCylTMSandI, EzCorrSandI, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEzregcylTMSandI, SolEz]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_z TM Inc + Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"],
   "E_z TE Does Not Exist",
   Show[EzRegCylTMplusTESandI,
    EzCorrSandI, PlotRange -> {{rhomin/lambda0, rhomax/lambda0},
      {0, Max[SolEzregcylTMplusTESandI, SolEz]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_z TM+TE Inc + Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"]},

  {Show[ErhoRegCylTMS, ErhoCorrS, PlotRange ->
     {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolErhoregcylTMS, SolErhoS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_ρ TM Scattered Smooth Cylinder (Blue)
      vs Corrugated Cylinder (Red, --)"],
   Show[ErhoRegCylTES, ErhoCorrS, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolErhoregcylTES, SolErhoS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_ρ TE Scattered Smooth Cylinder (Blue)
      vs Corrugated Cylinder (Red, --)"],
   Show[ErhoRegCylTMplusTES, ErhoCorrS, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolErhoregcylTMplusTES, SolErhoS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_ρ TM+TE Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"]},

  {Show[ErhoRegCylTMSandI, ErhoCorrSandI, PlotRange -> {{rhomin/lambda0,
      rhomax/lambda0}, {0, Max[SolErhoregcylTMSandI, SolErho]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "E_ρ TM Inc + Scattered Smooth Cylinder
```

```
      (Blue) vs Corrugated Cylinder (Red, --)"],
    Show[ErhoRegCylTESandI, ErhoCorrSandI, PlotRange -> {{rhomin/lambda0,
       rhomax/lambda0}, {0, Max[SolErhoregcylTESandI, SolErho]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "Eρ TE Inc + Scattered Smooth Cylinder (Blue) vs Corrugated
       Cylinder (Red, --)"], Show[ErhoRegCylTMplusTESandI,
     ErhoCorrSandI, PlotRange -> {{rhomin/lambda0, rhomax/lambda0},
      {0, Max[SolErhoregcylTMplusTESandI, SolErho]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "Eρ TM+TE Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"]},

{Show[EphiRegCylTMS, EphiCorrS, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEphiregcylTMS, SolEphiS]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "Eϕ TM Scattered Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)"],
    Show[EphiRegCylTES, EphiCorrS, PlotRange -> {{rhomin/lambda0,
       rhomax/lambda0}, {0, Max[SolEphiregcylTES, SolEphiS]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "Eϕ TE Scattered Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)"],
    Show[EphiRegCylTMplusTES, EphiCorrS, PlotRange -> {{rhomin/lambda0,
       rhomax/lambda0}, {0, Max[SolEphiregcylTMplusTES, SolEphiS]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "Eϕ TM+TE Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"]},

 {Show[EphiRegCylTMSandI, EphiCorrSandI, PlotRange -> {{rhomin/lambda0,
       rhomax/lambda0}, {0, Max[SolEphiregcylTMSandI, SolEphi]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "Eϕ TM Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"],
    Show[EphiRegCylTESandI, EphiCorrSandI, PlotRange -> {{rhomin/lambda0,
       rhomax/lambda0}, {0, Max[SolEphiregcylTESandI, SolEphi]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
```



```
     PlotLabel -> "Eϕ TE Inc + Scattered Smooth Cylinder (Blue) vs Corrugated
       Cylinder (Red, --)"], Show[EphiRegCylTMplusTESandI,
     EphiCorrSandI, PlotRange -> {{rhomin/lambda0, rhomax/lambda0},
      {0, Max[SolEphiregcylTMplusTESandI, SolEphi]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "Eϕ TM+TE Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"]},

 {Show[EAllRegCylTMS, EAllCorrS, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllS, SolEAllTMS]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "ETotal TM Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"],
    Show[EAllRegCylTES, EAllCorrS, PlotRange -> {{rhomin/lambda0,
       rhomax/lambda0}, {0, Max[SolEAllS, SolEAllTES]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "ETotal TE Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"],
    Show[EAllRegCylTMplusTES, EAllCorrS, PlotRange ->
      {{rhomin/lambda0, rhomax/lambda0}, {0, Max[SolEAllS, SolEAllTMplusTES]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "ETotal TM+TE Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"]},

 {Show[EAllRegCylTMSandI, EAllCorrSandI, PlotRange -> {{rhomin/lambda0,
       rhomax/lambda0}, {0, Max[SolEAllTMSandI, SolEAllSandI]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "ETotal TM Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"],
    Show[EAllRegCylTMplusTESandI, EAllCorrSandI, PlotRange -> {{rhomin/lambda0,
       rhomax/lambda0}, {0, Max[SolEAllTMplusTESandI, SolEAllSandI]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "ETotal TM+TE Inc + Scattered Smooth
       Cylinder (Blue) vs Corrugated Cylinder (Red, --)"],
    Show[EAllRegCylTMplusTESandI, EAllCorrSandI, PlotRange -> {{rhomin/lambda0,
       rhomax/lambda0}, {0, Max[SolEAllTMplusTESandI, SolEAllSandI]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
```



```
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "ETotal TM+TE Inc + Scattered Smooth
       Cylinder (Blue) vs Corrugated Cylinder (Red, --)"]}
  }]
In[217]:= Export[ToString[StringForm["``.jpg", RhoXYPlotsName]], RhoXYPlots];
In[218]:= Speak["The rho plots are done"]
```

Import Data [Changing Phi Data]

Boundary 'a' Data

```
In[219]:= dataa = Import["phi_outa.mx"];
In[220]:= a = dataa[[1]];
In[221]:= b = dataa[[2]];
In[222]:= ρ2 = dataa[[3]];
In[223]:= ρ1 = dataa[[4]];
In[224]:= EztempSa = dataa[[5]];
In[225]:= EztempIa = dataa[[6]];
In[226]:= ErhotempSa = dataa[[7]];
In[227]:= ErhotempIa = dataa[[8]];
In[228]:= EphitempSa = dataa[[9]];
In[229]:= EphitempIa = dataa[[10]];
In[230]:= EzregcylTESa = dataa[[11]];
In[231]:= EzregcylTMSa = dataa[[12]];
In[232]:= EzregcylTEIa = dataa[[13]];
In[233]:= EzregcylTMIa = dataa[[14]];
In[234]:= ErhoregcylTESa = dataa[[15]];
In[235]:= ErhoregcylTMSa = dataa[[16]];
In[236]:= ErhoregcylTEIa = dataa[[17]];
In[237]:= ErhoregcylTMIa = dataa[[18]];
In[238]:= EphiregcylTESa = dataa[[19]];
In[239]:= EphiregcylTMSa = dataa[[20]];
```



```
In[240]:= EphiregcylTEIa = dataa[[21]];
In[241]:= EphiregcylTMIa = dataa[[22]];
In[242]:= phirange = dataa[[23]];
In[243]:= ρ = dataa[[24]];
In[244]:= lambda0 = dataa[[25]];
In[245]:= θi = dataa[[26]];
In[246]:= phidelta = dataa[[27]];
```

Boundary 'b' Data

```
In[247]:= datab = Import["phi_outb.mx"];
In[248]:= a = datab[[1]];
In[249]:= b = datab[[2]];
In[250]:= ρ2 = datab[[3]];
In[251]:= ρ1 = datab[[4]];
In[252]:= EztempSb = datab[[5]];
In[253]:= EztempIb = datab[[6]];
In[254]:= ErhotempSb = datab[[7]];
In[255]:= ErhotempIb = datab[[8]];
In[256]:= EphitempSb = datab[[9]];
In[257]:= EphitempIb = datab[[10]];
In[258]:= EzregcylTESb = datab[[11]];
In[259]:= EzregcylTMSb = datab[[12]];
In[260]:= EzregcylTEIb = datab[[13]];
In[261]:= EzregcylTMIb = datab[[14]];
In[262]:= ErhoregcylTESb = datab[[15]];
In[263]:= ErhoregcylTMSb = datab[[16]];
In[264]:= ErhoregcylTEIb = datab[[17]];
In[265]:= ErhoregcylTMIb = datab[[18]];
In[266]:= EphiregcylTESb = datab[[19]];
In[267]:= EphiregcylTMSb = datab[[20]];
In[268]:= EphiregcylTEIb = datab[[21]];
```

```
In[269]:= EphiregcylTMIb = datab[[22]];
```

Comparison Paper - A. Freni

```
In[270]:= CrossPolarFEMdB = Import["CrossPolarFEM.csv"];
CrossPolarFEMdBXYPlot = ListLinePlot[CrossPolarFEMdB, PlotRange ->
    {{0, Pi}, {Min[CrossPolarFEMdB[[All, 2]]], Max[CrossPolarFEMdB[[All, 2]]]}},
   PlotStyle -> {Green, Dotted, Thickness[.01]},
   BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
   PlotLabel -> "σφθ/λ0 (dB) Corrugated Cylinder (RCS dBsm)",
   Frame -> True, FrameLabel -> {"ϕ", "σφθ/λ0 (dB)"},
   GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];

In[272]:= CrossPolarMoMdB = Import["CrossPolarMoM.csv"];
CrossPolarMoMdBXYPlot = ListLinePlot[CrossPolarMoMdB, PlotRange ->
    {{0, Pi}, {Min[CrossPolarMoMdB[[All, 2]]], Max[CrossPolarMoMdB[[All, 2]]]}},
   PlotStyle -> {Blue, Thickness[.005]},
   BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
   PlotLabel -> "σφθ/λ0 (dB) Corrugated Cylinder (RCS dBsm)",
   Frame -> True, FrameLabel -> {"ϕ", "σφθ/λ0 (dB)"},
   GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Equations [Changing Phi Data]

```
In[274]:= (*Combining of Boundar 'a' and Boundar 'b' fields*)
EztempS = EztempSa + EztempSb;
EztempI = EztempIa;
ErhotempS = ErhotempSa + ErhotempSb;
ErhotempI = ErhotempIa;
EphitempS = EphitempSa + EphitempSb;
EphitempI = EphitempIa;
EzregcylTES = EzregcylTESa + EzregcylTESb;
EzregcylTMS = EzregcylTMSa + EzregcylTMSb;
EzregcylTEI = EzregcylTEIa;
EzregcylTMI = EzregcylTMIa;
ErhoregcylTES = ErhoregcylTESa + ErhoregcylTESb;
ErhoregcylTMS = ErhoregcylTMSa + ErhoregcylTMSb;
ErhoregcylTEI = ErhoregcylTEIa;
ErhoregcylTMI = ErhoregcylTMIa;
EphiregcylTES = EphiregcylTESa + EphiregcylTESb;
EphiregcylTMS = EphiregcylTMSa + EphiregcylTMSb;
EphiregcylTEI = EphiregcylTEIa;
EphiregcylTMI = EphiregcylTMIa;

(*Additional changing phi plot equations*)
EzregcylTMSandI = EzregcylTMI + EzregcylTMS;
ErhoregcylTMSandI = ErhoregcylTMI + ErhoregcylTMS;
EphiregcylTMSandI = EphiregcylTMI + EphiregcylTMS;

EzregcylTESandI = EzregcylTEI + EzregcylTES;
ErhoregcylTESandI = ErhoregcylTEI + ErhoregcylTES;
EphiregcylTESandI = EphiregcylTEI + EphiregcylTES;

Eztemp = EztempI + EztempS;
Erhotemp = ErhotempI + ErhotempS;
Ephitemp = EphitempI + EphitempS;
```

Changing Phi Plot Calculations

Solution for Plots - Corrugated Cylinder

```
In[301]:= (*Incident + Scattered Solution*)
SolEz = N[Abs[Eztemp]];
```



```
dataEz = Transpose[{phirange, SolEz}];

SolEphi = N[Abs[Ephitemp]];
dataEphi = Transpose[{phirange, SolEphi}];

SolErho = N[Abs[Erhotemp]];
dataErho = Transpose[{phirange, SolErho}];

SolEAllSandI =
  N[Abs[Sqrt[((Erhotemp) * Cos[phirange] - (Ephitemp) * Sin[phirange]) ^2 +
       ((Erhotemp) * Sin[phirange] + (Ephitemp) * Cos[phirange]) ^2 + (Eztemp) ^2]]];
dataEAllSandI = Transpose[{phirange, SolEAllSandI}];

(*Scattered Only Solution*)
SolEzS = N[Abs[EztempS]];
dataEzS = Transpose[{phirange, SolEzS}];

SolEphiS = N[Abs[EphitempS]];
dataEphiS = Transpose[{phirange, SolEphiS}];

SolErhoS = N[Abs[ErhotempS]];
dataErhoS = Transpose[{phirange, SolErhoS}];

SolEAllS = N[Abs[Sqrt[(ErhotempS * Cos[phirange] - EphitempS * Sin[phirange]) ^2 +
       (ErhotempS * Sin[phirange] + EphitempS * Cos[phirange]) ^2 + (EztempS) ^2]]];
dataEAllS = Transpose[{phirange, SolEAllS}];

(*RCS Solution*)
RCSSolEz = (4 * Pi * (ρ^2)) * (((Abs[EztempS]) ^2) / ((Abs[EztempI]) ^2));
RCSdataEz = Transpose[{phirange, RCSSolEz}];

RCSSolErho = (4 * Pi * (ρ^2)) * (((Abs[ErhotempS]) ^2) / ((Abs[ErhotempI]) ^2));
RCSdataErho = Transpose[{phirange, RCSSolErho}];

RCSSolEphi = (4 * Pi * (ρ^2)) * (((Abs[EphitempS]) ^2) / ((Abs[EphitempI]) ^2));
RCSdataEphi = Transpose[{phirange, RCSSolEphi}];

RCSSolEAllS = Sqrt[(ErhotempS * Cos[phirange] - EphitempS * Sin[phirange]) ^2 +
     (ErhotempS * Sin[phirange] + EphitempS * Cos[phirange]) ^2 + (EztempS) ^2];
RCSSolEAllI = Sqrt[(ErhotempI * Cos[phirange] - EphitempI * Sin[phirange]) ^2 +
     (ErhotempI * Sin[phirange] + EphitempI * Cos[phirange]) ^2 + (EztempI) ^2];
RCSSolEAll = (4 * Pi * (ρ^2)) * (((Abs[RCSSolEAllS]) ^2) / ((Abs[RCSSolEAllI]) ^2));
RCSdataEAll = Transpose[{phirange, RCSSolEAll}];
```



```
(*RCS Solution - in dB*)
RCSdataEzdB = Transpose[{phirange, 10 * Log10[RCSSolEz]}];
RCSdataErhodB = Transpose[{phirange, 10 * Log10[RCSSolErho]}];
RCSdataEphidB = Transpose[{phirange, 10 * Log10[RCSSolEphi]}];
RCSdataEAlldB = Transpose[{phirange, 10 * Log10[RCSSolEAll]}];

In[331]:= (*RCS Solution*)
RCSSolEz = (4 * Pi * (ρ^2)) * (((Abs[EztempS]) ^2) / ((Abs[EztempI]) ^2));
RCSdataEz = Transpose[{phirange, RCSSolEz}];

RCSSolErho = (4 * Pi * (ρ^2)) * (((Abs[ErhotempS]) ^2) / ((Abs[ErhotempI]) ^2));
RCSdataErho = Transpose[{phirange, RCSSolErho}];

RCSSolEphi = (4 * Pi * (ρ^2)) * (((Abs[EphitempS]) ^2) / ((Abs[EphitempI]) ^2));
RCSdataEphi = Transpose[{phirange, RCSSolEphi}];

RCSSolEAllS = Sqrt[(ErhotempS * Cos[phirange] - EphitempS * Sin[phirange]) ^2 +
     (ErhotempS * Sin[phirange] + EphitempS * Cos[phirange]) ^2 + (EztempS) ^2];
RCSSolEAllI = Sqrt[(ErhotempI * Cos[phirange] - EphitempI * Sin[phirange]) ^2 +
     (ErhotempI * Sin[phirange] + EphitempI * Cos[phirange]) ^2 + (EztempI) ^2];
RCSSolEAll = (4 * Pi * (ρ^2)) * (((Abs[RCSSolEAllS]) ^2) / ((Abs[RCSSolEAllI]) ^2));
RCSdataEAll = Transpose[{phirange, RCSSolEAll}];

(*RCS Solution - in dB*)
RCSdataEzdB = Transpose[{phirange, 10 * Log10[RCSSolEz]}];
RCSdataErhodB = Transpose[{phirange, 10 * Log10[RCSSolErho]}];
RCSdataEphidB = Transpose[{phirange, 10 * Log10[RCSSolEphi]}];
RCSdataEAlldB = Transpose[{phirange, 10 * Log10[RCSSolEAll]}];

(*---------------------FOR COMPARE TO REFERENCE-----------*)
(* Convert to spherical *)
SolErSspherical = ErhotempS * Sin[θi] + EztempS * Cos[θi];
SolErIspherical = ErhotempI * Sin[θi] + EztempI * Cos[θi];

SolEthetaSspherical = ErhotempS * Cos[θi] - EztempS * Sin[θi];
SolEthetaIspherical = ErhotempI * Cos[θi] - EztempI * Sin[θi];

SolEphiSspherical = EphitempS;
SolEphiIspherical = EphitempI;
(* Convert to Co-Polar and Cross-Polar *)
EcopolarS =
  SolEthetaSspherical * Cos[phirange] - SolEphiSspherical * Sin[phirange];
```

```
EcopolarI = SolEthetaIspherical * Cos[phirange] - SolEphiIspherical * Sin[phirange];

EcrossopolarS =
  SolEthetaSspherical * Sin[phirange] + SolEphiSspherical * Cos[phirange];
EcrossopolarI = SolEthetaIspherical * Sin[phirange] +
   SolEphiIspherical * Cos[phirange];

(* Convert to RCS and RCS db - Co-Polar and Cross-Polar *)
RCSσθθ = (2 * Pi * ρ) * Abs[SolEthetaSspherical / SolEthetaIspherical]^2;
(*Co-Polar according to A. Freni, 1996,
Scattering from a dielectric cylinder axially loaded
with periodic metallic rings*)
RCSσϕθ = (2 * Pi * ρ) * Abs[SolEphiSspherical / SolEthetaIspherical]^2;
(*Cross-Polar according to A. Freni, 1996,
Scattering from a dielectric cylinder axially loaded
with periodic metallic rings*)
RCSσθθdB = Transpose[{phirange, 10 * Log10[RCSσθθ / lambda0]}];
RCSσϕθdB = Transpose[{phirange, 10 * Log10[RCSσϕθ / lambda0]}];

RCSCopolar = (2 * Pi * ρ) * Abs[EcopolarS / EcopolarI]^2;
RCSCrosspolar = (2 * Pi * ρ) * Abs[EcrossopolarS / EcrossopolarI]^2;

RCSCopolar2 = (2 * Pi * ρ) * Abs[EcopolarS / RCSSolEAllI]^2;
RCSCrosspolar2 = (2 * Pi * ρ) * Abs[EcrossopolarS / RCSSolEAllI]^2;

RCSCopolardB = Transpose[{phirange, 10 * Log10[RCSCopolar / lambda0]}];
RCSCrosspolardB = Transpose[{phirange, 10 * Log10[RCSCrosspolar / lambda0]}];

RCSCopolardB2 = Transpose[{phirange, 10 * Log10[RCSCopolar2 / lambda0]}];
RCSCrosspolardB2 = Transpose[{phirange, 10 * Log10[RCSCrosspolar2 / lambda0]}];
```

Solution for Plots - Regular "Smooth" Cylinder

TM Mode Solutions

```
In[367]:= (*Incident + Scattered Solution*)
SolEzregcylTMSandI = N[Abs[EzregcylTMSandI]];
dataEzregcylTMSandI = Transpose[{phirange, SolEzregcylTMSandI}];

SolErhoregcylTMSandI = N[Abs[ErhoregcylTMSandI]];
dataErhoregcylTMSandI = Transpose[{phirange, SolErhoregcylTMSandI}];
```



```
SolEphiregcylTMSandI = N[Abs[EphiregcylTMSandI]];
dataEphiregcylTMSandI = Transpose[{phirange, SolEphiregcylTMSandI}];

SolEAllTMS =
  N[Abs[Sqrt[(ErhoregcylTMS * Cos[phirange] - EphiregcylTMS * Sin[phirange])^2 +
      (ErhoregcylTMS * Sin[phirange] + EphiregcylTMS * Cos[phirange])^2 +
      (EzregcylTMS)^2]]];
dataEAllTMS = Transpose[{phirange, SolEAllTMS}];

(*Scattered Only Solution*)
SolEzregcylTMS = N[Abs[EzregcylTMS]];
dataEzregcylTMS = Transpose[{phirange, SolEzregcylTMS}];

SolErhoregcylTMS = N[Abs[ErhoregcylTMS]];
dataErhoregcylTMS = Transpose[{phirange, SolErhoregcylTMS}];

SolEphiregcylTMS = N[Abs[EphiregcylTMS]];
dataEphiregcylTMS = Transpose[{phirange, SolEphiregcylTMS}];

SolEAllTMSandI = N[Abs[Sqrt[
      ((ErhoregcylTMSandI) * Cos[phirange] - (EphiregcylTMSandI) * Sin[phirange])^2 +
       ((ErhoregcylTMSandI) * Sin[phirange] + (EphiregcylTMSandI) * Cos[phirange])^2 +
       (EzregcylTMSandI)^2]]];
dataEAllTMSandI = Transpose[{phirange, SolEAllTMSandI}];

(*RCS Solution*)
RCSSolEzregcylTM =
  (4 * Pi * (ρ^2)) * (((Abs[EzregcylTMS])^2) / ((Abs[EzregcylTMI])^2));
RCSdataEzregcylTM = Transpose[{phirange, RCSSolEzregcylTM}];

RCSSolErhoregcylTM =
  (4 * Pi * (ρ^2)) * (((Abs[ErhoregcylTMS])^2) / ((Abs[ErhoregcylTMI])^2));
RCSdataErhoregcylTM = Transpose[{phirange, RCSSolErhoregcylTM}];

RCSSolEphiregcylTM =
  (4 * Pi * (ρ^2)) * (((Abs[EphiregcylTMS])^2) / ((Abs[EphiregcylTMI])^2));
RCSdataEphiregcylTM = Transpose[{phirange, RCSSolEphiregcylTM}];

RCSSolEregcylTMAllS =
  Sqrt[(ErhoregcylTMS * Cos[phirange] - EphiregcylTMS * Sin[phirange])^2 +
    (ErhoregcylTMS * Sin[phirange] + EphiregcylTMS * Cos[phirange])^2 +
    (EzregcylTMS)^2];
RCSSolEregcylTMAllI = Sqrt[
```



```
    (ErhoregcylTMI * Cos[phirange] - EphiregcylTMI * Sin[phirange])^2 +
     (ErhoregcylTMI * Sin[phirange] + EphiregcylTMI * Cos[phirange])^2 +
     (EzregcylTMI)^2];
RCSSolEregcylTMAll = (4 * Pi * (ρ^2)) *
    (((Abs[RCSSolEregcylTMAllS])^2) / ((Abs[RCSSolEregcylTMAllI])^2));
RCSdataEregcylTMAll = Transpose[{phirange, RCSSolEregcylTMAll}];

(*RCS Solution - in dB*)
RCSdataEzregcylTMdB = Transpose[{phirange, 10 * Log10[RCSSolEzregcylTM]}];
RCSdataErhoregcylTMdB = Transpose[{phirange, 10 * Log10[RCSSolErhoregcylTM]}];
RCSdataEphiregcylTMdB = Transpose[{phirange, 10 * Log10[RCSSolEphiregcylTM]}];
RCSdataEregcylTMAlldB = Transpose[{phirange, 10 * Log10[RCSSolEregcylTMAll]}];
```

TE Mode Solutions

```
In[397]:= (*Incident + Scattered Solution*)
SolEzregcylTESandI = N[Abs[EzregcylTESandI]];
dataEzregcylTESandI = Transpose[{phirange, SolEzregcylTESandI}];

SolErhoregcylTESandI = N[Abs[ErhoregcylTESandI]];
dataErhoregcylTESandI = Transpose[{phirange, SolErhoregcylTESandI}];

SolEphiregcylTESandI = N[Abs[EphiregcylTESandI]];
dataEphiregcylTESandI = Transpose[{phirange, SolEphiregcylTESandI}];

SolEAllTESandI = N[Abs[Sqrt[
      ((ErhoregcylTESandI) * Cos[phirange] - (EphiregcylTESandI) * Sin[phirange])^2 +
       ((ErhoregcylTESandI) * Sin[phirange] + (EphiregcylTESandI) * Cos[phirange])^2 +
       (EzregcylTESandI)^2]]];
dataEAllTESandI = Transpose[{phirange, SolEAllTESandI}];

(*Scattered Only Solution*)
SolEzregcylTES = N[Abs[EzregcylTES]];
dataEzregcylTES = Transpose[{phirange, SolEzregcylTES}];

SolErhoregcylTES = N[Abs[ErhoregcylTES]];
dataErhoregcylTES = Transpose[{phirange, SolErhoregcylTES}];

SolEphiregcylTES = N[Abs[EphiregcylTES]];
dataEphiregcylTES = Transpose[{phirange, SolEphiregcylTES}];

SolEAllTES =
  N[Abs[Sqrt[(ErhoregcylTES * Cos[phirange] - EphiregcylTES * Sin[phirange])^2 +
```



```
      (ErhoregcylTES * Sin[phirange] + EphiregcylTES * Cos[phirange])^2 +
      (EzregcylTES)^2]]];
dataEAllTES = Transpose[{phirange, SolEAllTES}];

(*RCS Solution*)
(*RCSSolEzregcylTE = DNE;*)
(*RCSdataEzregcylTE =DNE;*)
RCSSolErhoregcylTE =
  (4 * Pi * (ρ^2)) * (((Abs[ErhoregcylTES])^2) / ((Abs[ErhoregcylTEI])^2));
RCSdataErhoregcylTE = Transpose[{phirange, RCSSolErhoregcylTE}];

RCSSolEphiregcylTE =
  (4 * Pi * (ρ^2)) * (((Abs[EphiregcylTES])^2) / ((Abs[EphiregcylTEI])^2));
RCSdataEphiregcylTE = Transpose[{phirange, RCSSolEphiregcylTE}];

RCSSolEregcylTEAllS =
  Sqrt[(ErhoregcylTES * Cos[phirange] - EphiregcylTES * Sin[phirange])^2 +
    (ErhoregcylTES * Sin[phirange] + EphiregcylTES * Cos[phirange])^2 +
    (EzregcylTES)^2];
RCSSolEregcylTEAllI = Sqrt[
   (ErhoregcylTEI * Cos[phirange] - EphiregcylTEI * Sin[phirange])^2 +
    (ErhoregcylTEI * Sin[phirange] + EphiregcylTEI * Cos[phirange])^2 +
    (EzregcylTEI)^2];
RCSSolEregcylTEAll = (4 * Pi * (ρ^2)) *
   (((Abs[RCSSolEregcylTEAllS])^2) / ((Abs[RCSSolEregcylTEAllI])^2));
RCSdataEregcylTEAll = Transpose[{phirange, RCSSolEregcylTEAll}];

(*RCS Solution - in dB*)
RCSdataErhoregcylTEdB = Transpose[{phirange, 10 * Log10[ RCSSolErhoregcylTE]}];
RCSdataEphiregcylTEdB = Transpose[{phirange, 10 * Log10[RCSSolEphiregcylTE]}];
RCSdataEregcylTEAlldB = Transpose[{phirange, 10 * Log10[RCSSolEregcylTEAll]}];
```

TM + TE Mode Solutions

```
(*Incident + Scattered Solution*)
SolEzregcylTMplusTESandI = N[Abs[EzregcylTESandI + EzregcylTMSandI]];
dataEzregcylTMplusTESandI = Transpose[{phirange, SolEzregcylTMplusTESandI}];

SolErhoregcylTMplusTESandI = N[Abs[ErhoregcylTESandI + ErhoregcylTMSandI]];
dataErhoregcylTMplusTESandI = Transpose[{phirange, SolErhoregcylTMplusTESandI}];

SolEphiregcylTMplusTESandI = N[Abs[EphiregcylTESandI + EphiregcylTMSandI]];
dataEphiregcylTMplusTESandI = Transpose[{phirange, SolEphiregcylTMplusTESandI}];
```

```
SolEAllTMplusTESandI =
  N[Abs[Sqrt[((ErhoregcylTESandI + ErhoregcylTMSandI) * Cos[phirange] -
        (EphiregcylTESandI + EphiregcylTMSandI) * Sin[phirange]) ^2 +
      ((ErhoregcylTESandI + ErhoregcylTMSandI) * Sin[phirange] +
        (EphiregcylTESandI + EphiregcylTMSandI) * Cos[phirange]) ^2 +
      (EzregcylTESandI + EzregcylTMSandI) ^2]]];
dataEAllTMplusTESandI = Transpose[{phirange, SolEAllTMplusTESandI}];

(*Scattered Only Solution*)
SolEzregcylTMplusTES = N[Abs[EzregcylTES + EzregcylTMS]];
dataEzregcylTMplusTES = Transpose[{phirange, SolEzregcylTMplusTES}];

SolErhoregcylTMplusTES = N[Abs[ErhoregcylTES + ErhoregcylTMS]];
dataErhoregcylTMplusTES = Transpose[{phirange, SolErhoregcylTMplusTES}];

SolEphiregcylTMplusTES = N[Abs[EphiregcylTES + EphiregcylTMS]];
dataEphiregcylTMplusTES = Transpose[{phirange, SolEphiregcylTMplusTES}];

SolEAllTMplusTES = N[Abs[Sqrt[((ErhoregcylTES + ErhoregcylTMS) * Cos[phirange] -
        (EphiregcylTES + EphiregcylTMS) * Sin[phirange]) ^2 +
      ((ErhoregcylTES + ErhoregcylTMS) * Sin[phirange] + (EphiregcylTES +
           EphiregcylTMS) * Cos[phirange]) ^2 + (EzregcylTES + EzregcylTMS) ^2]]];
dataEAllTMplusTES = Transpose[{phirange, SolEAllTMplusTES}];

(*RCS Solution*)
RCSSolEzregcylTMplusTE = (4 * Pi * (ρ^2)) *
   (((Abs[EzregcylTES + EzregcylTMS]) ^2) / ((Abs[EzregcylTEI + EzregcylTMI]) ^2));
RCSdataEzregcylTMplusTE = Transpose[{phirange, RCSSolEzregcylTMplusTE}];

RCSSolErhoregcylTMplusTE =
  (4 * Pi * (ρ^2)) * (((Abs[ErhoregcylTES + ErhoregcylTMS]) ^2) /
     ((Abs[ErhoregcylTEI + ErhoregcylTMI]) ^2));
RCSdataErhoregcylTMplusTE = Transpose[{phirange, RCSSolErhoregcylTMplusTE}];

RCSSolEphiregcylTMplusTE =
  (4 * Pi * (ρ^2)) * (((Abs[EphiregcylTES + EphiregcylTMS]) ^2) /
     ((Abs[EphiregcylTEI + EphiregcylTMI]) ^2));
RCSdataEphiregcylTMplusTE = Transpose[{phirange, RCSSolEphiregcylTMplusTE}];

RCSSolEregcylTMplusTEAllS = Sqrt[
   ((ErhoregcylTES + ErhoregcylTMS) * Cos[phirange] - (EphiregcylTES + EphiregcylTMS) *
        Sin[phirange]) ^2 + ((ErhoregcylTES + ErhoregcylTMS) * Sin[phirange] +
```



```
        (EphiregcylTES + EphiregcylTMS) * Cos[phirange]) ^
      2 + (EzregcylTES + EzregcylTMS) ^2];
RCSSolEregcylTMplusTEAllI = Sqrt[((ErhoregcylTEI + ErhoregcylTMI) * Cos[phirange] -
        (EphiregcylTEI + EphiregcylTMI) * Sin[phirange]) ^2 +
     ((ErhoregcylTEI + ErhoregcylTMI) * Sin[phirange] +
        (EphiregcylTEI + EphiregcylTMI) * Cos[phirange]) ^
      2 + (EzregcylTEI + EzregcylTMI) ^2];
RCSSolEregcylTMplusTEAll = (4 * Pi * (ρ^2)) *
   (((Abs[RCSSolEregcylTMplusTEAllS]) ^2) / ((Abs[RCSSolEregcylTMplusTEAllI]) ^2));
RCSdataEregcylTMplusTEAll = Transpose[{phirange, RCSSolEregcylTMplusTEAll}];

(*RCS Solution - in dB*)
RCSdataEzregcylTMplusTEdB =
  Transpose[{phirange, 10 * Log10[RCSSolEzregcylTMplusTE]}];
RCSdataErhoregcylTMplusTEdB =
  Transpose[{phirange, 10 * Log10[RCSSolErhoregcylTMplusTE]}];
RCSdataEphiregcylTMplusTEdB =
  Transpose[{phirange, 10 * Log10[RCSSolEphiregcylTMplusTE]}];
RCSdataEregcylTMplusTEAlldB =
  Transpose[{phirange, 10 * Log10[RCSSolEregcylTMplusTEAll]}];
```

Changing Phi Polar Plots

Table of Parameters


```
In[454]:= tablevalues = {{N[E0] "V/m", "-", "-", "-", "-"},
     {N[H0] "A/m", "-", "-", "-", "-"}, {f0 "Hz", "-", "-", "-", "-"},
     {lambda0 "m", "-", "-", "-", "-"}, {a / lambda0 "λ", "-", "-", "-", "-"},
     {b / lambda0 "λ", "-", "-", "-", "-"}, {ρ1 / lambda0 "λ", "-", "-", "-", "-"},
     {ρ2 / lambda0 "λ", "-", "-", "-", "-"}, {"-", N[(phirange[[1]]) * (180 / Pi)] "Deg",
      N[(phirange[[Length[phirange]]]) * (180 / Pi)] "Deg",
      N[(phidelta) * (180 / Pi)] "Deg", Length[phirange]}, {ρ / lambda0 "λ" , "-",
      "-", "-", "-"}, {zpoint / a "a" && zpoint / lambda0 "λ" , "-", "-", "-", "-"},
     {"-" , "-", "-", "-", jqty}, {N[θi * (180 / Pi)] "Deg", "-", "-", "-", "-"},
     {N[ϕi * (180 / Pi)] "Deg", "-", "-", "-", "-"},
     {"-", nmin, nmax, "1", nqty}, {"-", lmin, lmax, "1", lqty},
     {"-", mmin, mmax, "1", lqty}, {mmaxcheck, "-", "-", "-", "-"},
     {lmaxcheck, "-", "-", "-", "-"}, {"Boundary a+b", "-", "-", "-", "-"}};
   tablerowheading = {"E0", "H0", "λ", "Frequency", "a", "b", "ρ1", "ρ2",
     "ϕ range", "ρ (observed)", "z (observed) ", "Matching Points", "θi",
     "ϕi", "n", "l", "m", "max allowable m", "max allowable l", "Boundary"};
   tablecolheading = {"value", "min", "max", "delta", "Qty of Points"};

In[457]:= ChangingPhiTable = Grid[ArrayFlatten[{{{{" "}}, {tablecolheading}},
      {List /@tablerowheading, ArrayFlatten[tablevalues]}}],
    ItemStyle → {Bold, 20}, Frame → All, Background → {{LightGray}, {LightGray}}]

In[458]:= Export[ToString[StringForm["``.jpg", PhiTableName]], ChangingPhiTable]
```

Corrugated Cylinder Plots

Plot Ez (Scattered)

```
In[459]:= EzCorrS = ListPolarPlot[dataEzS, Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Ez Corrugated Cylinder (Scattered Field)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzS]}];
```

Plot Ez (Inc + Scattered)

```
In[460]:= EzCorrSandI = ListPolarPlot[dataEz, Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Ez Corrugated Cylinder (Incident + Scattered Field)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[SolEz]}];
```


Plot Erho (Scattered)

```
In[461]:= ErhoCorrS = ListPolarPlot[dataErhoS, Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eρ Corrugated Cylinder (Scattered Field)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoS]}];
```

Plot Erho (Inc + Scattered)

```
In[462]:= ErhoCorrSandI = ListPolarPlot[dataErho, Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eρ Corrugated Cylinder (Incident + Scattered Field)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[SolErho]}];
```

Plot Ephi (Scattered)

```
In[463]:= EphiCorrS = ListPolarPlot[dataEphiS, Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eϕ Corrugated Cylinder (Scattered Field)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiS]}];
```

Plot Ephi (Inc + Scattered)

```
In[464]:= EphiCorrSandI = ListPolarPlot[dataEphi, Joined → True, PolarGridLines → Automatic,
     PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eϕ Corrugated Cylinder (Incident + Scattered Field)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphi]}];
```

Plot EAll (Scattered)

```
In[465]:= EAllCorrS = ListPolarPlot[dataEAllS, Joined → True, PolarGridLines → Automatic,
    PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_Total Corrugated Cylinder (Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllS]}];
```

Plot EAll (Inc + Scattered)

```
In[466]:= EAllCorrSandI =
   ListPolarPlot[dataEAllSandI, Joined → True, PolarGridLines → Automatic,
    PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Red, Dashed},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_Total Corrugated Cylinder (Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllSandI]}];
```

Summary Corrugated Cylinder Plots

```
In[467]:= GraphicsGrid[
   {{EzCorrS, EzCorrSandI}, {ErhoCorrS, ErhoCorrSandI}, {EphiCorrS, EphiCorrSandI},
    {EAllCorrS, EAllCorrSandI}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TM Plots

Ez (TM Scattered)

```
In[468]:= EzRegCylTMS = ListPolarPlot[dataEzregcylTMS, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_z Smooth Cylinder (TM Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTMS]}];
```


Ez (TM Inc + Scattered)

```
In[469]:= EzRegCylTMSandI = ListPolarPlot[dataEzregcylTMSandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_z Smooth Cylinder (TM Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTMSandI]}];
```

Erho (TM Scattered)

```
In[470]:= ErhoRegCylTMS = ListPolarPlot[dataErhoregcylTMS, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_ρ Smooth Cylinder (TM Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTMS]}];
```

Erho (TM Inc + Scattered)

```
In[471]:= ErhoRegCylTMSandI = ListPolarPlot[dataErhoregcylTMSandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_ρ Smooth Cylinder (TM Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTMSandI]}];
```

Ephi (TM Scattered)

```
In[472]:= EphiRegCylTMS = ListPolarPlot[dataEphiregcylTMS, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_ϕ Smooth Cylinder (TM Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTMS]}];
```


Ephi (TM Inc + Scattered)

```
In[473]:= EphiRegCylTMSandI = ListPolarPlot[dataEphiregcylTMSandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_ϕ Smooth Cylinder (TM Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTMSandI]}];
```

EAll(TM Scattered)

```
In[474]:= EAllRegCylTMS = ListPolarPlot[dataEAllTMS, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_Total Smooth Cylinder (TM Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTMS]}];
```

EAll (TM Inc + Scattered)

```
In[475]:= EAllRegCylTMSandI = ListPolarPlot[dataEAllTMSandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_Total Smooth Cylinder (TM Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTMSandI]}];
```

Summary TM Plots

```
In[476]:= GraphicsGrid[{{EzRegCylTMS, EzRegCylTMSandI},
    {ErhoRegCylTMS, ErhoRegCylTMSandI}, {EphiRegCylTMS, EphiRegCylTMSandI},
    {EAllRegCylTMS, EAllRegCylTMSandI}}, Spacings -> {Scaled[0], Scaled[0]}];
```


Smooth Cylinder Plots - TE Plots

Ez (TE Scattered)

```
In[477]:= EzRegCylTES = ListPolarPlot[dataEzregcylTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_z Smooth Cylinder (TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTES]}];
```

Ez (TE Inc + Scattered)

```
In[478]:= EzRegCylTESandI = ListPolarPlot[dataEzregcylTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_z Smooth Cylinder (TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTESandI]}];
```

Erho (TE Scattered)

```
In[479]:= ErhoRegCylTES = ListPolarPlot[dataErhoregcylTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_ρ Smooth Cylinder (TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTES]}];
```

Erho (TE Inc + Scattered)

```
In[480]:= ErhoRegCylTESandI = ListPolarPlot[dataErhoregcylTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_ρ Smooth Cylinder (TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTESandI]}];
```

Ephi (TE Scattered)

```
In[481]:= EphiRegCylTES = ListPolarPlot[dataEphiregcylTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTES]}];
```

Ephi (TE Inc + Scattered)

```
In[482]:= EphiRegCylTESandI = ListPolarPlot[dataEphiregcylTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTESandI]}];
```

EAll (TE Scattered)

```
In[483]:= EAllRegCylTES = ListPolarPlot[dataEAllTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTES]}];
```

EAll (TE Inc + Scattered)

```
In[484]:= EAllRegCylTESandI = ListPolarPlot[dataEAllTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTESandI]}];
```

Summary TE Plots

```
In[485]:= GraphicsGrid[{{EzRegCylTES, EzRegCylTESandI},
    {ErhoRegCylTES, ErhoRegCylTESandI}, {EphiRegCylTES, EphiRegCylTESandI},
    {EAllRegCylTES, EAllRegCylTESandI}}, Spacings -> {Scaled[0], Scaled[0]}];
```


Smooth Cylinder Plots - TM + TE Plots

Ez (TM+TE Scattered)

```
In[486]:= EzRegCylTMplusTES = ListPolarPlot[dataEzregcylTMplusTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM + TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTMplusTES]}];
```

Ez (TM+TE Inc + Scattered)

```
In[487]:= EzRegCylTMplusTESandI = ListPolarPlot[dataEzregcylTMplusTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM + TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTMplusTESandI]}];
```

Erho (TM+TE Scattered)

```
In[488]:= ErhoRegCylTMplusTES = ListPolarPlot[dataErhoregcylTMplusTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM + TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTMplusTES]}];
```

Erho (TM+TE Inc + Scattered)

```
In[489]:= ErhoRegCylTMplusTESandI = ListPolarPlot[dataErhoregcylTMplusTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM + TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTMplusTESandI]}];
```


Ephi (TM+TE Scattered)

```
In[490]:= EphiRegCylTMplusTES = ListPolarPlot[dataEphiregcylTMplusTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TM + TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTMplusTES]}];
```

Ephi (TM+TE Inc + Scattered)

```
In[491]:= EphiRegCylTMplusTESandI = ListPolarPlot[dataEphiregcylTMplusTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TM + TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTMplusTESandI]}];
```

EAll (TM+TE Scattered)

```
In[492]:= EAllRegCylTMplusTES = ListPolarPlot[dataEAllTMplusTES, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM + TE Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTMplusTES]}];
```

EAll (TM+TE Inc + Scattered)

```
In[493]:= EAllRegCylTMplusTESandI = ListPolarPlot[dataEAllTMplusTESandI, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM + TE Incident + Scattered Field)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTMplusTESandI]}];
```


Summary TM+TE Plots

```
In[494]:= GraphicsGrid[{{EzRegCylTMplusTES, EzRegCylTMplusTESandI},
    {ErhoRegCylTMplusTES, ErhoRegCylTMplusTESandI},
    {EphiRegCylTMplusTES, EphiRegCylTMplusTESandI},
    {EAllRegCylTMplusTES, EAllRegCylTMplusTESandI}},
   Spacings -> {Scaled[0], Scaled[0]}];
```

Compare (Regular vs Corrugated) Plots

TM Compare

```
In[495]:= GraphicsGrid[{{ListPolarPlot[
      {dataEzregcylTMS, dataEzS}, Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
       "Ez TM Scattered Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTMS, SolEzS]}],

     ListPolarPlot[{dataEzregcylTMSandI, dataEz},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "Ez TM Incident + Scattered Smooth Cylinder (Blue)
         vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
      PolarAxesOrigin → {0, Max[SolEzregcylTMSandI, SolEz]}]},

    {ListPolarPlot[{dataErhoregcylTMS, dataErhoS},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
       "Eρ TM Scattered Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTMS, SolErhoS]}],

     ListPolarPlot[{dataErhoregcylTMSandI, dataErho},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "Eρ TM Incident + Scattered Smooth Cylinder (Blue)
```

```
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolErhoregcylTMSandI, SolErho]}]},

{ListPolarPlot[{dataEphiregcylTMS, dataEphiS},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
   "Eϕ TM Scattered Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTMS, SolEphiS]}],

 ListPolarPlot[{dataEphiregcylTMSandI, dataEphi},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "Eϕ TM Incident + Scattered Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEphiregcylTMSandI, SolEphi]}]},

{ListPolarPlot[{dataEAllTMS, dataEAllS}, Joined → True,
  PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
  PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
  PlotMarkers -> None, PlotLabel -> "ETotal TM Scattered Smooth
    Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTMS, SolEAllS]}],

 ListPolarPlot[{dataEAllTMSandI, dataEAllSandI},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "ETotal TM Incident + Scattered Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEAllTMSandI, SolEAllSandI]}]}
}];
```

TE Compare

```
GraphicsGrid[{{ListPolarPlot[
    {dataEzregcylTES, dataEzS}, Joined → True, PolarGridLines → Automatic,
```



```
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
   "Ez TE Scattered Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTES, SolEzS]}],

 ListPolarPlot[{dataEzregcylTESandI, dataEz},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "Ez TE Incident + Scattered Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEzregcylTESandI, SolEz]}]},

{ListPolarPlot[{dataErhoregcylTES, dataErhoS},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
   "Eρ TE Scattered Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTES, SolErhoS]}],

 ListPolarPlot[{dataErhoregcylTESandI, dataErho},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "Eρ TE Incident + Scattered Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolErhoregcylTESandI, SolErho]}]},

{ListPolarPlot[{dataEphiregcylTES, dataEphiS},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
   "Eϕ TE Scattered Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTES, SolEphiS]}],

 ListPolarPlot[{dataEphiregcylTESandI, dataEphi},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
```



```
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "Eϕ TE Incident + Scattered Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEphiregcylTESandI, SolEphi]}]},

{ListPolarPlot[{dataEAllTES, dataEAllS}, Joined → True,
  PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
  PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
  PlotMarkers -> None, PlotLabel -> "ETotal TE Scattered Smooth
    Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
  PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTES, SolEAllS]}],

 ListPolarPlot[{dataEAllTESandI, dataEAllSandI},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "ETotal TE Incident + Scattered Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEAllTESandI, SolEAllSandI]}]}
}];
```

TM+TE Compare

```
In[497]:= GraphicsGrid[{{ListPolarPlot[
    {dataEzregcylTMplusTES, dataEzS}, Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "Ez TM+TE Scattered Smooth Cylinder (Blue) vs
    Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEzregcylTMplusTES, SolEzS]}],

 ListPolarPlot[{dataEzregcylTMplusTESandI, dataEz},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "Ez TM+TE Incident + Scattered Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEzregcylTMplusTESandI, SolEz]}]},
```



```
{ListPolarPlot[{dataErhoregcylTMplusTES, dataErhoS},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "Eρ TM+TE Scattered Smooth Cylinder (Blue) vs
    Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolErhoregcylTMplusTES, SolErhoS]}],

 ListPolarPlot[{dataErhoregcylTMplusTESandI, dataErho},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "Eρ TM+TE Incident + Scattered Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolErhoregcylTMplusTESandI, SolErho]}]},

{ListPolarPlot[{dataEphiregcylTMplusTES, dataEphiS},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "Eϕ TM+TE Scattered Smooth Cylinder (Blue) vs
    Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEphiregcylTMplusTES, SolEphiS]}],

 ListPolarPlot[{dataEphiregcylTMplusTESandI, dataEphi},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "Eϕ TM+TE Incident + Scattered Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEphiregcylTMplusTESandI, SolEphi]}]},

{ListPolarPlot[{dataEAllTMplusTES, dataEAllS},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "ETotal TM+TE Scattered Smooth Cylinder (Blue)
    vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
```

```
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEAllTMplusTES, SolEAllS]}],

 ListPolarPlot[{dataEAllTMplusTESandI, dataEAllSandI},
  Joined → True, PolarGridLines → Automatic,
  PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
  BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
  PlotLabel -> "ETotal TM+TE Incident + Scattered Smooth Cylinder (Blue)
     vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
  BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
  PolarAxesOrigin → {0, Max[SolEAllTMplusTESandI, SolEAllSandI]}]}
}];
```

TM, TE, and TM+TE Compare

```
PhiPolarPlots =
 GraphicsGrid[{{ListPolarPlot[{dataEzregcylTMS, dataEzS}, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Ez TM Scattered (V/m) Smooth
       Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEzregcylTMS, SolEzS]}],
   "Ez TE Does Not Exist",
   ListPolarPlot[{dataEzregcylTMplusTES, dataEzS},
    Joined → True, PolarGridLines → Automatic,
    PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez TM+TE (V/m) Scattered Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[SolEzregcylTMplusTES, SolEzS]}]},


  {ListPolarPlot[{dataEzregcylTMSandI, dataEz},
    Joined → True, PolarGridLines → Automatic,
    PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez TM Incident + Scattered (V/m) Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[SolEzregcylTMSandI, SolEz]}],
   "Ez TE Does Not Exist",
```



```
   ListPolarPlot[{dataEzregcylTMplusTESandI, dataEz},
    Joined → True, PolarGridLines → Automatic,
    PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez TM+TE Incident + Scattered (V/m) Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[SolEzregcylTMplusTESandI, SolEz]}]},


  {ListPolarPlot[{dataErhoregcylTMS, dataErhoS},
    Joined → True, PolarGridLines → Automatic,
    PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ TM Scattered (V/m) Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[SolErhoregcylTMS, SolErhoS]}],
   ListPolarPlot[{dataErhoregcylTES, dataErhoS}, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Eρ TE Scattered (V/m) Smooth
       Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolErhoregcylTES, SolErhoS]}],
   ListPolarPlot[{dataErhoregcylTMplusTES, dataErhoS}, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> {Blue, {Red, Dashed}},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ TM+TE Scattered (V/m) Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[SolErhoregcylTMplusTES, SolErhoS]}]},

  {ListPolarPlot[{dataErhoregcylTMSandI, dataErho},
    Joined → True, PolarGridLines → Automatic,
    PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ TM Incident + Scattered (V/m) Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[SolErhoregcylTMSandI, SolErho]}],
   ListPolarPlot[{dataErhoregcylTESandI, dataErho}, Joined → True,
```



```
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> {Blue, {Red, Dashed}},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ TE Incident + Scattered (V/m) Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[SolErhoregcylTESandI, SolErho]}],
   ListPolarPlot[{dataErhoregcylTMplusTESandI, dataErho},
    Joined → True, PolarGridLines → Automatic,
    PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ TM+TE Incident + Scattered (V/m) Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[SolErhoregcylTMplusTESandI, SolErho]}]},


  {ListPolarPlot[{dataEphiregcylTMS, dataEphiS},
    Joined → True, PolarGridLines → Automatic,
    PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ TM Scattered (V/m) Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[SolEphiregcylTMS, SolEphiS]}],
   ListPolarPlot[{dataEphiregcylTES, dataEphiS}, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Eϕ TE Scattered (V/m) Smooth
       Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEphiregcylTES, SolEphiS]}],
   ListPolarPlot[{dataEphiregcylTMplusTES, dataEphiS}, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> {Blue, {Red, Dashed}},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ TM+TE Scattered (V/m) Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[SolEphiregcylTMplusTES, SolEphiS]}]},


  {ListPolarPlot[{dataEphiregcylTMSandI, dataEphi},
```



```
    Joined → True, PolarGridLines → Automatic,
    PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ TM Incident + Scattered (V/m) Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[SolEphiregcylTMSandI, SolEphi]}],
   ListPolarPlot[{dataEphiregcylTESandI, dataEphi}, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> {Blue, {Red, Dashed}},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ TE Incident + Scattered (V/m) Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[SolEphiregcylTESandI, SolEphi]}],
   ListPolarPlot[{dataEphiregcylTMplusTESandI, dataEphi},
    Joined → True, PolarGridLines → Automatic,
    PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ TM+TE Incident + Scattered (V/m) Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[SolEphiregcylTMplusTESandI, SolEphi]}]},


  {ListPolarPlot[{dataEAllTMS, dataEAllS}, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "ETotal TM Scattered (V/m) Smooth
       Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTMS, SolEAllS]}],
   ListPolarPlot[{dataEAllTES, dataEAllS}, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "ETotal TE Scattered (V/m) Smooth
       Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTES, SolEAllS]}],
   ListPolarPlot[{dataEAllTMplusTES, dataEAllS}, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "ETotal TM+TE Scattered (V/m)
```

PhD Research Garcia_Boundary a_plus_b.nb | 57

```
        Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0, Max[SolEAllTMplusTES, SolEAllS]}]},


  {ListPolarPlot[{dataEAllTMSandI, dataEAllSandI},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "ETotal TM Incident + Scattered (V/m) Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
      PolarAxesOrigin → {0, Max[SolEAllTMSandI, SolEAllSandI]}],
    ListPolarPlot[{dataEAllTESandI, dataEAllSandI}, Joined → True,
      PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
      PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "ETotal TE Incident + Scattered (V/m) Smooth Cylinder
         (Blue) vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
      PolarAxesOrigin → {0, Max[SolEAllTESandI, SolEAllSandI]}],

    ListPolarPlot[{dataEAllTMplusTESandI, dataEAllSandI},
       Joined → True, PolarGridLines → Automatic,
       PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
       BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
       PlotLabel -> "ETotal TM+TE Incident + Scattered (V/m) Smooth Cylinder
          (Blue) vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
       BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
       PolarAxesOrigin → {0, Max[SolEAllTMplusTESandI, SolEAllSandI]}];
  }
 }]
Export[ToString[StringForm["``.jpg", PhiPolarPlotsName]], PhiPolarPlots]
```

Changing Phi RCS Polar Plots

Corrugated Cylinder Plots

Plot Ez

Printed by Wolfram Mathematica Student Edition

58 | PhD Research Garcia_Boundary a_plus_b.nb

```
In[500]:= RCSEzCorr = ListPolarPlot[RCSdataEz, Joined → True,
     PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
     PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Ez Corrugated Cylinter (RCS m²)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEz]}];
```

Plot Erho

```
In[501]:= RCSErhoCorr = ListPolarPlot[RCSdataErho, Joined → True,
     PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
     PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Eρ Corrugated Cylinter (RCS m²)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolErho]}];
```

Plot Ephi

```
In[502]:= RCSEphiCorr = ListPolarPlot[RCSdataEphi, Joined → True,
     PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
     PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Eϕ Corrugated Cylinter (RCS m²)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEphi]}];
```

Plot EAll

```
In[503]:= RCSEAllCorr = ListPolarPlot[RCSdataEAll, Joined → True,
     PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
     PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "ETotal Corrugated Cylinter (RCS m²)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEAll]}];
```

Summary Corrugated Cylinder Plots

```
In[504]:= GraphicsGrid[{{RCSEzCorr}, {RCSErhoCorr}, {RCSEphiCorr}, {RCSEAllCorr}},
    Spacings -> {Scaled[0], Scaled[0]}];
```

Printed by Wolfram Mathematica Student Edition

PhD Research Garcia_Boundary a_plus_b.nb | 59

Smooth Cylinder Plots - TM Plots

Ez (TM RCS)

```
In[505]:= RCSEzRegCylTM = ListPolarPlot[RCSdataEzregcylTM, Joined → True,
     PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Ez Smooth Cylinder (TM RCS m²)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEzregcylTM]}];
```

Erho (TM RCS)

```
In[506]:= RCSErhoRegCylTM = ListPolarPlot[RCSdataErhoregcylTM, Joined → True,
     PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Eρ Smooth Cylinder (TM RCS m²)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolErhoregcylTM]}];
```

Ephi (TM RCS)

```
In[507]:= RCSEphiRegCylTM = ListPolarPlot[RCSdataEphiregcylTM, Joined → True,
     PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Eϕ Smooth Cylinder (TM RCS m²)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEphiregcylTM]}];
```

EAll (TM RCS)

```
In[508]:= RCSEAllRegCylTM = ListPolarPlot[RCSdataEregcylTMAll, Joined → True,
     PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "ETotal Smooth Cylinder (TM RCS m²)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEregcylTMAll]}];
```

Summary TM Plots

```
In[509]:= GraphicsGrid[{{RCSEzRegCylTM}, {RCSErhoRegCylTM}, {RCSEphiRegCylTM},
     {RCSEAllRegCylTM}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Printed by Wolfram Mathematica Student Edition

60 | PhD Research Garcia_Boundary a_plus_b.nb

Smooth Cylinder Plots - TE Plots

Ez (TE RCS) - DOES NOT EXIST

Erho (TE RCS)

```
In[510]:= RCSErhoRegCylTE = ListPolarPlot[RCSdataErhoregcylTE, Joined → True,
     PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Eρ Smooth Cylinder (TE RCS m²)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolErhoregcylTE]}];
```

Ephi (TE RCS)

```
In[511]:= RCSEphiRegCylTE = ListPolarPlot[RCSdataEphiregcylTE, Joined → True,
     PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Eϕ Smooth Cylinder (TE RCS m²)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEphiregcylTE]}];
```

EAll (TE RCS)

```
In[512]:= RCSEAllRegCylTE = ListPolarPlot[RCSdataEregcylTEAll, Joined → True,
     PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
     PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "ETotal Smooth Cylinder (TE RCS m²)",
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEregcylTEAll]}];
```

Summary TE Plots

```
In[513]:= GraphicsGrid[{{RCSErhoRegCylTE}, {RCSEphiRegCylTE}, {RCSEAllRegCylTE}},
    Spacings -> {Scaled[0], Scaled[0]}];
```

Printed by Wolfram Mathematica Student Edition

PhD Research Garcia_Boundary a_plus_b.nb | 61

Smooth Cylinder Plots - TM + TE Plots

Ez (TM + TE RCS)

```
RCSEzRegCylTMplusTE = ListPolarPlot[RCSdataEzregcylTMplusTE, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Ez Smooth Cylinder (TM + TE RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEzregcylTMplusTE]}];
```

Erho (TM + TE RCS)

```
RCSErhoRegCylTMplusTE = ListPolarPlot[RCSdataErhoregcylTMplusTE, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Eρ Smooth Cylinder (TM + TE RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolErhoregcylTMplusTE]}];
```

Ephi (TM + TE RCS)

```
RCSEphiRegCylTMplusTE = ListPolarPlot[RCSdataEphiregcylTMplusTE, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Eϕ Smooth Cylinder (TM + TE RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEphiregcylTMplusTE]}];
```

EAll (TM + TE RCS)

```
In[517]:= RCSEAllRegCylTMplusTE = ListPolarPlot[RCSdataEregcylTMplusTEAll, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
   PlotLabel -> "ETotal Smooth Cylinder (TM + TE RCS m²)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEregcylTMplusTEAll]}];
```

Printed by Wolfram Mathematica Student Edition

62 | PhD Research Garcia_Boundary a_plus_b.nb

Summary TM + TE Plots

```
In[518]:= GraphicsGrid[
   {{RCSEzRegCylTMplusTE}, {RCSErhoRegCylTMplusTE}, {RCSEphiRegCylTMplusTE},
    {RCSEAllRegCylTMplusTE}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Compare (Regular vs Corrugated) Plots

TM, TE, and TM+TE Compare

```
In[519]:= PhiRCSPolarPlots =
  GraphicsGrid[{{ListPolarPlot[{RCSdataEzregcylTM, RCSdataEz}, Joined → True,
      PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
      PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
       "Ez TM RCS (m²) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEzregcylTM, RCSSolEz]}]
     ,
     "Ez TE Does Not Exist"
     ,
     ListPolarPlot[{RCSdataEzregcylTMplusTE, RCSdataEz},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "Ez TM+TE RCS (m²) Smooth Cylinder (Blue) vs
         Corrugated Cylinder (Red, --)", ImageSize -> 700,
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
      PolarAxesOrigin → {0, Max[RCSSolEzregcylTMplusTE, RCSSolEz]}]},

    {ListPolarPlot[{RCSdataErhoregcylTM, RCSdataErho},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
       "Eρ TM RCS (m²) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolErhoregcylTM, RCSSolErho]}]
     ,
     ListPolarPlot[{RCSdataErhoregcylTE, RCSdataErho},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
```

Printed by Wolfram Mathematica Student Edition

PhD Research Garcia_Boundary a_plus_b.nb | 63

```
       "Eρ TE RCS (m²) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolErhoregcylTE, RCSSolErho]}]
     ,
     ListPolarPlot[{RCSdataErhoregcylTMplusTE, RCSdataErho},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "Eρ TM+TE RCS (m²) Smooth Cylinder (Blue) vs
         Corrugated Cylinder (Red, --)", ImageSize -> 700,
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
      PolarAxesOrigin → {0, Max[RCSSolErhoregcylTMplusTE, RCSSolErho]}]
    },

    {ListPolarPlot[{RCSdataEphiregcylTM, RCSdataEphi},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
       "Eϕ TM RCS (m²) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEphiregcylTM, RCSSolEphi]}]
     ,
     ListPolarPlot[{RCSdataEphiregcylTE, RCSdataEphi},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
       "Eϕ TE RCS (m²) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0, Max[RCSSolEphiregcylTE, RCSSolEphi]}]
     ,
     ListPolarPlot[{RCSdataEphiregcylTMplusTE, RCSdataEphi},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "Eϕ TM+TE RCS (m²) Smooth Cylinder (Blue) vs
         Corrugated Cylinder (Red, --)", ImageSize -> 700,
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
      PolarAxesOrigin → {0, Max[RCSSolEphiregcylTMplusTE, RCSSolEphi]}]
    },

    {ListPolarPlot[{RCSdataEregcylTMAll, RCSdataEAll},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
```

Printed by Wolfram Mathematica Student Edition

64 | PhD Research Garcia_Boundary a_plus_b.nb

```
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "ETotal TM RCS (m²) Smooth Cylinder (Blue) vs
         Corrugated Cylinder (Red, --)", ImageSize -> 700,
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
      PolarAxesOrigin → {0, Max[RCSSolEregcylTMAll, RCSSolEAll]}]
     ,
     ListPolarPlot[{RCSdataEregcylTEAll, RCSdataEAll},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "ETotal TE RCS (m²) Smooth Cylinder (Blue) vs
         Corrugated Cylinder (Red, --)", ImageSize -> 700,
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
      PolarAxesOrigin → {0, Max[RCSSolEregcylTEAll, RCSSolEAll]}]
     ,
     ListPolarPlot[{RCSdataEregcylTMplusTEAll, RCSdataEAll},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
      PlotLabel -> "ETotal TM+TE RCS (m²) Smooth Cylinder (Blue)
         vs Corrugated Cylinder (Red, --)", ImageSize -> 700,
      BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
      PolarAxesOrigin → {0, Max[RCSSolEregcylTMplusTEAll, RCSSolEAll]}]
    }

   }]
In[520]:= Export[ToString[StringForm["``.jpg", PhiRCSPolarPlotsName]], PhiRCSPolarPlots];
```

Changing Phi RCS dB Polar Plots

Corrugated Cylinder Plots

Plot Ez

```
In[521]:= RCSEzCorrdB = ListPolarPlot[RCSdataEzdB, Joined → True,
   PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
   PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Ez Corrugated Cylinder (RCS dBsm)",
   ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   PolarAxes → True, PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEz]]]}];
```

Printed by Wolfram Mathematica Student Edition

Plot Erho

```
In[522]:= RCSErhoCorrdB = ListPolarPlot[RCSdataErhodB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Eρ Corrugated Cylinder (RCS dBsm)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolErho]]]}];
```

Plot Ephi

```
In[523]:= RCSEphiCorrdB = ListPolarPlot[RCSdataEphidB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Eφ Corrugated Cylinder (RCS dBsm)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEphi]]]}];
```

Plot EAll

```
In[524]:= RCSEAllCorrdB = ListPolarPlot[RCSdataEAlldB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "ETotal Corrugated Cylinder (RCS dBsm)",
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    PolarAxes → True, PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEAll]]]}];
```

Summary Corrugated Cylinder Plots

```
In[525]:= GraphicsGrid[{{RCSEzCorrdB}, {RCSErhoCorrdB}, {RCSEphiCorrdB}, {RCSEAllCorrdB}},
   Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TM Plots

Ez (TM RCS)

```
In[526]:= RCSEzRegCylTMdB = ListPolarPlot[RCSdataEzregcylTMdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEzregcylTM]]]}];
```


Erho (TM RCS)

```
In[527]:= RCSErhoRegCylTMdB = ListPolarPlot[RCSdataErhoregcylTMdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolErhoregcylTM]]]}];
```

Ephi (TM RCS)

```
In[528]:= RCSEphiRegCylTMdB = ListPolarPlot[RCSdataEphiregcylTMdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eφ Smooth Cylinder (TM RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEphiregcylTM]]]}];
```

EAll (TM RCS)

```
In[529]:= RCSEAllRegCylTMdB = ListPolarPlot[RCSdataEregcylTMAlldB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEregcylTMAll]]]}];
```

Summary TM Plots

```
In[530]:= GraphicsGrid[{{RCSEzRegCylTMdB}, {RCSErhoRegCylTMdB}, {RCSEphiRegCylTMdB},
    {RCSEAllRegCylTMdB}}, Spacings -> {Scaled[0], Scaled[0]}];
```


Smooth Cylinder Plots - TE Plots

Ez (TE RCS) - DOES NOT EXIST

Erho (TE RCS)

```
In[531]:= RCSErhoRegCylTEdB = ListPolarPlot[RCSdataErhoregcylTEdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TE RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolErhoregcylTE]]]}];
```

Ephi (TE RCS)

```
In[532]:= RCSEphiRegCylTEdB = ListPolarPlot[RCSdataEphiregcylTEdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eφ Smooth Cylinder (TE RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEphiregcylTE]]]}];
```

EAll (TE RCS)

```
In[533]:= RCSEAllRegCylTEdB = ListPolarPlot[RCSdataEregcylTEAlldB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TE RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEregcylTEAll]]]}];
```

Summary TE Plots

```
In[534]:= GraphicsGrid[{{RCSErhoRegCylTEdB}, {RCSEphiRegCylTEdB}, {RCSEAllRegCylTEdB}},
   Spacings -> {Scaled[0], Scaled[0]}];
```


Smooth Cylinder Plots - TM + TE Plots

Ez (TM + TE RCS)

```
RCSEzRegCylTMplusTEdB = ListPolarPlot[RCSdataEzregcylTMplusTEdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM + TE RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEzregcylTMplusTE]]]}];
```

Erho (TM + TE RCS)

```
RCSErhoRegCylTMplusTEdB = ListPolarPlot[RCSdataErhoregcylTMplusTEdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM + TE RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolErhoregcylTMplusTE]]]}];
```

Ephi (TM + TE RCS)

```
RCSEphiRegCylTMplusTEdB = ListPolarPlot[RCSdataEphiregcylTMplusTEdB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eφ Smooth Cylinder (TM + TE RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEphiregcylTMplusTE]]]}];
```

EAll (TM + TE RCS)

```
In[538]:= RCSEAllRegCylTMplusTEdB = ListPolarPlot[RCSdataEregcylTMplusTEAlldB, Joined → True,
    PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM + TE RCS dBsm)", ImageSize -> 700,
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, PolarAxes → True,
    PolarAxesOrigin → {0, Max[Abs[10 * Log10[RCSSolEregcylTMplusTEAll]]]}];
```

Summary TM + TE Plots

```
In[539]:= GraphicsGrid[
  {{RCSEzRegCylTMplusTEdB}, {RCSErhoRegCylTMplusTEdB}, {RCSEphiRegCylTMplusTEdB},
   {RCSEAllRegCylTMplusTEdB}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Compare (Regular vs Corrugated) Plots

TM, TE, and TM+TE Compare

```
In[540]:= PhiRCSdBPolarPlots =
  GraphicsGrid[{{ListPolarPlot[{RCSdataEzregcylTMdB, RCSdataEzdB},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
       "Ez TM RCS (dBsm) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin →
       {0, Max[Abs[10 * Log10[RCSSolEzregcylTM]], Abs[10 * Log10[RCSSolEz]]]}]
     ,
     "Ez TE Does Not Exist"
     ,
     ListPolarPlot[{RCSdataEzregcylTMplusTEdB, RCSdataEzdB}, Joined → True,
      PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
      PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
      PlotMarkers -> None, PlotLabel -> "Ez TM+TE RCS (dBsm) Smooth
         Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0,
        Max[Abs[10 * Log10[RCSSolEzregcylTMplusTE]], Abs[10 * Log10[RCSSolEz]]]}]},

    {ListPolarPlot[{RCSdataErhoregcylTMdB, RCSdataErhodB},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
       "Eρ TM RCS (dBsm) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin →
       {0, Max[Abs[10 * Log10[RCSSolErhoregcylTM]], Abs[10 * Log10[RCSSolErho]]]}]
     ,
     ListPolarPlot[{RCSdataErhoregcylTEdB, RCSdataErhodB},
      Joined → True, PolarGridLines → Automatic,
```



```
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
       "Eρ TE RCS (dBsm) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin →
       {0, Max[Abs[10 * Log10[RCSSolErhoregcylTE]], Abs[10 * Log10[RCSSolErho]]]}]
     ,
     ListPolarPlot[{RCSdataErhoregcylTMplusTEdB, RCSdataErhodB}, Joined → True,
      PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
      PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
      PlotMarkers -> None, PlotLabel -> "Eρ TM+TE RCS (dBsm) Smooth
         Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0,
        Max[Abs[10 * Log10[RCSSolErhoregcylTMplusTE]], Abs[10 * Log10[RCSSolErho]]]}]
    },

    {ListPolarPlot[{RCSdataEphiregcylTMdB, RCSdataEphidB},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
       "Eϕ TM RCS (dBsm) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin →
       {0, Max[Abs[10 * Log10[RCSSolEphiregcylTM]], Abs[10 * Log10[RCSSolEphi]]]}]
     ,
     ListPolarPlot[{RCSdataEphiregcylTEdB, RCSdataEphidB},
      Joined → True, PolarGridLines → Automatic,
      PolarTicks → {"Degrees", Automatic}, PlotStyle -> {Blue, {Red, Dashed}},
      BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None, PlotLabel ->
       "Eϕ TE RCS (dBsm) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin →
       {0, Max[Abs[10 * Log10[RCSSolEphiregcylTE]], Abs[10 * Log10[RCSSolEphi]]]}]
     ,
     ListPolarPlot[{RCSdataEphiregcylTMplusTEdB, RCSdataEphidB}, Joined → True,
      PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
      PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
      PlotMarkers -> None, PlotLabel -> "Eϕ TM+TE RCS (dBsm) Smooth
         Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0,
        Max[Abs[10 * Log10[RCSSolEphiregcylTMplusTE]], Abs[10 * Log10[RCSSolEphi]]]}]
```



```
    },

    {ListPolarPlot[{RCSdataEregcylTMAlldB, RCSdataEAlldB}, Joined → True,
      PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
      PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
      PlotMarkers -> None, PlotLabel -> "ETotal TM RCS (dBsm) Smooth
         Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin →
       {0, Max[Abs[10 * Log10[RCSSolEregcylTMAll]], Abs[10 * Log10[RCSSolEAll]]]}]
     ,
     ListPolarPlot[{RCSdataEregcylTEAlldB, RCSdataEAlldB}, Joined → True,
      PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
      PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
      PlotMarkers -> None, PlotLabel -> "ETotal TE RCS (dBsm) Smooth
         Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin →
       {0, Max[Abs[10 * Log10[RCSSolEregcylTEAll]], Abs[10 * Log10[RCSSolEAll]]]}]
     ,
     ListPolarPlot[{RCSdataEregcylTMplusTEAlldB, RCSdataEAlldB}, Joined → True,
      PolarGridLines → Automatic, PolarTicks → {"Degrees", Automatic},
      PlotStyle -> {Blue, {Red, Dashed}}, BaseStyle -> AbsoluteThickness[2],
      PlotMarkers -> None, PlotLabel -> "ETotal TM+TE RCS (dBsm) Smooth
         Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
      PolarAxes → True, PolarAxesOrigin → {0,
        Max[Abs[10 * Log10[RCSSolEregcylTMplusTEAll]], Abs[10 * Log10[RCSSolEAll]]]}]
    }

   }]
  Export[ToString[StringForm["``.jpg", PhiRCSdBPolarPlotsName]], PhiRCSdBPolarPlots]
```

Changing Phi RCS dB XY Plots

Corrugated Cylinder Plots

Plot Ez


```
In[542]:= RCSEzCorrdBxy = ListLinePlot[RCSdataEzdB,
    PlotRange -> {{0, 2 Pi}, {Min[10 * Log10[RCSSolEz]], Max[10 * Log10[RCSSolEz]]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Ez Corrugated Cylinder (RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot Erho

```
In[543]:= RCSErhoCorrdBxy = ListLinePlot[RCSdataErhodB, PlotRange ->
     {{0, 2 Pi}, {Min[10 * Log10[RCSSolErho]], Max[10 * Log10[RCSSolErho]]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Eρ Corrugated Cylinder (RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot Ephi

```
In[544]:= RCSEphiCorrdBxy = ListLinePlot[RCSdataEphidB, PlotRange ->
     {{0, 2 Pi}, {Min[10 * Log10[RCSSolEphi]], Max[10 * Log10[RCSSolEphi]]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "Eϕ Corrugated Cylinder (RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot EAll

```
In[545]:= RCSEAllCorrdBxy = ListLinePlot[RCSdataEAlldB, PlotRange ->
     {{0, 2 Pi}, {Min[10 * Log10[RCSSolEAll]], Max[10 * Log10[RCSSolEAll]]}},
    PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "ETotal Corrugated Cylinder (RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary Corrugated Cylinder Plots

```
In[546]:= GraphicsGrid[{{RCSEzCorrdBxy}, {RCSErhoCorrdBxy}, {RCSEphiCorrdBxy},
    {RCSEAllCorrdBxy}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TM Plots

Ez (TM RCS)

```
In[547]:= RCSEzRegCylTMdBxy = ListLinePlot[RCSdataEzregcylTMdB, PlotRange -> {{0, 2 Pi},
      {Min[10 * Log10[RCSSolEzregcylTM]], Max[10 * Log10[RCSSolEzregcylTM]]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "E_z Smooth Cylinder (TM RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TM RCS)

```
In[548]:= RCSErhoRegCylTMdBxy = ListLinePlot[RCSdataErhoregcylTMdB, PlotRange -> {{0, 2 Pi},
      {Min[10 * Log10[RCSSolErhoregcylTM]], Max[10 * Log10[RCSSolErhoregcylTM]]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "E_ρ Smooth Cylinder (TM RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TM RCS)

```
In[549]:= RCSEphiRegCylTMdBxy = ListLinePlot[RCSdataEphiregcylTMdB, PlotRange -> {{0, 2 Pi},
      {Min[10 * Log10[RCSSolEphiregcylTM]], Max[10 * Log10[RCSSolEphiregcylTM]]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "E_ϕ Smooth Cylinder (TM RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TM RCS)

```
In[550]:= RCSEAllRegCylTMdBxy = ListLinePlot[RCSdataEregcylTMAlldB, PlotRange -> {{0, 2 Pi},
      {Min[10 * Log10[RCSSolEregcylTMAll]], Max[10 * Log10[RCSSolEregcylTMAll]]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_Total Smooth Cylinder (TM RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Summary TM Plots

```
In[551]:= GraphicsGrid[{{RCSEzRegCylTMdBxy}, {RCSErhoRegCylTMdBxy}, {RCSEphiRegCylTMdBxy},
    {RCSEAllRegCylTMdBxy}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TE Plots

Ez (TE RCS) - DOES NOT EXIST

Erho (TE RCS)

```
In[552]:= RCSErhoRegCylTEdBxy = ListLinePlot[RCSdataErhoregcylTEdB, PlotRange -> {{0, 2 Pi},
      {Min[10 * Log10[RCSSolErhoregcylTE]], Max[10 * Log10[RCSSolErhoregcylTE]]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "E_ρ Smooth Cylinder (TE RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TE RCS)

```
In[553]:= RCSEphiRegCylTEdBxy = ListLinePlot[RCSdataEphiregcylTEdB, PlotRange -> {{0, 2 Pi},
      {Min[10 * Log10[RCSSolEphiregcylTE]], Max[10 * Log10[RCSSolEphiregcylTE]]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2],
    PlotMarkers -> None, PlotLabel -> "E_ϕ Smooth Cylinder (TE RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TE RCS)

```
In[554]:= RCSEAllRegCylTEdBxy = ListLinePlot[RCSdataEregcylTEAlldB, PlotRange -> {{0, 2 Pi},
      {Min[10 * Log10[RCSSolEregcylTEAll]], Max[10 * Log10[RCSSolEregcylTEAll]]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_Total Smooth Cylinder (TE RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Summary TE Plots

```
In[555]:= GraphicsGrid[{{RCSErhoRegCylTEdBxy}, {RCSEphiRegCylTEdBxy},
    {RCSEAllRegCylTEdBxy}}, Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TM + TE Plots

Ez (TM + TE RCS)

```
In[556]:= RCSEzRegCylTMplusTEdBxy = ListLinePlot[RCSdataEzregcylTMplusTEdB,
    PlotRange -> {{0, 2 Pi}, {Min[10 * Log10[RCSSolEzregcylTMplusTE]],
      Max[10 * Log10[RCSSolEzregcylTMplusTE]]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_z Smooth Cylinder (TM+TE RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TM + TE RCS)

```
In[557]:= RCSErhoRegCylTMplusTEdBxy = ListLinePlot[RCSdataErhoregcylTMplusTEdB,
    PlotRange -> {{0, 2 Pi}, {Min[10 * Log10[RCSSolErhoregcylTMplusTE]],
      Max[10 * Log10[RCSSolErhoregcylTMplusTE]]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_ρ Smooth Cylinder (TM+TE RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TM + TE RCS)

```
In[558]:= RCSEphiRegCylTMplusTEdBxy = ListLinePlot[RCSdataEphiregcylTMplusTEdB,
    PlotRange -> {{0, 2 Pi}, {Min[10 * Log10[RCSSolEphiregcylTMplusTE]],
      Max[10 * Log10[RCSSolEphiregcylTMplusTE]]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_ϕ Smooth Cylinder (TM+TE RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


EAll (TM + TE RCS)

```
In[559]:= RCSEAllRegCylTMplusTEdBxy = ListLinePlot[RCSdataEregcylTMplusTEAlldB,
    PlotRange -> {{0, 2 Pi}, {Min[10 * Log10[RCSSolEregcylTMplusTEAll]],
      Max[10 * Log10[RCSSolEregcylTMplusTEAll]]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "E_Total Smooth Cylinder (TM+TE RCS dBsm)",
    Frame -> True, FrameLabel -> {"ϕ", "RCS (dBsm)"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary TM + TE Plots

```
In[560]:= GraphicsGrid[{{RCSEzRegCylTMplusTEdBxy}, {RCSErhoRegCylTMplusTEdBxy},
    {RCSEphiRegCylTMplusTEdBxy}, {RCSEAllRegCylTMplusTEdBxy}},
   Spacings -> {Scaled[0], Scaled[0]}];
```

Compare (Regular vs Corrugated) Plots

TM, TE, and TM+TE Compare

```
In[561]:= PhiRCSdBXYPlots =
  GraphicsGrid[{{Show[{RCSEzRegCylTMdBxy, RCSEzCorrdBxy}, PlotRange ->
       {{0, 2 Pi}, {Min[10 * Log10[RCSSolEzregcylTM], 10 * Log10[RCSSolEz]],
         Max[10 * Log10[RCSSolEzregcylTM], 10 * Log10[RCSSolEz]]}}, PlotLabel ->
       "E_z TM RCS (dBsm) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
     ,
     "Ez TE RCS dB Does Not Exist"
     ,
     Show[{RCSEzRegCylTMplusTEdBxy, RCSEzCorrdBxy}, PlotRange ->
       {{0, 2 Pi}, {Min[10 * Log10[RCSSolEzregcylTMplusTE], 10 * Log10[RCSSolEz]],
         Max[10 * Log10[RCSSolEzregcylTMplusTE], 10 * Log10[RCSSolEz]]}},
      PlotLabel -> "E_z TM+TE RCS (dBsm) Smooth Cylinder (Blue)
         vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
    },
    {Show[{RCSErhoRegCylTMdBxy, RCSErhoCorrdBxy}, PlotRange ->
       {{0, 2 Pi}, {Min[10 * Log10[RCSSolErhoregcylTM], 10 * Log10[RCSSolErho]],
         Max[10 * Log10[RCSSolErhoregcylTM], 10 * Log10[RCSSolErho]]}}, PlotLabel ->
       "E_ρ TM RCS (dBsm) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
      ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
```

```
 ,
 Show[{RCSErhoRegCylTEdBxy, RCSErhoCorrdBxy}, PlotRange ->
   {{0, 2 Pi}, {Min[10 * Log10[RCSSolErhoregcylTE], 10 * Log10[RCSSolErho]],
     Max[10 * Log10[RCSSolErhoregcylTE], 10 * Log10[RCSSolErho]]}}, PlotLabel ->
   "Eρ TE RCS Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
 ,
 Show[{RCSErhoRegCylTMplusTEdBxy, RCSErhoCorrdBxy}, PlotRange -> {{0, 2 Pi},
    {Min[10 * Log10[RCSSolErhoregcylTMplusTE], 10 * Log10[RCSSolErho]],
     Max[10 * Log10[RCSSolErhoregcylTMplusTE], 10 * Log10[RCSSolErho]]}},
  PlotLabel -> "Eρ TM+TE RCS (dBsm) Smooth Cylinder (Blue)
     vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
},

{Show[{RCSEphiRegCylTMdBxy, RCSEphiCorrdBxy}, PlotRange ->
   {{0, 2 Pi}, {Min[10 * Log10[RCSSolEphiregcylTM], 10 * Log10[RCSSolEphi]],
     Max[10 * Log10[RCSSolEphiregcylTM], 10 * Log10[RCSSolEphi]]}}, PlotLabel ->
   "Eφ TM RCS (dBsm) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
 ,
 Show[{RCSEphiRegCylTEdBxy, RCSEphiCorrdBxy}, PlotRange ->
   {{0, 2 Pi}, {Min[10 * Log10[RCSSolEphiregcylTE], 10 * Log10[RCSSolEphi]],
     Max[10 * Log10[RCSSolEphiregcylTE], 10 * Log10[RCSSolEphi]]}}, PlotLabel ->
   "Eφ TE RCS (dBsm) Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
 ,
 Show[{RCSEphiRegCylTMplusTEdBxy, RCSEphiCorrdBxy}, PlotRange -> {{0, 2 Pi},
    {Min[10 * Log10[RCSSolEphiregcylTMplusTE], 10 * Log10[RCSSolEphi]],
     Max[10 * Log10[RCSSolEphiregcylTMplusTE], 10 * Log10[RCSSolEphi]]}},
  PlotLabel -> "Eφ TM+TE RCS (dBsm) Smooth Cylinder (Blue)
     vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
},

{Show[{RCSEAllRegCylTMdBxy, RCSEAllCorrdBxy}, PlotRange ->
   {{0, 2 Pi}, {Min[10 * Log10[RCSSolEregcylTMAll], 10 * Log10[RCSSolEAll]],
     Max[10 * Log10[RCSSolEregcylTMAll], 10 * Log10[RCSSolEAll]]}},
  PlotLabel -> "ETotal TM RCS (dBsm) Smooth Cylinder (Blue)
     vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
 ,
 Show[{RCSEAllRegCylTEdBxy, RCSEAllCorrdBxy}, PlotRange ->
```



```
   {{0, 2 Pi}, {Min[10 * Log10[RCSSolEregcylTEAll], 10 * Log10[RCSSolEAll]],
     Max[10 * Log10[RCSSolEregcylTEAll], 10 * Log10[RCSSolEAll]]}},
  PlotLabel -> "ETotal TE RCS (dBsm) Smooth Cylinder (Blue)
     vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
 ,
 Show[{RCSEAllRegCylTMplusTEdBxy, RCSEAllCorrdBxy}, PlotRange -> {{0, 2 Pi},
    {Min[10 * Log10[RCSSolEregcylTMplusTEAll], 10 * Log10[RCSSolEAll]],
     Max[10 * Log10[RCSSolEregcylTMplusTEAll], 10 * Log10[RCSSolEAll]]}},
  PlotLabel -> "ETotal TM+TE RCS (dBsm) Smooth Cylinder (Blue)
     vs Corrugated Cylinder (Red, --)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}]
 }

}]
In[562]:= Export[ToString[StringForm["``.jpg", PhiRCSdBXYPlotsName]], PhiRCSdBXYPlots]
Out[562]= PhiRCSdBXYPlotsName.jpg
```

RCS dBsm Data Export

```
In[563]:= tablevalues = {10 * Log10[RCSSolEz], 10 * Log10[RCSSolEzregcylTMplusTE],
     10 * Log10[RCSSolErho], 10 * Log10[RCSSolErhoregcylTMplusTE],
     10 * Log10[RCSSolEphi], 10 * Log10[RCSSolEphiregcylTMplusTE],
     10 * Log10[RCSSolEAll], 10 * Log10[RCSSolEregcylTMplusTEAll]};
In[564]:= tablecolheading =
    {{"EzCorrugCyl RMS dBsm", "EzSmoothCyl RMS dBsm", "ErhoCorrugCyl RMS dBsm",
      "ErhoSmoothCyl RMS dBsm", "EphiCorrugCyl RMS dBsm", "EphiSmoothCyl RMS dBsm",
      "EtotCorrugCyl RMS dBsm", "EtotSmoothCyl RMS dBsm"}};
In[565]:= Export[ToString[StringForm["``.xlsx", RCSdBdataSheetName]],
    Join[tablecolheading, Transpose[tablevalues]]]
Out[565]= RCSdBdataSheetName.xlsx
In[566]:= Speak["The polar plots are done"]
```


Changing Phi XY Plots

Corrugated Cylinder Plots

Plot Ez (Scattered)

```
In[567]:= EzCorrS = ListLinePlot[dataEzS, PlotRange -> {{0, 2 * Pi}, {0, Max[SolEzS]}},
     PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Ez Corrugated Cylinder (Scattered Field)",
     Frame -> True, FrameLabel -> {"φ", "V/m"},
     GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot Ez (Inc + Scattered)

```
In[568]:= EzCorrSandI = ListLinePlot[dataEz,
     PlotRange -> {{0, 2 * Pi}, {0, Max[SolEz]}}, PlotStyle -> {Red, Dashed},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Ez Corrugated Cylinder (Incident + Scattered Field)",
     Frame -> True, FrameLabel -> {"φ", "V/m"},
     GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot Erho (Scattered)

```
In[569]:= ErhoCorrS =
    ListLinePlot[dataErhoS, PlotRange -> {{0, 2 * Pi}, {0, Max[SolErhoS]}},
     PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Eρ Corrugated Cylinder (Scattered Field)",
     Frame -> True, FrameLabel -> {"φ", "V/m"},
     GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Plot Erho (Inc + Scattered)

```
In[570]:= ErhoCorrSandI = ListLinePlot[dataErho,
     PlotRange -> {{0, 2 * Pi}, {0, Max[SolErho]}}, PlotStyle -> {Red, Dashed},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eρ Corrugated Cylinder (Incident + Scattered Field)",
     Frame -> True, FrameLabel -> {"φ", "V/m"},
     GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot Ephi (Scattered)

```
In[571]:= EphiCorrS =
    ListLinePlot[dataEphiS, PlotRange -> {{0, 2 * Pi}, {0, Max[SolEphiS]}},
     PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
     PlotMarkers -> None, PlotLabel -> "Eφ Corrugated Cylinder (Scattered Field)",
     Frame -> True, FrameLabel -> {"φ", "V/m"},
     GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot Ephi (Inc + Scattered)

```
In[572]:= EphiCorrSandI = ListLinePlot[dataEphi,
     PlotRange -> {{0, 2 * Pi}, {0, Max[SolEphi]}}, PlotStyle -> {Red, Dashed},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "Eφ Corrugated Cylinder (Incident + Scattered Field)",
     Frame -> True, FrameLabel -> {"φ", "V/m"},
     GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot EAll (Scattered)

```
In[573]:= EAllCorrS = ListLinePlot[dataEAllS,
     PlotRange -> {{0, 2 * Pi}, {0, Max[SolEAllS]}}, PlotStyle -> {Red, Dashed},
     BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
     PlotLabel -> "ETotal Corrugated Cylinder (Scattered Field)",
     Frame -> True, FrameLabel -> {"φ", "V/m"},
     GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
     ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Plot EAll (Inc + Scattered)

```
In[574]:= EAllCorrSandI = ListLinePlot[dataEAllSandI,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolEAllSandI]}}, PlotStyle -> {Red, Dashed},
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Corrugated Cylinder (Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary TM Plots

```
In[575]:= GraphicsGrid[{{EzCorrS, EzCorrSandI}, {ErhoCorrS, ErhoCorrSandI},
    {EphiCorrS, EphiCorrSandI}, {EAllCorrS, EAllCorrSandI}},
   Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TM Plots

Ez (TM Scattered)

```
In[576]:= EzRegCylTMS = ListLinePlot[dataEzregcylTMS,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolEzregcylTMS]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ez (TM Inc + Scattered)

```
In[577]:= EzRegCylTMSandI = ListLinePlot[dataEzregcylTMSandI,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolEzregcylTMSandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Erho (TM Scattered)

```
In[578]:= ErhoRegCylTMS = ListLinePlot[dataErhoregcylTMS,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolErhoregcylTMS]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TM Inc + Scattered)

```
In[579]:= ErhoRegCylTMSandI = ListLinePlot[dataErhoregcylTMSandI,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolErhoregcylTMSandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TM Scattered)

```
In[580]:= EphiRegCylTMS = ListLinePlot[dataEphiregcylTMS,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolEphiregcylTMS]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TM Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TM Inc + Scattered)

```
In[581]:= EphiRegCylTMSandI = ListLinePlot[dataEphiregcylTMSandI,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolEphiregcylTMSandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TM Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


EAll (TM Scattered)

```
In[582]:= EAllRegCylTMS =
   ListLinePlot[dataEAllTMS, PlotRange -> {{0, 2 * Pi}, {0, Max[SolEAllTMS]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TM Inc + Scattered)

```
In[583]:= EAllRegCylTMSandI = ListLinePlot[dataEAllTMSandI,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolEAllTMSandI]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary TM Plots

```
GraphicsGrid[{{EzRegCylTMS, EzRegCylTMSandI}, {ErhoRegCylTMS, ErhoRegCylTMSandI},
    {EphiRegCylTMS, EphiRegCylTMSandI}, {EAllRegCylTMS, EAllRegCylTMSandI}},
   Spacings -> {Scaled[0], Scaled[0]}];
```

Smooth Cylinder Plots - TE Plots

Ez (TE Scattered)

```
In[585]:= EzRegCylTES = ListLinePlot[dataEzregcylTES,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolEzregcylTES]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TE Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Ez (TE Inc + Scattered)

```
In[586]:= EzRegCylTESandI = ListLinePlot[dataEzregcylTESandI,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolEzregcylTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TE Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TE Scattered)

```
In[587]:= ErhoRegCylTES = ListLinePlot[dataErhoregcylTES,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolErhoregcylTES]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TE Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TE Inc + Scattered)

```
In[588]:= ErhoRegCylTESandI = ListLinePlot[dataErhoregcylTESandI,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolErhoregcylTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TE Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TE Scattered)

```
In[589]:= EphiRegCylTES = ListLinePlot[dataEphiregcylTES,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolEphiregcylTES]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TE Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TE Inc + Scattered)

```
In[590]:= EphiRegCylTESandI = ListLinePlot[dataEphiregcylTESandI,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolEphiregcylTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TE Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TE Scattered)

```
In[591]:= EAllRegCylTES =
   ListLinePlot[dataEAllTES, PlotRange -> {{0, 2*Pi}, {0, Max[SolEAllTES]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder ((TE Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TE Inc + Scattered)

```
In[592]:= EAllRegCylTESandI = ListLinePlot[dataEAllTESandI,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolEAllTESandI]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TE Incident + Scattered Field)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Summary TE Plots

```
In[593]:= GraphicsGrid[{{EzRegCylTES, EzRegCylTESandI},
    {ErhoRegCylTES, ErhoRegCylTESandI}, {EphiRegCylTES, EphiRegCylTESandI},
    {EAllRegCylTES, EAllRegCylTESandI}}, Spacings -> {Scaled[0], Scaled[0]}];
```


Smooth Cylinder Plots - TM + TE Plots

Ez (TM+TE Scattered)

```
In[594]:= EzRegCylTMplusTES = ListLinePlot[dataEzregcylTMplusTES,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolEzregcylTMplusTES]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM + TE Scattered)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ez (TM+TE Inc + Scattered)

```
In[595]:= EzRegCylTMplusTESandI = ListLinePlot[dataEzregcylTMplusTESandI,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolEzregcylTMplusTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Ez Smooth Cylinder (TM + TE Inc and Scattered)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TM+TE Scattered)

```
In[596]:= ErhoRegCylTMplusTES = ListLinePlot[dataErhoregcylTMplusTES,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolErhoregcylTMplusTES]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM + TE Scattered)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Erho (TM+TE Inc + Scattered)

```
In[597]:= ErhoRegCylTMplusTESandI = ListLinePlot[dataErhoregcylTMplusTESandI,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolErhoregcylTMplusTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eρ Smooth Cylinder (TM + TE Inc and Scattered)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Ephi (TM+TE Scattered)

```
In[598]:= EphiRegCylTMplusTES = ListLinePlot[dataEphiregcylTMplusTES,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolEphiregcylTMplusTES]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TM + TE Scattered)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

Ephi (TM+TE Inc + Scattered)

```
In[599]:= EphiRegCylTMplusTESandI = ListLinePlot[dataEphiregcylTMplusTESandI,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolEphiregcylTMplusTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "Eϕ Smooth Cylinder (TM + TE Inc and Scattered)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TM+TE Scattered)

```
In[600]:= EAllRegCylTMplusTES = ListLinePlot[dataEAllTMplusTES,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolEAllTMplusTES]}}, PlotStyle -> Blue,
    BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM + TE Scattered)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

EAll (TM+TE Inc + Scattered)

```
In[601]:= EAllRegCylTMplusTESandI = ListLinePlot[dataEAllTMplusTESandI,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolEAllTMplusTESandI]}},
    PlotStyle -> Blue, BaseStyle -> AbsoluteThickness[2], PlotMarkers -> None,
    PlotLabel -> "ETotal Smooth Cylinder (TM + TE Inc and Scattered)",
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    GridLines -> {{{Pi, {Thick, Gray, Dashed}}}, Automatic}, Background -> White,
    ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```


Summary TM+TE Plots

```
GraphicsGrid[{{EzRegCylTMplusTES, EzRegCylTMplusTESandI},
   {ErhoRegCylTMplusTES, ErhoRegCylTMplusTESandI},
   {EphiRegCylTMplusTES, EphiRegCylTMplusTESandI},
   {EAllRegCylTMplusTES, EAllRegCylTMplusTESandI}},
  Spacings -> {Scaled[0], Scaled[0]}];
```

Compare (Smooth vs Corrugated) Plots

TM Compare

```
In[603]:= GraphicsGrid[{{Show[EzRegCylTMS, EzCorrS,
     PlotRange -> {{0, 2*Pi}, {0, Max[SolEzregcylTMS, SolEzS]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"}, PlotLabel ->
      "Ez TM Scattered Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)"],
   Show[EzRegCylTMSandI, EzCorrSandI, PlotRange ->
      {{0, 2*Pi}, {0, Max[SolEzregcylTMSandI, SolEz]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "Ez TM Inc + Scattered Smooth Cylinder
        (Blue) vs Corrugated Cylinder (Red, --)"]},
  {Show[ErhoRegCylTMS, ErhoCorrS, PlotRange -> {{0, 2*Pi},
       {0, Max[SolErhoregcylTMS, SolErhoS]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, Frame -> True,
     FrameLabel -> {"ρ/λ", "V/m"}, PlotLabel ->
      "Eρ TM Scattered Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)"],
   Show[ErhoRegCylTMSandI, ErhoCorrSandI, PlotRange ->
      {{0, 2*Pi}, {0, Max[SolErhoregcylTMSandI, SolErho]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "Eρ TM Inc + Scattered Smooth Cylinder
        (Blue) vs Corrugated Cylinder (Red, --)"]},
  {Show[EphiRegCylTMS, EphiCorrS, PlotRange -> {{0, 2*Pi},
       {0, Max[SolEphiregcylTMS, SolEphiS]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, Frame -> True,
     FrameLabel -> {"ρ/λ", "V/m"}, PlotLabel ->
      "Eϕ TM Scattered Smooth Cylinder (Blue) vs Corrugated Cylinder (Red, --)"],
   Show[EphiRegCylTMSandI, EphiCorrSandI, PlotRange ->
      {{0, 2*Pi}, {0, Max[SolEphiregcylTMSandI, SolEphi]}},
```

```
   BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
   PlotLabel -> "Eϕ TM Inc + Scattered Smooth Cylinder (Blue) vs
      Corrugated Cylinder (Red, --)"]}, {Show[EAllRegCylTMS,
   EAllCorrS, PlotRange -> {{0, 2*Pi}, {0, Max[SolEAllS, SolEAllTMS]}},
   BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
   PlotLabel -> "ETotal TM Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"],
 Show[EAllRegCylTMSandI, EAllCorrSandI, PlotRange ->
    {{0, 2*Pi}, {0, Max[SolEAllTMSandI, SolEAllSandI]}},
   BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
   Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
   PlotLabel -> "ETotal TM Inc + Scattered Smooth Cylinder
      (Blue) vs Corrugated Cylinder (Red, --)"]}
}]
```

TE Compare

```
In[604]:= GraphicsGrid[{{Show[EzRegCylTES, EzCorrS,
     PlotRange -> {{0, 2*Pi}, {0, Max[SolEzregcylTES, SolEzS]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, Frame -> True,
     FrameLabel -> {"ρ/λ", "V/m"}, PlotLabel -> "Ez TE Scattered Smooth
        Cylinder (Blue) vs Corrugated Cylinder (Red, --)"],
 Show[EzRegCylTESandI, EzCorrSandI, PlotRange -> {{0, 2*Pi},
      {0, Max[SolEzregcylTESandI, SolEz]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "Ez TE Inc + Scattered Smooth Cylinder
        (Blue) vs Corrugated Cylinder (Red, --)"]},
{Show[ErhoRegCylTES, ErhoCorrS, PlotRange -> {{0, 2*Pi},
      {0, Max[SolErhoregcylTES, SolErhoS]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "Eρ TE Scattered Smooth Cylinder (Blue)
        vs Corrugated Cylinder (Red, --)"],
 Show[ErhoRegCylTESandI, ErhoCorrSandI, PlotRange ->
     {{0, 2*Pi}, {0, Max[SolErhoregcylTESandI, SolErho]}},
     BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
     Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
     PlotLabel -> "Eρ TE Inc + Scattered Smooth Cylinder
        (Blue) vs Corrugated Cylinder (Red, --)"]},
{Show[EphiRegCylTES, EphiCorrS, PlotRange -> {{0, 2*Pi},
```



```
      {0, Max[SolEphiregcylTES, SolEphiS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "Eϕ TE Scattered Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)"],
 Show[EphiRegCylTESandI, EphiCorrSandI, PlotRange ->
     {{0, 2*Pi}, {0, Max[SolEphiregcylTESandI, SolEphi]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "Eϕ TE Inc + Scattered Smooth Cylinder (Blue) vs
       Corrugated Cylinder (Red, --)"]}, {Show[EAllRegCylTES,
    EAllCorrS, PlotRange -> {{0, 2*Pi}, {0, Max[SolEAllTES, SolEAllS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "ETotal TE Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"],
 Show[EAllRegCylTESandI, EAllCorrSandI, PlotRange ->
     {{0, 2*Pi}, {0, Max[SolEAllTESandI, SolEAllSandI]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "ETotal TE Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"]}
 }];
```

TM+TE Compare

```
In[605]:= GraphicsGrid[{{Show[EzRegCylTMplusTES, EzCorrS,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolEzregcylTMplusTES, SolEzS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "Ez TM+TE Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"],
 Show[EzRegCylTMplusTESandI, EzCorrSandI, PlotRange ->
     {{0, 2*Pi}, {0, Max[SolEzregcylTMplusTESandI, SolEz]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "Ez TM+TE Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"]},
{Show[ErhoRegCylTMplusTES, ErhoCorrS, PlotRange -> {{0, 2*Pi},
      {0, Max[SolErhoregcylTMplusTES, SolErhoS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "Eρ TM+TE Scattered Smooth Cylinder
```



```
       (Blue) vs Corrugated Cylinder (Red, --)"],
 Show[ErhoRegCylTMplusTESandI, ErhoCorrSandI, PlotRange ->
     {{0, 2*Pi}, {0, Max[SolErhoregcylTMplusTESandI, SolErho]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "Eρ TM+TE Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"]},
{Show[EphiRegCylTMplusTES, EphiCorrS, PlotRange -> {{0, 2*Pi},
      {0, Max[SolEphiregcylTMplusTES, SolEphiS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "Eϕ TM+TE Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"],
 Show[EphiRegCylTMplusTESandI, EphiCorrSandI, PlotRange ->
     {{0, 2*Pi}, {0, Max[SolEphiregcylTMplusTESandI, SolEphi]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "Eϕ TM+TE Inc + Scattered Smooth Cylinder (Blue) vs Corrugated
       Cylinder (Red, --)"]}, {Show[EAllRegCylTES, EAllCorrS,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolEAllS, SolEAllTMplusTES]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "ETotal TM+TE Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"],
 Show[EAllRegCylTMplusTESandI, EAllCorrSandI, PlotRange ->
     {{0, 2*Pi}, {0, Max[SolEAllTMplusTESandI, SolEAllSandI]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ρ/λ", "V/m"},
    PlotLabel -> "ETotal TM+TE Inc + Scattered Smooth
       Cylinder (Blue) vs Corrugated Cylinder (Red, --)"]}
 }];
```

TM, TE, TM+TE Compare

```
In[606]:= RhoXYPlots = GraphicsGrid[{{Show[EzRegCylTMS, EzCorrS,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolEzregcylTMS, SolEzS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, Frame -> True,
    FrameLabel -> {"ϕ", "V/m"}, PlotLabel -> "Ez TM Scattered Smooth
       Cylinder (Blue) vs Corrugated Cylinder (Red, --)"],
   "Ez TE Does Not Exist",
   Show[EzRegCylTMplusTES, EzCorrS,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolEzregcylTMplusTES, SolEzS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, Frame -> True,
```



```
    FrameLabel -> {"ϕ", "V/m"}, PlotLabel -> "Ez TM+TE Scattered Smooth
       Cylinder (Blue) vs Corrugated Cylinder (Red, --)"]},

  {Show[EzRegCylTMSandI, EzCorrSandI,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolEzregcylTMSandI, SolEz]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "Ez TM Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"],
   "Ez TE Does Not Exist",
   Show[EzRegCylTMplusTESandI, EzCorrSandI,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolEzregcylTMplusTESandI, SolEz]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "Ez TM+TE Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"]},

  {Show[ErhoRegCylTMS, ErhoCorrS,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolErhoregcylTMS, SolErhoS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, Frame -> True,
    FrameLabel -> {"ϕ", "V/m"}, PlotLabel -> "Eρ TM Scattered Smooth
       Cylinder (Blue) vs Corrugated Cylinder (Red, --)"],
   Show[ErhoRegCylTES, ErhoCorrS, PlotRange -> {{0, 2*Pi},
      {0, Max[SolErhoregcylTES, SolErhoS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "Eρ TE Scattered Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)"],
   Show[ErhoRegCylTMplusTES, ErhoCorrS, PlotRange ->
     {{0, 2*Pi}, {0, Max[SolErhoregcylTMplusTES, SolErhoS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "Eρ TM+TE Scattered Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)"]},

  {Show[ErhoRegCylTMSandI, ErhoCorrSandI,
    PlotRange -> {{0, 2*Pi}, {0, Max[SolErhoregcylTMSandI, SolErho]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "Eρ TM Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"],
   Show[ErhoRegCylTESandI, ErhoCorrSandI, PlotRange ->
     {{0, 2*Pi}, {0, Max[SolErhoregcylTESandI, SolErho]}},
```

```
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "Eρ TE Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"],
   Show[ErhoRegCylTMplusTESandI, ErhoCorrSandI, PlotRange ->
     {{0, 2 * Pi}, {0, Max[SolErhoregcylTMplusTESandI, SolErho]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "Eρ TM+TE Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"]},

{Show[EphiRegCylTMS, EphiCorrS,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolEphiregcylTMS, SolEphiS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, Frame -> True,
    FrameLabel -> {"ϕ", "V/m"}, PlotLabel -> "Eϕ TM Scattered Smooth
       Cylinder (Blue) vs Corrugated Cylinder (Red, --)"],
   Show[EphiRegCylTES, EphiCorrS, PlotRange -> {{0, 2 * Pi},
      {0, Max[SolEphiregcylTES, SolEphiS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "Eϕ TE Scattered Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)"],
   Show[EphiRegCylTMplusTES, EphiCorrS, PlotRange ->
     {{0, 2 * Pi}, {0, Max[SolEphiregcylTMplusTES, SolEphiS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "Eϕ TM+TE Scattered Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)"]},

{Show[EphiRegCylTMSandI, EphiCorrSandI,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolEphiregcylTMSandI, SolEphi]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "Eϕ TM Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"],
    Show[EphiRegCylTESandI, EphiCorrSandI, PlotRange ->
     {{0, 2 * Pi}, {0, Max[SolEphiregcylTESandI, SolEphi]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "Eϕ TE Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"],
   Show[EphiRegCylTMplusTESandI, EphiCorrSandI, PlotRange ->
     {{0, 2 * Pi}, {0, Max[SolEphiregcylTMplusTESandI, SolEphi]}},
```



```
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "Eϕ TM+TE Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"]},

  {Show[EAllRegCylTMS, EAllCorrS,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolEAllS, SolEAllTMS]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}, Frame -> True,
    FrameLabel -> {"ϕ", "V/m"}, PlotLabel -> "ETotal TM Scattered Smooth
       Cylinder (Blue) vs Corrugated Cylinder (Red, --)"],
   Show[EAllRegCylTES, EAllCorrS, PlotRange -> {{0, 2 * Pi},
      {0, Max[SolEAllS, SolEAllTES]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "ETotal TE Scattered Smooth Cylinder (Blue)
       vs Corrugated Cylinder (Red, --)"],
   Show[EAllRegCylTMplusTES, EAllCorrS, PlotRange ->
     {{0, 2 * Pi}, {0, Max[SolEAllS, SolEAllTMplusTES]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "ETotal TM+TE Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"]},

{Show[EAllRegCylTMSandI, EAllCorrSandI,
    PlotRange -> {{0, 2 * Pi}, {0, Max[SolEAllTMSandI, SolEAllSandI]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "ETotal TM Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"],
   Show[EAllRegCylTESandI, EAllCorrSandI, PlotRange ->
     {{0, 2 * Pi}, {0, Max[SolEAllTESandI, SolEAllSandI]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "ETotal TE Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"],
   Show[EAllRegCylTMplusTESandI, EAllCorrSandI, PlotRange ->
     {{0, 2 * Pi}, {0, Max[SolEAllTMplusTESandI, SolEAllSandI]}},
    BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"},
    Frame -> True, FrameLabel -> {"ϕ", "V/m"},
    PlotLabel -> "ETotal TM+TE Inc + Scattered Smooth Cylinder
       (Blue) vs Corrugated Cylinder (Red, --)"]}
 }]
```


Changing Phi RCS dB XY Plots - Compare with Reference [Cross-Polar]

Corrugated Cylinder Plots

Plot Cross-Polar [RCSσϕθ According to A. Ferni]

```
(*Cross-Polar according to A. Freni, 1996,
Scattering from a dielectric cylinder axially loaded
with periodic metallic rings*)
RCSErhoCorrdBxy = ListLinePlot[RCSσϕθdB, PlotRange ->
     {{0, Pi}, {Min[10 * Log10[RCSσϕθ / lambda0]], Max[10 * Log10[RCSσϕθ / lambda0]]}},
   PlotStyle -> {Red, Dashed}, BaseStyle -> AbsoluteThickness[2],
   PlotMarkers -> None, PlotLabel -> "Cross-Polar Corrugated Cylinder",
   Frame -> True, FrameLabel -> {"ϕ", "σφθ/λ0 (dB)"},
   GridLines -> {{{Pi / 2, {Thick, Gray, Dashed}}}, Automatic},
   Background -> White, ImageSize -> 700,
   BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];

Show[{RCSErhoCorrdBxy, CrossPolarFEMdBXYPlot, CrossPolarMoMdBXYPlot}, PlotRange ->
   {{0, Pi}, {Min[10 * Log10[RCSσϕθ / lambda0], CrossPolarFEMdB, CrossPolarMoMdB],
     Max[10 * Log10[RCSσϕθ / lambda0], CrossPolarFEMdB, CrossPolarMoMdB]}},
  PlotLabel -> "Cross-Polar Corr. Cyl. σφθ/λ0 (dB) Paper
     (Red, --) vs FEM (Green, ||) vs MoM (Blue)",
  ImageSize -> 700, BaseStyle -> {FontSize -> 14, FontWeight -> "Bold"}];
```

APPENDIX B

Derivation for Fundamental Equations of Guided Waves from Maxwell's Equations

Start with Maxwell's equations in differential form, $\nabla\times\vec{E}$ and $\nabla\times\vec{H}$ [12, p. 2], and expand into cylindrical coordinates [12, p. 925]

$$\nabla\times\vec{E} = -j\omega\mu\vec{H} = \hat{\rho}\left(\frac{1}{\rho}\frac{\partial E_z}{\partial\phi} - \frac{\partial E_\phi}{\partial z}\right) + \hat{\phi}\left(\frac{\partial E_\rho}{\partial z} - \frac{\partial E_z}{\partial\rho}\right) + \hat{z}\left(\frac{1}{\rho}\frac{\partial(\rho E_\phi)}{\partial\rho} - \frac{1}{\rho}\frac{\partial E_\rho}{\partial\phi}\right)$$

$$\nabla\times\vec{H} = j\omega\varepsilon\vec{E} = \hat{\rho}\left(\frac{1}{\rho}\frac{\partial H_z}{\partial\phi} - \frac{\partial H_\phi}{\partial z}\right) + \hat{\phi}\left(\frac{\partial H_\rho}{\partial z} - \frac{\partial H_z}{\partial\rho}\right) + \hat{z}\left(\frac{1}{\rho}\frac{\partial(\rho H_\phi)}{\partial\rho} - \frac{1}{\rho}\frac{\partial H_\rho}{\partial\phi}\right)$$

Replace, $\frac{\partial}{\partial z} = -j\beta$ based on the relationship the relationship $\frac{\partial}{\partial z}\left(e^{-j\beta z}\right) = -j\beta e^{-j\beta z}$

and separate into cylindrical components

$$-j\omega\mu H_\rho = \frac{1}{\rho}\frac{\partial E_z}{\partial\phi} - \frac{\partial E_\phi}{\partial z} = \frac{1}{\rho}\frac{\partial E_z}{\partial\phi} + j\beta E_\phi$$

$$j\omega\varepsilon E_\rho = \frac{1}{\rho}\frac{\partial H_z}{\partial\phi} - \frac{\partial H_\phi}{\partial z} = \frac{1}{\rho}\frac{\partial H_z}{\partial\phi} + j\beta H_\phi$$

$$-j\omega\mu H_\phi = \frac{\partial E_\rho}{\partial z} - \frac{\partial E_z}{\partial\rho} = -j\beta E_\rho - \frac{\partial E_z}{\partial\rho}$$

$$j\omega\varepsilon E_\phi = \frac{\partial H_\rho}{\partial z} - \frac{\partial H_z}{\partial\rho} = -j\beta H_\rho - \frac{\partial H_z}{\partial\rho}$$

$$-j\omega\mu H_z = \frac{1}{\rho}\frac{\partial(\rho E_\phi)}{\partial\rho} - \frac{1}{\rho}\frac{\partial E_\rho}{\partial\phi}$$

$$j\omega\varepsilon E_z = \frac{1}{\rho}\frac{\partial(\rho H_\phi)}{\partial\rho} - \frac{1}{\rho}\frac{\partial H_\rho}{\partial\phi}$$

Also,

$$k = \frac{2\pi}{\lambda} = \omega\sqrt{\mu\varepsilon}$$

$$k_c{}^2 = k^2 - \beta^2 = \omega^2\mu\varepsilon - \beta^2$$

Solving for E_ρ

$$E_\rho = \frac{1}{j\omega\varepsilon}\left(\frac{1}{\rho}\frac{\partial H_z}{\partial \phi} + j\beta H_\phi\right)$$

Substitute

$$H_\phi = \frac{1}{j\omega\mu}\left(j\beta E_\rho + \frac{\partial E_z}{\partial \rho}\right)$$

$$E_\rho = \frac{1}{j\omega\varepsilon}\left(\frac{1}{\rho}\frac{\partial H_z}{\partial \phi} + \frac{j\beta}{j\omega\mu}\left(j\beta E_\rho + \frac{\partial E_z}{\partial \rho}\right)\right) = \frac{1}{j\omega\varepsilon}\left(\frac{1}{\rho}\frac{\partial H_z}{\partial \phi} - \frac{\beta^2 E_\rho}{j\omega\mu} + \frac{j\beta}{j\omega\mu}\frac{\partial E_z}{\partial \rho}\right)$$

$$E_\rho = \frac{1}{j\omega\varepsilon\rho}\frac{\partial H_z}{\partial \phi} + \frac{\beta^2 E_\rho}{\omega^2\mu\varepsilon} - \frac{j\beta}{\omega^2\mu\varepsilon}\frac{\partial E_z}{\partial \rho}$$

$$E_\rho - \frac{\beta^2 E_\rho}{\omega^2\mu\varepsilon} = \frac{1}{j\omega\varepsilon\rho}\frac{\partial H_z}{\partial \phi} - \frac{j\beta}{\omega^2\mu\varepsilon}\frac{\partial E_z}{\partial \rho}$$

$$E_\rho\left(1 - \frac{\beta^2}{\omega^2\mu\varepsilon}\right) = \frac{1}{j\omega\varepsilon\rho}\frac{\partial H_z}{\partial \phi} - \frac{j\beta}{\omega^2\mu\varepsilon}\frac{\partial E_z}{\partial \rho}$$

Multiply both sides by $\omega^2\mu\varepsilon$

$$E_\rho(\omega^2\mu\varepsilon - \beta^2) = \frac{\omega\mu}{j\rho}\frac{\partial H_z}{\partial \phi} - j\beta\frac{\partial E_z}{\partial \rho}$$

Replace ${k_c}^2 = \omega^2\mu\varepsilon - \beta^2$and solve for E_ρ

$$\boxed{E_\rho = \frac{-j}{{k_c}^2}\left(\frac{\omega\mu}{\rho}\frac{\partial H_z}{\partial \phi} + \beta\frac{\partial E_z}{\partial \rho}\right)}$$

Solving for H_ϕ

$$H_\phi = \frac{1}{j\omega\mu}\left(j\beta E_\rho + \frac{\partial E_z}{\partial \rho}\right) = \frac{1}{j\omega\mu}\left(j\beta\left(\frac{1}{j\omega\varepsilon}\left(\frac{1}{\rho}\frac{\partial H_z}{\partial \phi} + j\beta H_\phi\right)\right) + \frac{\partial E_z}{\partial \rho}\right)$$

$$H_\phi = \frac{1}{j\omega\mu}\left(\frac{j\beta}{j\omega\varepsilon\rho}\frac{\partial H_z}{\partial \phi} - \frac{\beta^2}{j\omega\varepsilon}H_\phi + \frac{\partial E_z}{\partial \rho}\right) = -\frac{j\beta}{\omega^2\mu\varepsilon\rho}\frac{\partial H_z}{\partial \phi} + \frac{\beta^2}{\omega^2\mu\varepsilon}H_\phi + \frac{1}{j\omega\mu}\frac{\partial E_z}{\partial \rho}$$

$$H_\phi\left(1 - \frac{\beta^2}{\omega^2\mu\varepsilon}\right) = -\frac{j\beta}{\omega^2\mu\varepsilon\rho}\frac{\partial H_z}{\partial \phi} + \frac{1}{j\omega\mu}\frac{\partial E_z}{\partial \rho}$$

Multiply both sides by $\omega^2\mu\varepsilon$

$$H_\phi(\omega^2\mu\varepsilon - \beta^2) = -\frac{j\beta}{\rho}\frac{\delta H_z}{\delta \phi} + \frac{\omega\varepsilon}{j}\frac{\partial E_z}{\partial \rho} = -j\left(\frac{\beta}{\rho}\frac{\partial H_z}{\partial \phi} + \omega\varepsilon\frac{\partial E_z}{\partial \rho}\right)$$

$$H_\phi = \frac{-j}{{k_c}^2}\left(\frac{\beta}{\rho}\frac{\partial H_z}{\partial \phi} + \omega\varepsilon\frac{\partial E_z}{\partial \rho}\right)$$

Solving for E_ϕ

$$E_\phi = \frac{1}{j\omega\varepsilon}\left(-j\beta H_\rho - \frac{\partial H_z}{\partial \rho}\right)$$

Substitute,

$$H_\rho = -\frac{1}{j\omega\mu}\left(\frac{1}{\rho}\frac{\partial E_z}{\partial \phi} + j\beta E_\phi\right)$$

$$E_\phi = \frac{1}{j\omega\varepsilon}\left(\frac{j\beta}{j\omega\mu\rho}\frac{\partial E_z}{\partial \phi} - \frac{\beta^2}{j\omega\mu}E_\phi - \frac{\partial H_z}{\partial \rho}\right) = \frac{-j\beta}{\omega^2\mu\varepsilon\rho}\frac{\partial E_z}{\partial \phi} + \frac{\beta^2}{\omega^2\mu\varepsilon}E_\phi - \frac{1}{j\omega\varepsilon}\frac{\partial H_z}{\partial \rho}$$

$$E_\phi\left(1 - \frac{\beta^2}{\omega^2\mu\varepsilon}\right) = \frac{-j\beta}{\omega^2\mu\varepsilon\rho}\frac{\partial E_z}{\partial \phi} - \frac{1}{j\omega\varepsilon}\frac{\partial H_z}{\partial \rho}$$

Multiply both sides by $\omega^2\mu\varepsilon$

$$E_\phi(\omega^2\mu\varepsilon - \beta^2) = \frac{-j\beta}{\rho}\frac{\partial E_z}{\partial \phi} - \frac{\omega\mu}{j}\frac{\partial H_z}{\partial \rho} = -j\left(\frac{\beta}{\rho}\frac{\partial E_z}{\partial \phi} - \omega\mu\frac{\partial H_z}{\partial \rho}\right)$$

$$E_\phi = \frac{-j}{{k_c}^2}\left(\frac{\beta}{\rho}\frac{\delta E_z}{\delta \phi} - \omega\mu\frac{\delta H_z}{\delta \rho}\right)$$

Solving for H_ρ

$$H_\rho = \frac{-1}{j\omega\mu}\left(\frac{1}{\rho}\frac{\partial E_z}{\partial \phi} + j\beta E_\phi\right)$$

Substitute,

$$E_\phi = \frac{1}{j\omega\varepsilon}\left(j\beta H_\rho - \frac{\partial H_z}{\partial \rho}\right)$$

$$H_\rho = \frac{-1}{j\omega\mu}\left(\frac{1}{\rho}\frac{\partial E_z}{\partial \phi} + j\beta\left(\frac{1}{j\omega\varepsilon}\left(j\beta H_\rho - \frac{\partial H_z}{\partial \rho}\right)\right)\right) = \frac{-1}{j\omega\mu}\left(\frac{1}{\rho}\frac{\partial E_z}{\partial \phi} + \frac{\beta^2}{j\omega\varepsilon}H_\rho - \frac{j\beta}{j\omega\varepsilon}\frac{\partial H_z}{\partial \rho}\right)$$

$$H_\rho = \frac{-1}{j\omega\mu\rho}\frac{\partial E_z}{\partial \phi} + \frac{\beta^2}{\omega^2\mu\varepsilon}H_\rho - \frac{j\beta}{\omega^2\mu\varepsilon}\frac{\partial H_z}{\partial \rho}$$

$$H_\rho\left(1 - \frac{\beta^2}{\omega^2\mu\varepsilon}\right) = \frac{-1}{j\omega\mu\rho}\frac{\partial E_z}{\partial \phi} - \frac{j\beta}{\omega^2\mu\varepsilon}\frac{\partial H_z}{\partial \rho}$$

Multiply both sides by $\omega^2\mu\varepsilon$

$$H_\rho(\omega^2\mu\varepsilon - \beta^2) = \frac{-\omega\varepsilon}{j\rho}\frac{\partial E_z}{\partial \phi} - j\beta\frac{\partial H_z}{\partial \rho} = -j\left(\frac{-\omega\varepsilon}{\rho}\frac{\partial E_z}{\partial \phi} + \beta\frac{\partial H_z}{\partial \rho}\right)$$

$$\boxed{H_\rho = \frac{-j}{{k_c}^2}\left(\frac{-\omega\varepsilon}{\rho}\frac{\partial E_z}{\partial \phi} + \beta\frac{\partial H_z}{\partial \rho}\right)}$$

Note that in paper k_c is substituted with k_ρ and β is substituted with k_z.

REFERENCES

[1] G. Manara, G. Pelosi, A. Monorchi and R. Coccioli, "Plane-wave scattering from cylinders with transverse corrugations," *Electronics Letters,* vol. 316, pp. 437-438, 16 March 1995.

[2] G. Manara, A. Monorchio, G. Pelosi and R. Coccioli, "Electromagnetic Scattering From Corrugated Cylinders," in *Antennas and Propagation Society International Symposium*, 1995.

[3] A. A. Kishk and P.-S. Kildal, "An asymptotic boundary condition for corrugated surfaces and its application to calculate scattering from circular cylinders with dielectric filled corrugations," in *Antennas and Propagation Society International Symposium*, 1996.

[4] A. A. Kishk, K. S. P., A. Monorchi and G. Manara, "Asymptotic boundary condition for corrugated surfaces, and its application to scattering from circular cylinders with dielectric filled corrugations," *IEE Proceedings - Microwaves, Antennas and Propagation,* vol. 145, no. 1, pp. 116-122, 1998.

[5] A. Freni, "Plane wave scattering from a cylinder loaded periodically with groups of metallic rings," *Electronics Letters,* vol. 32, no. 10, pp. 874-875, 9 May 1996.

[6] T. C. Rao, "Plane Wave Scattering by a Corrugated Conducting Cylinder at Oblique Incidence," *IEEE Transactions on Antennas and Propagation,* vol. 36, no. 8, pp. 1184-1188, 1988.

[7] P. Hillion, "Scattering of Te and Tm," *Journal of Electromagnetic Waves and Applications,* vol. 14, no. 5, pp. 655-662, 2000.

[8] R. E. Collins, Field Theory of Guided Waves, Second ed., New York: John Wiley & Sons, Inc., 1991.

[9] A. F. Peterson, S. L. Ray and R. Mittra, Computational Methods for Electromagnetics, New York: IEEE Press, 1998.

[10] J. F. G. Farrell and D. H. Kuhn, "Mutual coupling in infinite planar arrays of rectangular waveguide horns," *IEEE Transactions on Antennas and Propogation,* Vols. AP-16, pp. 405-414, 1968.

[11] T. Bohning, *Electrom agnetic scattering from a periodic array of open-ended,* PhD Dissertation, Florida Atlantic University, 1992.

[12] C. A. Balanis, Advanced Engineering Electromagnetics, Hoboken: John Wiley & Sons, 1989.

[13] D. M. Pozar, Microwave Engineering, 3rd ed., Hoboken, NJ: John Wiley & Sons, 2005.

[14] R. F. Harrington, Time-Harmonic Electromagnetic Fields, New York: IEEE Press, 2001.

[15] Wikipedia, "Bloch wave," Wikipedia, 23 September 2016. [Online]. Available: https://en.wikipedia.org/wiki/Bloch_wave. [Accessed 5 November 2016].

[16] F. T. Ulaby, Fundamentals of Applied Electromagnetics, 5th ed., Upper Saddle River: Pearson Prentice Hall, 2007.

[17] Wolfram Research, Inc., *Mathematica,* Version 10.3 ed., Champaign, Illinois: Wolfram Research, Inc., 2016.

[18] J. W. Helton and R. L. Miller, "NCAlgebra," February 2016. [Online]. Available: http://math.ucsd.edu/~ncalg/.

[19] M. T. Heath, "The Numerical Solution of Ill-Conditioned Systems of Linear Equations," National Technical Information Service U.S. Department of Commerce, Springfield, 1974.

[20] E. W. Weisstein, "Moore-Penrose Matrix Inverse," From MathWorld--A Wolfram Web Resource, [Online]. Available: http://mathworld.wolfram.com/Moore-PenroseMatrixInverse.html. [Accessed 4 November 2016].

[21] Wolfram Research, "LeastSquares," Wolfram Research, 2007. [Online]. Available: https://reference.wolfram.com/language/ref/LeastSquares.html. [Accessed 4 November 2016].

[22] C. Wolff, "Radar Basics," Christian Wolff, [Online]. Available: http://www.radartutorial.eu/01.basics/Rayleigh-%20versus%20Mie-Scattering.en.html. [Accessed 4 November 2016].

[23] A. E. Fuhs, "Radar Cross Section Lectures," Naval Postgraduate School, Monterey, 1982.

[24] Catslash, "File:Radar cross section of metal sphere from Mie theory.svg," Wikipedia, 27 June 2009. [Online]. Available:

https://commons.wikimedia.org/wiki/File:Radar_cross_section_of_metal_sphere_from_Mie_theory.svg. [Accessed 4 November 2016].

[25] A. A. Kishk, P. -S. Kildal, A. Monorchio, G. Manara and Z. Sipus, "Validation of the Asymptotic Cormgation Boundary Condition for Circular Cylinders with Dielectric Filled Corrugations," *IEEE Conference Publications,* vol. 3, pp. 2080-2083, 1997.

[26] K. A. Ahmed and P.-S. Kildal, "Asymptotic Boundary Conditions for Strip-Loaded Scatterers Applied to Circular Dielectric Cylinders Under Oblique Incidence," *IEEE Transactions on Antennas and Propagation,* vol. 45, no. 1, pp. 51-55, 1997.

[27] A. Freni, C. Mias and F. L. Ronald, "Hybrid Finite-Element Analysis of Electromagnetic Plane Wave Scattering from Axially Periodic Cylindrical Structures," *IEEE Transactions on Antennas and Propagation,* vol. 46, no. 12, pp. 1859 - 1866, 1998.

[28] A. Freni, "Scattering from a dielectric cylinder axially loaded with periodic metallic rings," *IEE Proc.-Microw. Antennas Propug.,,* vol. 143, no. 3, pp. 233-237, 1996.

[29] C. A. Balanis, Antenna Theory: Analysis and Design, 3rd ed., Hoboken: John Wiley & Sons, 2005.

[30] A. Rohatgi, "http://arohatgi.info/WebPlotDigitizer/," May 2016. [Online]. Available: http://arohatgi.info/WebPlotDigitizer. [Accessed 12 November 2016].

[31] S. Garcia, J. Bagby and I. Morazzani, "Plane-Wave Scattering of a Periodic Corrugated Cylinder," in *Proc. of the International Microwave Symposium (IMS)*, Honolulu, 2017.

[32] M. T. Wlodarczyk and S. R. Seshadri, "Excitation and scattering of guided modes on a dielectric cylinder with a periodically varying radius," *Journal of Applied Physics,* vol. 57, no. 3, pp. 943-955, 1984.

[33] G. Zheng, X. Yin, J. Wang, M. Guo and X. Wang, "“Complex permittivity and microwave absorbing property," *J. Mater. Sci. Technol.,* vol. 28(8), p. 745–50, 2012.

[34] J. Bagby and S. Garcia, "Electromagnetic Scattering from a Loaded Corrugated Cylinder," in *Proc. of the Progress In Electromagnetics Research Symposium (PIERS)*, St Petersburg, 2017.

[35] B. A. Munk, Frequency Selective Surfaces: Theory and Design, New York: John Wiley & Sons, 2000.