Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Parabolic Reflector Modelling Techniques with a Laterally Displaced Feed

Frédéric Arpin

Defence R&D Canada – Ottawa
TECHNICAL MEMORANDUM
DRDC Ottawa TM 2005-098
July 2005

Canada

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUL 2005		**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Parabolic Reflector Modelling Techniques with a Laterally Displaced Feed	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Defence R&D Canada -Ottawa,National Defence Headquarters,Ottawa,CA,K1A OK2**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

This document looks at two techniques for modelling the radiation pattern of a parabolic reflector antenna. Both these methods use physical optics to determine the induced currents on the reflector due to the feed. The first method uses MATLAB's numerical integration routine to compute the far-field radiation pattern. The second method involves segmenting the reflector surface and approximating the amplitude and phase of the surface currents by a first degree polynomial. This approximation allows the integration to be performed analytically in closed-form. Three different reflector configurations are investigated including two configurations with a laterally displaced feed. Both these techniques yield accurate results but the latter technique had a significant improvement in computational efficiency.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**66**	

Parabolic Reflector Modelling Techniques with a Laterally Displaced Feed

Frédéric Arpin

Defence R&D Canada – Ottawa

Technical Memorandum

DRDC Ottawa TM 2005-098

July 2005

Abstract

This document looks at two techniques for modelling the radiation pattern of a parabolic reflector antenna. Both these methods use physical optics to determine the induced currents on the reflector due to the feed. The first method uses MATLAB's numerical integration routine to compute the far-field radiation pattern. The second method involves segmenting the reflector surface and approximating the amplitude and phase of the surface currents by a first degree polynomial. This approximation allows the integration to be performed analytically in closed-form. Three different reflector configurations are investigated including two configurations with a laterally displaced feed. Both these techniques yield accurate results but the latter technique had a significant improvement in computational efficiency.

Résumé

Le présent document traite de deux méthodes de modélisation du diagramme de rayonnement d'une antenne à réflecteur parabolique. Les deux méthodes font appel à l'optique physique pour déterminer les courants induits sur le réflecteur par l'alimentation. La première méthode utilise une routine d'intégration numérique MATLAB pour calculer le diagramme de rayonnement en champ lointain. La deuxième effectue une segmentation de la surface du réflecteur et une approximation de l'amplitude et de la phase des courants de surface au moyen d'un polynôme du premier degré. Cette approximation permet d'effectuer une intégration analytique en forme fermée. Trois configurations de réflecteur différentes sont étudiées, y compris deux configurations avec une alimentation déplacée latéralement. Ces deux méthodes donnent des résultats précis, mais la deuxième présente une efficacité computationnelle nettement supérieure.

This page intentionally left blank.

Executive summary

DRDC Ottawa is investigating the design of missile models that can be used to predict missile performance in the presence and absence of Electronic Countermeasures (ECM). This is the first of a series of publications that will deal with the different components that make up the seeker and guidance system of a radar guided anti-ship missile. This publication examines different methods of modelling the missile seeker antenna accurately and efficiently. Once an appropriate antenna modelling method has been identified, it will be integrated into a larger simulation framework that attempts to capture the operation of the complete missile in its operating environment. While antenna modelling is computationally intense it is important that the chosen model not significantly slow down the missile engagement simulation. Determining the best ECM technique requires a large number of simulation runs, therefore it is important that these individual runs execute in seconds versus hours.

This document looks at two techniques for modelling the radiation pattern of a parabolic reflector antenna. Both these methods use physical optics to determine the induced currents on the reflector due to the feed. The first method uses MATLAB's numerical integration routine to compute the far-field radiation pattern. The second method involves segmenting the reflector surface and approximating the amplitude and phase of the surface currents by a first degree polynomial. This approximation allows the integration to be performed analytically in closed- form. Three different reflector configurations are investigated including two configurations with a laterally displaced feed. Both of the methods discussed (Physical Optics/Surface Current (PO/SC) Method and Ludwig's Method) give accurate results when compared to patterns computed using commercially available reflector modelling software. Where these two methods differ is in the amount of time required to compute these results. The PO/SC Method, with a lateral displacement of the feed, where the integrand of the radiation integral increases in complexity, requires in excess of one hour to evaluate one principal plane pattern. Incorporating this model into a seeker model would therefore be impractical. As an alternative, Ludwig's Method, regardless of the lateral displacement of the feed, can accurately compute the principal plane pattern in less than 10 seconds. Because of its speed and accuracy, future work will involve applying Ludwig's method to monopulse reflector antennas having a compound feed. Finally this more efficient monopulse antenna model could then be incorporated into a missile seeker model.

A number of reports will be written over the next year that will provide the foundation for future countermeasures research. These reports are the result of studies carried out as part of Project 11at and will be used to develop future Thrust 11a Projects that will be required to provide support for the development of EW techniques in the future. By providing a deeper understanding of the interactions between the jammer and threats, these studies will also help to define the requirements for future Naval EW systems.

Frédéric Arpin. 2005. Parabolic Reflector Modelling Techniques with a Laterally Displaced Feed. DRDC Ottawa TM 2005-098. Defence R&D Canada – Ottawa.

Sommaire

RDDC Ottawa mène des recherches sur la conception de modèles de missile susceptibles de servir à la prédiction du rendement des missiles en présence et en l'absence de contre-mesures électroniques (CME). Le présent document est le premier d'une série de publications sur les différents composants de l'autodirecteur et du système de guidage d'un missile antinavires guidé au radar. La présente publication examine différentes méthodes de modélisation de l'antenne de l'autodirecteur du missile avec précision et de façon efficiente. Une fois qu'une méthode appropriée de modélisation de l'antenne aura été identifiée, elle sera intégrée à un cadre de simulation plus vaste qui saisit autant que possible le fonctionnement du missile complet dans son milieu opérationnel. Bien que la modélisation de l'antenne soit intense sur le plan des calculs, il est important que le modèle choisi ne ralentisse pas significativement la simulation de l'engagement du missile. Pour déterminer la meilleure technique CME, il faut un grand nombre de passages en simulation ; c'est pourquoi il est important que les passages individuels soient exécutés en quelques secondes plutôt qu'en quelques heures.

Le présent document traite de deux méthodes de modélisation du diagramme de rayonnement d'une antenne à réflecteur parabolique. Les deux méthodes font appel à l'optique physique pour déterminer les courants induits sur le réflecteur par l'alimentation. La première méthode utilise une routine d'intégration numérique MATLAB pour calculer le diagramme de rayonnement en champ lointain. La deuxième effectue une segmentation de la surface du réflecteur et une approximation de l'amplitude et de la phase des courants de surface au moyen d'un polynôme de premier degré. Cette approximation permet d'éffectuer une intégration analytique en forme fermée. Trois configurations de réflecteur différentes sont étudiées, y compris deux configurations avec une alimentation déplacée latéralement. Les deux méthodes examinées (méthode de détermination du courant de surface à l'aide de l'optique physique et méthode de Ludwig) donnent des résultats précis par rapport aux diagrammes calculés au moyen de logiciels commerciaux de modélisation du réflecteur. Là où ces deux méthodes diffèrent, c'est dans le temps requis pour le calcul des résultats. La méthode de détermination du courant de surface à l'aide de l'optique physique, qui tient compte du déplacement latéral de l'alimentation, où la fonction à intégrer de l'intégrale de rayonnement augmente en complexité, requiert plus d'une heure pour l'évaluation d'un diagramme sur le plan principal. Il ne serait donc pas pratique d'intégrer cette méthode à un modèle d'autodirecteur. Par contre, la méthode de Ludwig, peu importe le déplacement latéral de l'alimentation, peut calculer avec précision le diagramme sur le plan principal en moins de 10 secondes. Grâce à la rapidité et à la précision de cette méthode, les recherches futures feront appel à l'application de techniques de modélisation, fondées sur la méthode de Ludwig, aux antennes à réflecteur ayant une alimentation composée pour utilisation dans des systèmes monopulses. Enfin, un modèle d'antenne monopulse pourrait alors être intégré à un modèle d'autodirecteur de missile.

Au cours de l'année qui vient, un certain nombre de rapports seront rédigés pour mettre en place les fondements des recherches futures sur les contre-mesures. Ces rapports seront les résultats d'études menées dans le cadre du projet 11at et serviront à la définition des projets

futurs du volet 11a qui seront requis à l’appui de la mise au point de techniques de guerre électronique (GE) à l’avenir. En permettant une meilleure compréhension des interactions entre le brouilleur et les menaces, ces études contribueront aussi à la définition des besoins en ce qui concerne les futurs systèmes GE navals.

Frédéric Arpin. 2005. Parabolic Reflector Modelling Techniques with a Laterally Displaced Feed . DRDC Ottawa TM 2005-098. R & D pour la défense Canada – Ottawa.

This page intentionally left blank.

Table of contents

List of figures

This page intentionally left blank.

1 Introduction

This document will investigate two methods for the modelling of the radiation pattern of a parabolic reflector with a laterally displaced feed. Both techniques are based on physical optics and both techniques involve determining the surface currents induced on the reflector surface.

The first technique will integrate the induced surface currents over the reflector surface which will yield the far-field radiation pattern. This integration will be performed using MATLAB's built-in numerical integration routine that uses recursive adaptive Simpson quadrature to approximate the integral.

The second method, instead of integrating the surface currents directly, imposes a integration grid on the surface of the reflector, where within each grid element the amplitude and phase of the integrand is approximated by a simple linear function. This function over the incremental surface area, which make up a grid element, can be integrated analytically.

The goal here is to determine a computationally efficient modelling technique that can be used to model monopulse tracking antennas which can then be incorporated into a missile seeker model.

2 Modelling of Far-Field Radiation Patterns for Reflector Antennas

2.1 PO/Surface Current Method (PO/SC)

Consider the problem geometry shown in Figure 1. To determine the radiation pattern of this reflector we make use of the far-field radiation integral given as [1]

$$\bar{E}^{rad}(\theta_p,\phi_p) = -jkZ_o\frac{e^{-jk|\bar{r}_p|}}{|\bar{r}_p|}(\bar{\bar{I}}-\hat{r}_p\hat{r}_p)\int\int_S \bar{J}_s e^{jk\bar{r}_s\cdot\hat{r}_p}ds \tag{1}$$

Where $\bar{r}_p$, $\bar{r}_s$ are defined in Figure 1, k is the free space propagation constant, Z_o is the intrinsic impedance of free space, $\bar{\bar{I}}$ is a unit dyadic and $\bar{J}_s$ is the induced surface current on S.

In order to evaluate (1) we must first determine the surface current $\bar{J}_s$, induced onto the reflector surface, by the means of the physical optics (PO) approximation. The PO approximation makes three assumptions, namely:

- The reflector has an electrically large radius of curvature so that locally at each reflection point the surface of the reflector can be considered planar.
- The radius of curvature of the incident field is large and locally at each reflection point can be viewed as a plane wave.

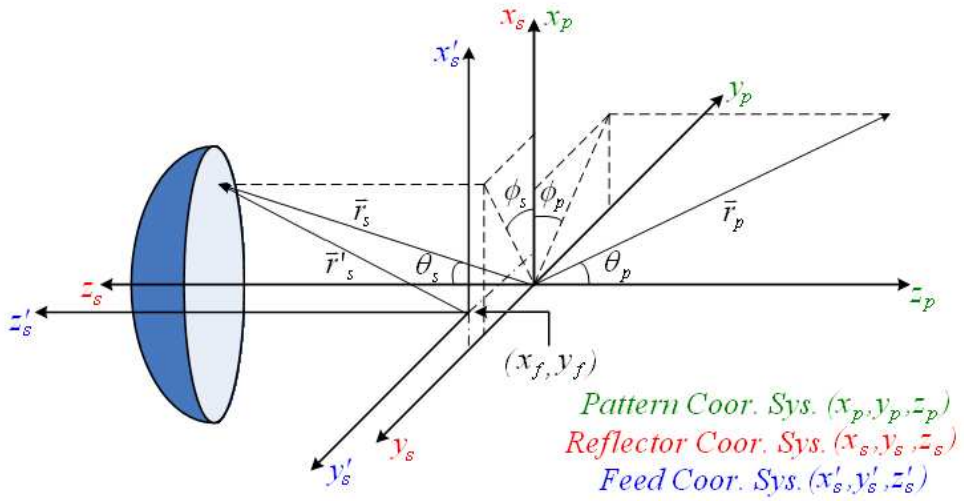

Figure 1: *Parabolic Reflector Problem Geometry*

- The reflector is considered to be a perfectly conducting surface which yields a reflected field equal to the incident field.

If we then have a locally plane wave incident on a locally plane perfectly conducting surface, referring to Figure 2, the boundary conditions tell us that the current induced on the

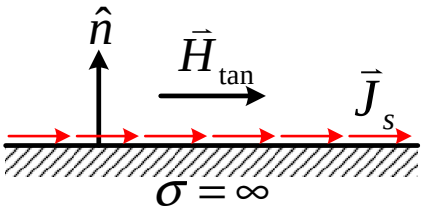

Figure 2: *Induced Currents on a Perfect Conductor*

surface of the conductor will be given by

$$\bar{J_s} = \bar{H}_{tan} = \hat{n} \times \bar{H} = \hat{n} \times (\bar{H_i} + \bar{H_r}) \tag{2}$$

Now considering the third assumption above we have that the reflected field is equal to the incident field, we can then write the induced surface current as

$$\bar{J_s} = \hat{n} \times (\bar{H_i} + \bar{H_i}) = 2\hat{n} \times \bar{H_i} \tag{3}$$

We can now proceed to determining the equations defining $\bar{J_s}$. We will start by defining the electric field from the feed antenna, for the purpose of this report we will model the feed in terms of its principal plane patterns. For a y-polarized feed we have

$$\bar{E}^{inc}(\bar{r'}_s) = \left[(C_E(\theta'_s)sin\phi'_s)\hat{\theta}'_s + (C_H(\theta'_s)cos\phi'_s)\hat{\phi}'_s\right] \frac{e^{-jkr'_s}}{r'_s} \tag{4}$$

where $C_E(\theta'_s)$ and $C_H(\theta'_s)$ define the E-Plane and H-Plane principal patterns respectively, which are defined as

$$C_E(\theta'_s) = (cos\theta'_s)^{qE}$$
$$C_H(\theta'_s) = (cos\theta'_s)^{qH}$$

qE and qH are selected such as to match the $C_E(\theta'_s)$ and $C_H(\theta'_s)$ to the principal plane patterns of the actual feed. The magnetic incident field can be related to the incident electric field by the following equation

$$\bar{H}^{inc}(\bar{r'}_s) = \frac{1}{Z_o}\bar{r'}_s \times \bar{E}^{inc}(\bar{r'}_s) \tag{5}$$

We can then determine the expression for the incident magnetic field given (4) as follows

$$\hat{r'}_s \times \bar{E}^{inc}(\bar{r'}_s) = \begin{vmatrix} \hat{r'}_s & \hat{\theta'}_s & \hat{\phi'}_s \\ 1 & 0 & 0 \\ 0 & E_{\theta'_s} & E_{\phi'_s} \end{vmatrix}$$

$$\bar{H}^{inc}(\hat{r'}_s) = \frac{1}{Z_o}\bar{r'}_s \times \bar{E}^{inc}(\bar{r'}_s) = \left[(-C_H(\theta'_s)cos\phi'_s)\hat{\theta'}_s + (C_E(\theta'_s)sin\phi'_s)\hat{\phi'}_s\right]\frac{e^{-jkr'_s}}{Z_o r'_s} \tag{6}$$

This can now be expressed in cartesian components as

$$\begin{aligned} \bar{H}^{inc}(\bar{r'}_s) &= \left[(-C_H cos\theta'_s cos^2\phi'_s - C_E sin^2\phi'_s)\hat{x}'_s\right. \\ &+ (-C_H cos\theta'_s cos\phi'_s sin\phi'_s + C_E cos\phi'_s sin\phi'_s)\hat{y}'_s \\ &+ \left.(C_H sin\theta'_s cos\phi'_s)\hat{z}'_s\right]\frac{e^{-jkr'_s}}{Z_o r'_s} \end{aligned} \tag{7}$$

We will next define the unit normal $\hat{n}$ to the reflector surface. We are considering a parabolic reflector and the surface is defined as

$$r_s = \frac{2F}{1+cos\theta_s} \tag{8}$$

where F is the focal length of the parabola. (8) can be rewritten as

$$0 = 2F - r_s - r_s cos\theta_s \tag{9}$$

The surface of the reflector is then defined as

$$S = 2f - r_s - r_s cos\theta_s \tag{10}$$

The normal of this surface can be determined by taking the gradient of S as follows

$$\bar{n} = \nabla S = \frac{\delta S}{\delta r_s}\hat{r}_s + \frac{1}{r_s}\frac{\delta S}{\delta \theta_s}\hat{\theta}_s + \frac{1}{r_s sin\theta_s}\frac{\delta S}{\delta \phi_s}\hat{\phi}_s \tag{11}$$

we then have

$$\bar{n} = -(1+cos\theta_s)\hat{r}_s + (sin\theta_s)\hat{\theta}_s \tag{12}$$

Now the unit normal is given by

$$\hat{n} = \frac{\bar{n}}{\|\bar{n}\|} \tag{13}$$

We then have, using several trigonometric identities and simplifications

$$\hat{n} = (-sin(\theta_s/2)cos\phi_s)\hat{x}_s - (sin(\theta_s/2)sin\phi_s)\hat{y}_s - (cos(\theta_s/2))\hat{z}_s \tag{14}$$

which is the normal that is required to determine the induced current given in (3). It should be noted that $\hat{x}_s = \hat{x}'_s$, $\hat{y}_s = \hat{y}'_s$ and $\hat{z}_s = \hat{z}'_s$ which is evident in Figure 1. Using (eq2.1.7) and (eq2.1.3) in (eq2.1.3) we have

$$\hat{n} \times \bar{H}^{inc}(\bar{r'}_s) = \begin{vmatrix} \hat{x}_s & \hat{y}_s & \hat{z}_s \\ n_x & n_y & n_z \\ H_x & H_y & H_z \end{vmatrix}$$

$$\hat{n} \times \bar{H}^{inc}(\bar{r'}_s) = (n_y H_z - n_z H_y)\hat{x}_s + (n_z H_x - n_x H_z)\hat{y}_s + (n_x H_y - n_y H_x)\hat{z}_s \tag{15}$$

$$\begin{aligned} (n_y H_z - n_z H_y) &= \big[-C_H sin(\theta_s/2)sin\phi_s sin\theta'_s cos\phi'_s \\ &- C_H cos(\theta_s/2)cos\theta'_s cos\phi'_s sin\phi'_s \\ &+ C_E cos(\theta_s/2)cos\phi'_s sin\phi'_s\big]\frac{e^{-jkr'_s}}{Z_o r'_s} \end{aligned} \tag{16}$$

$$\begin{aligned} (n_z H_x - n_x H_z) &= \big[C_H cos(\theta_s/2)cos^2\phi'_s cos\theta'_s \\ &+ C_E cos(\theta_s/2)sin^2\phi'_s \\ &+ C_H sin(\theta_s/2)cos\phi_s cos\phi'_s sin\theta'_s\big]\frac{e^{-jkr'_s}}{Z_o r'_s} \end{aligned} \tag{17}$$

$$\begin{aligned} (n_x H_y - n_y H_x) &= \big[C_H sin(\theta_s/2)cos\phi_s cos\theta'_s cos\phi'_s sin\phi'_s \\ &- C_E sin(\theta_s/2)cos\phi_s cos\phi'_s sin\phi'_s \\ &- C_H sin(\theta_s/2)sin\phi_s cos^2\phi'_s cos\theta'_s \\ &- C_E sin(\theta_s/2)sin\phi_s sin^2\phi'_s\big]\frac{e^{-jkr'_s}}{Z_o r'_s} \end{aligned} \tag{18}$$

We then have

$$\begin{aligned} \bar{J}(x,y,z) &= 2\hat{n} \times \bar{H}^{inc}(\bar{r'}_s) \\ &= 2(n_y H_z - n_z H_y)\hat{x}_s + 2(n_z H_x - n_x H_z)\hat{y}_s + 2(n_x H_y - n_y H_x)\hat{z}_s \end{aligned} \tag{19}$$

Therefore,

$$\begin{aligned} J_x &= \big[-C_H sin(\theta_s/2)sin\phi_s sin\theta'_s cos\phi'_s \\ &- C_H cos(\theta_s/2)cos\theta'_s cos\phi'_s sin\phi'_s \\ &+ C_E cos(\theta_s/2)cos\phi'_s sin\phi'_s\big]\frac{2e^{-jkr'_s}}{Z_o r'_s} \end{aligned} \tag{20}$$

$$
\begin{aligned}
J_y &= \left[C_H cos(\theta_s/2)cos^2\phi'_s cos\theta'_s \right. \\
&+ C_E cos(\theta_s/2)sin^2\phi'_s \\
&+ \left. C_H sin(\theta_s/2)cos\phi_s cos\phi'_s sin\phi'_s\right] \frac{2e^{-jkr'_s}}{Z_o r'_s}
\end{aligned} \tag{21}
$$

$$
\begin{aligned}
J_z &= \left[C_H sin(\theta_s/2)cos\phi_s cos\theta'_s cos\phi'_s sin\phi'_s \right. \\
&- C_E sin(\theta_s/2)cos\phi_s cos\phi'_s sin\phi'_s \\
&- C_H sin(\theta_s/2)sin\phi_s cos^2\phi'_s cos\theta'_s \\
&- \left. C_E sin(\theta_s/2)sin\phi_s sin^2\phi'_s\right] \frac{2e^{-jkr'_s}}{Z_o r'_s}
\end{aligned} \tag{22}
$$

Now prior to integrating these induced surface currents we must define the primed quantities, namely r'_s, θ'_s and ϕ'_s, in terms of r_s, θ_s, ϕ_s which are the known quantities based on the reflector geometry. The vector $\bar{r'}_s$ is given by

$$\bar{r'}_s = \bar{r}_s - \bar{f} \tag{23}$$

Note that $\bar{f}$ is the vector that defines the feed position and is given as

$$\bar{f} = x_f \hat{x}_s + y_f \hat{y}_s \tag{24}$$

and $\bar{r}_s$ is given as

$$\bar{r}_s = (r_s sin\theta_s cos\phi_s)\hat{x}_s + (r_s sin\theta_s sin\phi_s)\hat{y}_s + (r_s cos\theta_s)\hat{z}_s \tag{25}$$

We then have

$$\bar{r'}_s = (r_s sin\theta_s cos\phi_s - f_x)\hat{x}_s + (r_s sin\theta_s sin\phi_s - f_y)\hat{y}_s + (r_s cos\theta_s)\hat{z}_s \tag{26}$$

Using (26) we can now define r'_s, θ'_s and ϕ'_s as follows

$$r'_s = \sqrt{r'^2_{sx} + r'^2_{sy} + r'^2_{sz}}$$

$$r'_s = \sqrt{r_s^2 + f_x^2 + f_y^2 - 2r_s sin\theta_s (f_x cos\phi_s + f_y sin\phi_s)} \tag{27}$$

$$\theta'_s = tan^{-1}\left[\frac{\sqrt{r'^2_{sx} + r'^2_{sy}}}{r'_{sz}}\right]$$

$$\theta'_s = tan^{-1}\left[\frac{\sqrt{(r_s sin\theta_s cos\phi_s - f_x)^2 + (r_s sin\theta_s sin\phi_s - f_y)^2}}{r_s cos\theta_s}\right] \tag{28}$$

and

$$\phi'_s = tan^{-1}\left(\frac{r'_{sy}}{r'_{sx}}\right)$$

$$\phi'_s = tan^{-1}\left(\frac{r_s sin\theta_s sin\phi_s - f_y}{r_s sin\theta_s cos\phi_s - f_x}\right) \tag{29}$$

Prior to integrating the induced surface currents (20)-(22) we also need to define the two additional terms in the integration shown in equation (1), namely the exponential involving a dot product and the differential surface ds.
First let us define the dot product in the exponential term, we have

$$\bar{r_s} \cdot \hat{r}_p = r_{sx} r_{px} + r_{sy} r_{py} + r_{sz} r_{pz} \tag{30}$$

where

$$\bar{r_s} = (r_s sin\theta_s cos\phi_s)\hat{x}_s + (r_s sin\theta_s sin\phi_s)\hat{y}_s + (r_s cos\theta_s)\hat{z}_s$$

and

$$\hat{r}_p = (sin\theta_p cos\phi_p)\hat{x}_p + (sin\theta_p sin\phi_p)\hat{y}_p + (cos\theta_p)\hat{z}_p$$

Before performing the dot product we must first express $\hat{r}_p$ in the x_s, y_s, z_s coordinate system, which gives

$$\hat{r}_p = (sin\theta_p cos\phi_p)\hat{x}_s - (sin\theta_p sin\phi_p)\hat{y}_s - (cos\theta_p)\hat{z}_s$$

We then have

$$\bar{r_s} \cdot \hat{r}_p = r_s sin\theta_s cos\phi_s sin\theta_p cos\phi_p - r_s sin\theta_s sin\phi_s sin\theta_p sin\phi_p - r_s cos\theta_s cos\theta_p \tag{31}$$

Next the differential surface area ds is defined as

$$ds = r_s^2 sec(\theta_s/2) sin(\theta_s) d\theta_s d\phi_s \tag{32}$$

Expressions (20-22) along with (31) and (32) can now be numerically integrated separately yielding

$$F_x = \int\int_S J_x e^{jk\bar{r_s}\cdot\hat{r}_p} ds \tag{33}$$

$$F_y = \int\int_S J_y e^{jk\bar{r_s}\cdot\hat{r}_p} ds \tag{34}$$

$$F_z = \int\int_S J_z e^{jk\bar{r_s}\cdot\hat{r}_p} ds \tag{35}$$

Now that the integral has been evaluated we can finish evaluating expression(1). To do this we first must evaluate the following:

$$(\bar{\bar{I}} - \hat{r}_p \hat{r}_p) \cdot \bar{F} \tag{36}$$

which can be rewritten as

$$\bar{F} - \hat{r}_p (\hat{r}_p \cdot \bar{F}) \tag{37}$$

It should be noted that $\bar{F}$ is defined in the x_s, y_s, z_s coordinate system and $\hat{r}_p$ is defined in the x_p, y_p, z_p system, as a consequence $\hat{r}_p$ in the x_s, y_s, z_s system is given as

$$\hat{r}_p = (sin\theta_p cos\phi_p)\hat{x}_s - (sin\theta_p sin\phi_p)\hat{y}_s - (cos\theta_p)\hat{z}_s \tag{38}$$

We then have

$$\hat{r}_p \cdot \bar{F} = (F_x sin\theta_p cos\phi_p) + (-F_y sin\theta_p sin\phi_p) + (-F_y cos\theta_p) \tag{39}$$

Then

$$\begin{aligned}\hat{r}_p(\hat{r}_p\cdot\bar{F}) &= (F_x sin^2\theta_p cos^2\phi_p - F_y sin^2\theta_p sin\phi_p cos\phi_p - F_z cos\theta_p sin\theta_p cos\phi_p)\hat{x}_p \\ &+ (F_x sin^2\theta_p cos\phi_p sin\phi_p - F_y sin^2\theta_p sin^2\phi_p - F_z cos\theta_p sin\theta_p sin\phi_p)\hat{y}_p \\ &+ (F_x sin\theta_p cos\theta_p cos\phi_p - F_y sin\theta_p cos\theta_p sin\phi_p - F_z cos^2\theta_p)\hat{z}_p \end{aligned} \quad (40)$$

We can now evaluate $\bar{F}-\hat{r}_p(\hat{r}_p\cdot\bar{F})$ however $\bar{F}$ is defined in the x_s,y_s,z_s system and $\hat{r}_p(\hat{r}_p\cdot\bar{F})$ is defined in x_p,y_p,z_p system. We can then define $\bar{F}$ in the x_p,y_p,z_p system and we have $\bar{F}=F_x\hat{x}_p-F_y\hat{y}_p-F_z\hat{z}_p$, we then have

$$\begin{aligned}\bar{F}-\hat{r}_p(\hat{r}_p\cdot\bar{F}) &= (F_x - F_x sin^2\theta_p cos^2\phi_p - F_y sin^2\theta_p sin\phi_p cos\phi_p - F_z cos\theta_p sin\theta_p cos\phi_p)\hat{x}_p \\ &- (F_y + F_x sin^2\theta_p cos\phi_p sin\phi_p - F_y sin^2\theta_p sin^2\phi_p - F_z cos\theta_p sin\theta_p sin\phi_p)\hat{y}_p \\ &- (F_z + F_x sin\theta_p cos\theta_p cos\phi_p - F_y sin\theta_p cos\theta_p sin\phi_p - F_z cos^2\theta_p)\hat{z}_p \end{aligned} \quad (41)$$

Finally the reflector pattern is given as

$$\begin{aligned}E_x^{rad}(\theta_p,\phi_p) &= -jkZ_o\frac{e^{-jkr_p}}{r_p}[(\bar{F}-\hat{r}_p(\hat{r}_p\cdot\bar{F}))\cdot\hat{x}_p] \\ E_y^{rad}(\theta_p,\phi_p) &= -jkZ_o\frac{e^{-jkr_p}}{r_p}[(\bar{F}-\hat{r}_p(\hat{r}_p\cdot\bar{F}))\cdot\hat{y}_p] \\ E_z^{rad}(\theta_p,\phi_p) &= -jkZ_o\frac{e^{-jkr_p}}{r_p}[(\bar{F}-\hat{r}_p(\hat{r}_p\cdot\bar{F}))\cdot\hat{z}_p] \end{aligned} \quad (42)$$

The radiation pattern can also be expressed in its co-pol and x-pol components as follows [2]

$$\begin{aligned}\bar{E}_{co}^{rad}(\theta_p,\phi_p) &= [-(1-cos\theta_p)sin\phi_p cos\phi_p]E_x^{rad} \\ &+ [1-sin^2\phi_p(1-cos\theta_p)]E_y^{rad} \\ &- [sin\theta_p sin\phi_p]E_z^{rad} \end{aligned} \quad (43)$$

$$\begin{aligned}\bar{E}_x^{rad}(\theta_p,\phi_p) &= [1-cos^2\phi_p(1-cos\theta_p)]E_x^{rad} \\ &- [(1-cos\theta_p)sin\phi_p cos\phi_p]E_y^{rad} \\ &- [sin\theta_p cos\phi_p]E_z^{rad} \end{aligned} \quad (44)$$

2.2 Ludwig's Method

The Ludwig method is one of the first numerical techniques to evaluate the radiation integral, with the first work dating back to 1968 [3]. Ludwig's method also makes use of the PO approximation and its starting point is the radiation integral given in equation (1) as follows

$$\bar{E}^{rad}(\theta_p,\phi_p) = -jkZ_o\frac{e^{-jkr_p}}{r_p}(\bar{\bar{I}}-\hat{r}_p\hat{r}_p)\int\int_S \bar{J}_s e^{jk\bar{r}_s\cdot\hat{r}_p}ds \quad (45)$$

The integration of equation (45) can be rewritten in a more general form as

$$\bar{F}(\theta_p,\phi_p) = \int\int_S \bar{K}(\theta_s,\phi_s)e^{jk\gamma(\theta_s,\phi_s,\theta_p,\phi_p)}d\theta d\phi \quad (46)$$

$\bar{K}(\theta_s,\phi_s)$ is given by equations (20)-(22), from the previous section, with the exponential terms removed and the non-differential portion of equation (32). The exponential term will be lumped with the $\gamma(\theta_s,\phi_s,\theta_p,\phi_p)$ term of equation (46). We then have $\bar{K}(\theta_s,\phi_s)$, in its three cartesian components given as

$$\begin{aligned} K_x(\theta_s,\phi_s) &= \big[-C_H sin(\theta_s/2)sin\phi_s sin\theta'_s cos\phi'_s \\ &- C_H cos(\theta_s/2)cos\theta'_s cos\phi'_s sin\phi'_s \\ &+ C_E cos(\theta_s/2)cos\phi'_s sin\phi'_s\big]\frac{2r_s^2 sec(\theta_s/2)sin(\theta_s)}{Z_o r'_s} \end{aligned} \tag{47}$$

$$\begin{aligned} K_y(\theta_s,\phi_s) &= \big[C_H cos(\theta_s/2)cos^2\phi'_s cos\theta'_s \\ &+ C_E cos(\theta_s/2)sin^2\phi'_s \\ &+ C_H sin(\theta_s/2)cos\phi_s cos\phi'_s sin\theta'_s\big]\frac{2r_s^2 sec(\theta_s/2)sin(\theta_s)}{Z_o r'_s} \end{aligned} \tag{48}$$

$$\begin{aligned} K_z(\theta_s,\phi_s) &= \big[C_H sin(\theta_s/2)cos\phi_s cos\theta'_s cos\phi'_s sin\phi'_s \\ &- C_E sin(\theta_s/2)cos\phi_s cos\phi'_s sin\phi'_s \\ &- C_H sin(\theta_s/2)sin\phi_s cos^2\phi'_s cos\theta'_s \\ &- C_E sin(\theta_s/2)sin\phi_s sin^2\phi'_s\big]\frac{2r_s^2 sec(\theta_s/2)sin(\theta_s)}{Z_o r'_s} \end{aligned} \tag{49}$$

and $\gamma(\theta_s,\phi_s,\theta_p,\phi_p)$ will be given by

$$\gamma(\theta_s,\phi_s,\theta_p,\phi_p) = r'_s + \bar{r_s}\cdot\hat{r}_p \tag{50}$$

$\bar{r_s}\cdot\hat{r}_p$ is given by equation (31) of the previous section.

Now consider a fixed output radiation pattern point (θ_p^p,ϕ_p^q) for a single cartesian component of F and K, the integration is then

$$F_{pq} = \int_0^{\theta_s^M}\int_0^{\phi_s^N} K(\theta_s,\phi_s)e^{jk\gamma(\theta_s,\phi_s)}d\theta d\phi \tag{51}$$

Here K is specified on an $M\times N$ integration grid for the points (θ_s^m,ϕ_s^n). We can now consider how the integrand, $(K(\theta_s,\phi_s)e^{jk\gamma(\theta_s,\phi_s)})$, behaves over an incremental surface ΔS_{mn} on the surface of the reflector where

$$\Delta S_{mn} = \{(\theta_s,\phi_s) : \theta_s^m \le \theta_s \le \theta_s^{m+1}, \phi_s^n \le \phi_s \le \phi_s^{n+1}\} \tag{52}$$

If ΔS_{mn} is physically on the order of a wavelength the electrical path length term $(jk\gamma(\theta_s,\phi_s))$ cannot vary by more than 2π. Now consider the behavior of $K(\theta_s,\phi_s)$ over the incremental area ΔS_{mn}. Since typically electromagnetic fields do not have abrupt changes over a distance on the order of a wavelength, we can then say that the integrand, $(K(\theta_s,\phi_s)e^{jk\gamma(\theta_s,\phi_s)})$,

will not vary abruptly and be well behaved over the incremental area ΔS_{mn}. This being said K and γ can be separately approximated by a linear function over ΔS_{mn}. We then have

$$\begin{aligned} K(\theta_s,\phi_s) &\simeq a_{mn}+b_{mn}(\theta_s-\theta_s^m) \;+\; c_{mn}(\phi_s-\phi_s^n) \\ \gamma(\theta_s,\phi_s) &\simeq \alpha_{mn}+\beta_{mn}(\theta_s-\theta_s^m) \;+\; \xi_{mn}(\phi_s-\phi_s^n) \\ for(\theta_s,\phi_s) &\in \Delta S_{mn} \end{aligned} \tag{53}$$

The method used in [3], for determining the coefficients of (53) is to use a best fit mean-squared plane to the values of the functions, $(K(\theta_s,\phi_s)e^{jk\gamma(\theta_s,\phi_s)})$, at the corners of ΔS_{mn}. This gives

$$\begin{aligned} a_{mn} &= \frac{1}{4}\left[3K(\theta_s^m,\phi_s^n)-K(\theta_s^{m+1},\phi_s^{n+1})\right. \\ &+ \left. K(\theta_s^{m+1},\phi_s^n)+K(\theta_s^m,\phi_s^{n+1})\right] \\ b_{mn} &= \frac{1}{2\Delta\theta_s^m}\left[K(\theta_s^{m+1},\phi_s^n)-K(\theta_s^m,\phi_s^n)\right. \\ &+ \left. K(\theta_s^{m+1},\phi_s^{n+1})-K(\theta_s^m,\phi_s^{n+1})\right] \\ c_{mn} &= \frac{1}{2\Delta\phi_s^n}\left[K(\theta_s^m,\phi_s^{n+1})-K(\theta_s^m,\phi_s^n)\right. \\ &+ \left. K(\theta_s^{m+1},\phi_s^{n+1})-K(\theta_s^{m+1},\phi_s^n)\right] \end{aligned} \tag{54}$$

and

$$\begin{aligned} \alpha_{mn} &= \frac{1}{4}\left[3\gamma(\theta_s^m,\phi_s^n)-\gamma(\theta_s^{m+1},\phi_s^{n+1})\right. \\ &+ \left. \gamma(\theta_s^{m+1},\phi_s^n)+\gamma(\theta_s^m,\phi_s^{n+1})\right] \\ \beta_{mn} &= \frac{1}{2\Delta\theta_s^m}\left[\gamma(\theta_s^{m+1},\phi_s^n)-\gamma(\theta_s^m,\phi_s^n)\right. \\ &+ \left. \gamma(\theta_s^{m+1},\phi_s^{n+1})-\gamma(\theta_s^m,\phi_s^{n+1})\right] \\ \xi_{mn} &= \frac{1}{2\Delta\phi_s^n}\left[\gamma(\theta_s^m,\phi_s^{n+1})-\gamma(\theta_s^m,\phi_s^n)\right. \\ &+ \left. \gamma(\theta_s^{m+1},\phi_s^{n+1})-\gamma(\theta_s^{m+1},\phi_s^n)\right] \end{aligned} \tag{55}$$

where

$$\begin{aligned} \Delta\theta_s^m &= \theta_s^{m+1}-\theta_s^m \\ \Delta\phi_s^n &= \phi_s^{n+1}-\phi_s^n \end{aligned} \tag{56}$$

With (53)-(56) we can now perform the integration in closed form, and over the incremental surface area ΔS_{mn} we get the following contribution

$$\begin{aligned} \Delta F_{mn} &= \left\{ a_{mn}\left[\frac{e^{jk\beta_{mn}\Delta\theta_s^m}-1}{jk\beta_{mn}}\right]\left[\frac{e^{jk\xi_{mn}\Delta\phi_s^n}-1}{jk\xi_{mn}}\right]\right. \\ &+ b_{mn}\left[\frac{\Delta\theta_s^m}{jk\beta_{mn}}e^{jk\beta_{mn}\Delta\theta_s^m}-\frac{e^{jk\beta_{mn}\Delta\theta_s^m}-1}{(jk\beta_{mn})^2}\right]\left[\frac{e^{jk\xi_{mn}\Delta\phi_s^n}-1}{jk\xi_{mn}}\right] \\ &+ \left. c_{mn}\left[\frac{e^{jk\beta_{mn}\Delta\theta_s^m}-1}{jk\beta_{mn}}\right]\left[\frac{\Delta\phi_s^n}{jk\xi_{mn}}e^{jk\xi_{mn}\Delta\phi_s^n}-\frac{e^{jk\xi_{mn}\Delta\phi_s^n}-1}{(jk\xi_{mn})^2}\right]\right\} \end{aligned} \tag{57}$$

To get the value of the integral we simply need to sum all these contributions over m and n. This can then be repeated for each radiation pattern observation point. Therefore in order to obtain the complete desired radiation pattern we would then have to perform $M \times N \times P \times Q$ computations. However it should be noted that equations (47)-(49), defining K are independent of the output observation angles θ_p and ϕ_p, therefore a_{mn}, b_{mn} and c_{mn} only need to be computed once. This will significantly reduce the computation time, however it does require some memory allocation for storage of these coefficients. We can now use equations (42)-(44), from the previous section to compute the complete co-pol and x-pol radiation patterns.

3 Results

Each of the formulations of the previous section were coded in MATLAB. We will now look at radiation patterns for parabolic reflector antennas using these modelling methods. We will specifically examine the co-pol and x-pol patterns in the two principal planes for three different configurations, namely

- Reflector #1
  - $D = 20\lambda$
  - $F/D = 0.8$
  - Feed Location: $(x,y,z) = (0,0,0)$
- Reflector #2
  - $D = 20\lambda$
  - $F/D = 0.8$
  - Feed Location: $(x,y,z) = (\lambda/4,0,0)$
- Reflector #3
  - $D = 20\lambda$
  - $F/D = 0.8$
  - Feed Location: $(x,y,z) = (0,\lambda/2,0)$

The radiation patterns for each of the reflector configurations, and each of the methods described in the previous section, will be compared to the results from a commercial reflector modelling tool, GRASP by Ticra [4]. We will be looking at the radiation patterns for 301 points between $-30^\circ \leq \theta_p \leq 30^\circ$

3.1 Feed Pattern

For the purpose of this modelling the feed pattern is approximated by equation (4), recall

$$\bar{E}^{inc}(\bar{r'}_s) = \left[(C_E(\theta'_s)sin\phi'_s)\hat{\theta}'_s + (C_H(\theta'_s)cos\phi'_s)\hat{\phi}'_s\right] \frac{e^{-jkr'_s}}{r'_s} \tag{58}$$

This approximate feed pattern will make use of the principal plane patterns (E-plane $\rightarrow$ $\phi_s' = 0^\circ$ and H-plane $\rightarrow$ $\phi_s' = 90^\circ$) where $C_E(\theta_s') = cos(\theta_s')^{qE}$ and $C_H(\theta_s') = cos(\theta_s')^{qH}$ represent the E and H-plane patterns respectively. We must then select qE and qH in order to match the principal plane patterns of the actual feed that we are modelling. This feed approximation will model the field from the feed, for any angle ϕ_s' by means of interpolation. As a general rule [5], to achieve peak aperture efficiency, the edge illumination should be approximately -11dB. We will then select qE and qH that would give us this desired edge illumination. For the purpose of this modelling we will look at a balanced feed where $qE = qH$. This type of balanced feed has a very low cross-polarized field [6].

For the reflector we are considering (i.e. Reflector #1), we will have $qE = qH = 6.5$ and with this we have the feed pattern shown in Figure 3. Note that θ_o is the reflector sub-

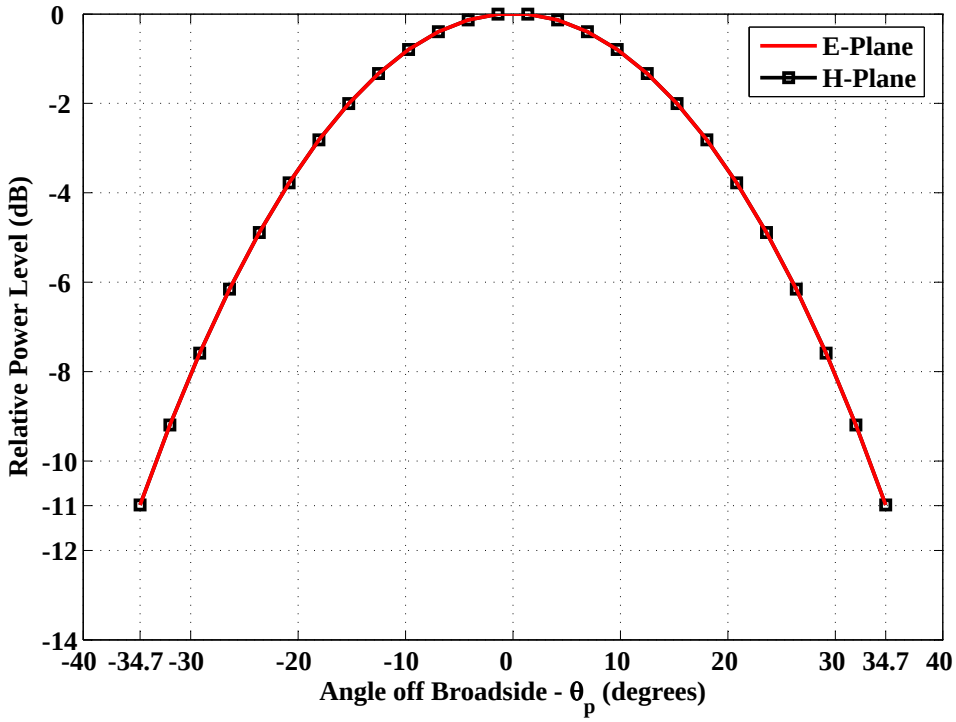

Figure 3: *Feed Pattern using (58) with* $qE = qH = 6.5$

tended angle, depicted in Figure 4. The subtended angle is calculated using the following expression

$$\theta_o = 2\tan^{-1}\frac{D}{4F}; \tag{59}$$

In the case of Reflector #1 the subtended angle $\theta_o = 34.708$. We can then see from Figure 3 that the edge illumination is approximately $-11dB$. It should also be noted that for the other two reflector configuration, with the laterally displaced feeds, we will not be modifying the feed model (i.e. qE and qH will remain unchanged). As a result we will no longer have a

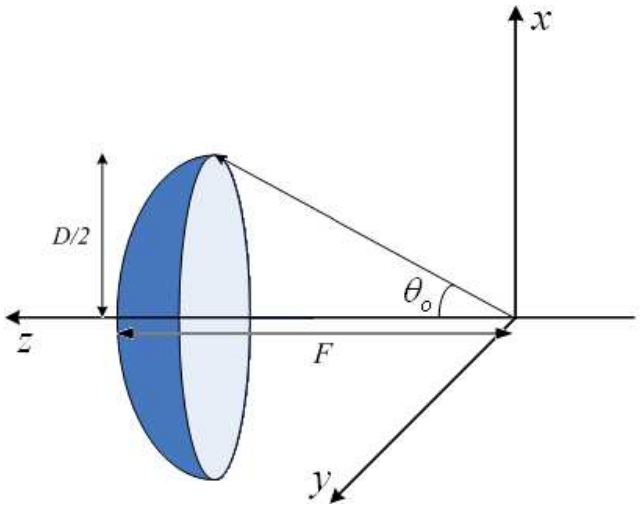

Figure 4: *Reflector Geometry with Subtended Angle* θ_o

uniform $-11dB$ edge illumination on the reflector, because of the feed displacement from the center axis.

3.2 PO/Surface Current Method

This method, as stated previously, is based on numerically integrating the surface currents induced on the reflector. In order to perform these integrations we make use of MATLAB's internal integration routines, namely *dlbquad* which uses a recursive adaptive Simpson quadrature to numerically evaluate the desired integral. This MATLAB routine will approximate the integral within a defined absolute error tolerance which by default is set to 1×10^{-6}. This error tolerance governs the speed and accuracy of the numerical integration. A large error tolerance will result in rapid computation with decreased accuracy and a small error tolerance will provide very accurate results at the expense of evaluation speed. In this section we will look how this error tolerance affects the accuracy and computing speed of the radiation patterns.

3.2.1 Reflector #1

We will first look at how the error tolerance will affect the accuracy of the radiation pattern. In Figures 5-9 we see the co-pol pattern for a single pattern cut ($\phi_p = 90^\circ$) for 5 different error tolerances. The computed patterns are compared to the pattern computed using GRASP.

We can clearly see from Figures 5-9 how the error tolerance affects the accuracy of the

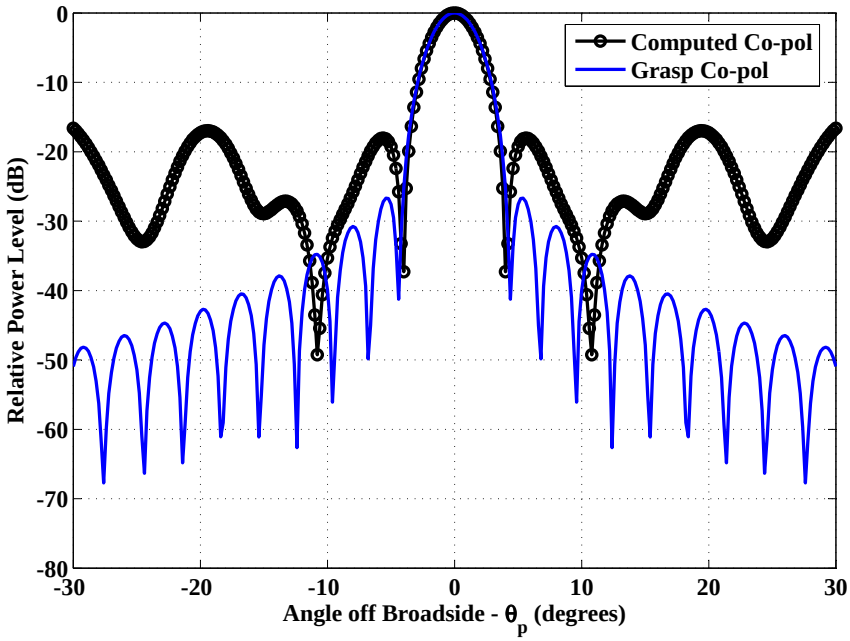

Figure 5: *Radiation Pattern for Reflector #1, MATLAB Numerical Integration, Error Tolerance* $= 1 \times 10^{-4}$, $\phi_p = 90^\circ$

results. An error tolerance of 1×10^{-8} was requires to accurately model the radiation pattern. Table 1 shows the CPU time (3.0GHz Pentium IV, 512MB RAM) required to compute each of the patterns for the different error tolerances. We can also see in this table the value

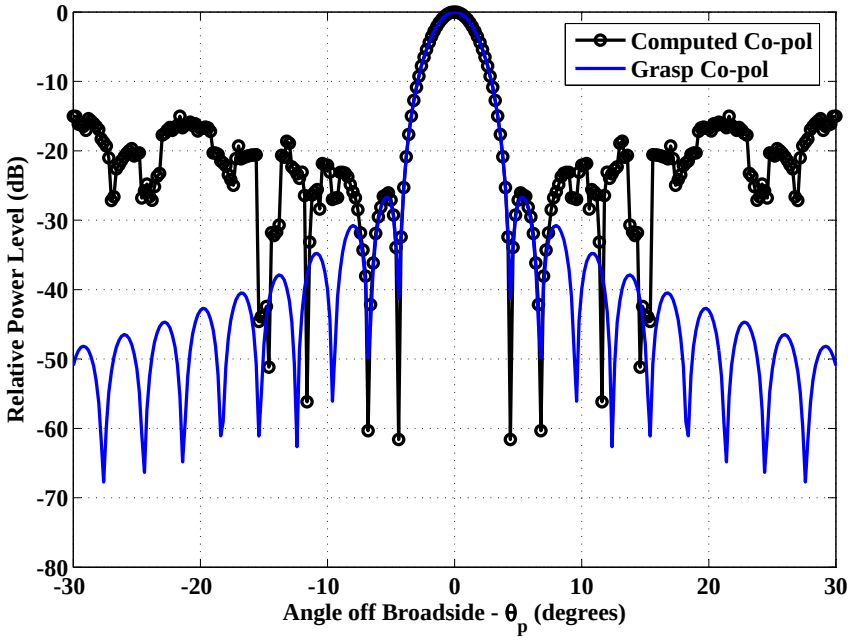

Figure 6: *Radiation Pattern for Reflector #1, MATLAB Numerical Integration, Error Tolerance* $= 1 \times 10^{-5}$, $\phi_p = 90^\circ$

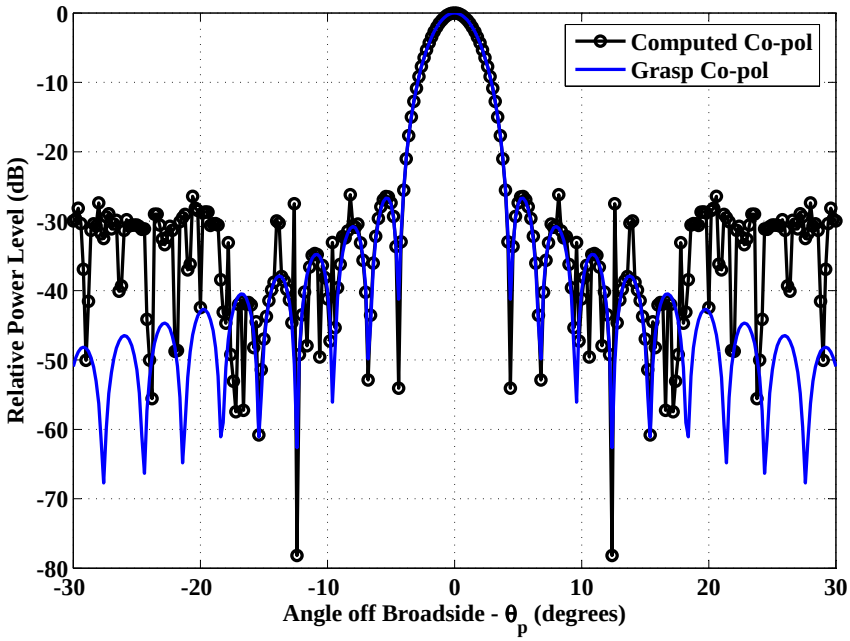

Figure 7: *Radiation Pattern for Reflector #1, MATLAB Numerical Integration, Error Tolerance* $= 1 \times 10^{-6}$, $\phi_p = 90^\circ$

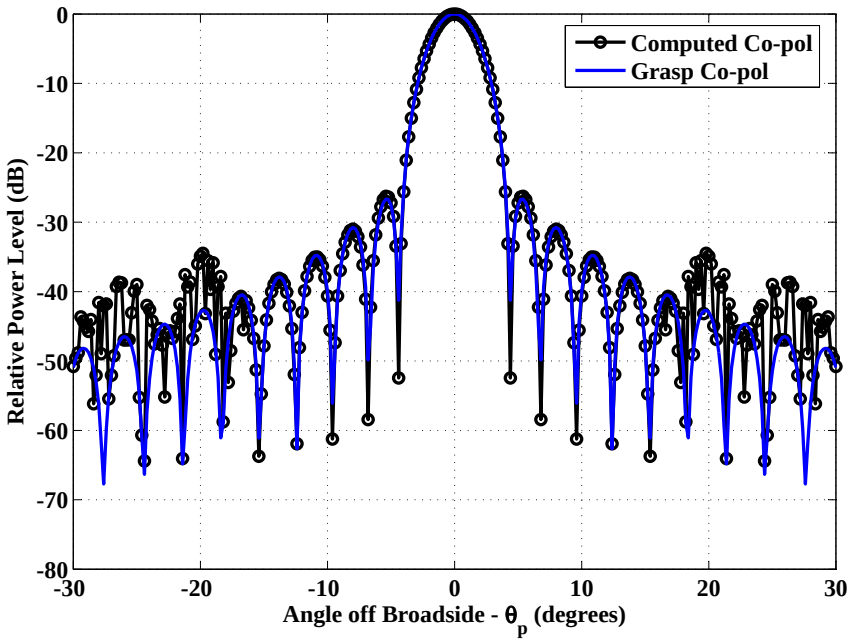

Figure 8: *Radiation Pattern for Reflector #1, MATLAB Numerical Integration, Error Tolerance* $= 1 \times 10^{-7}$, $\phi_p = 90^\circ$

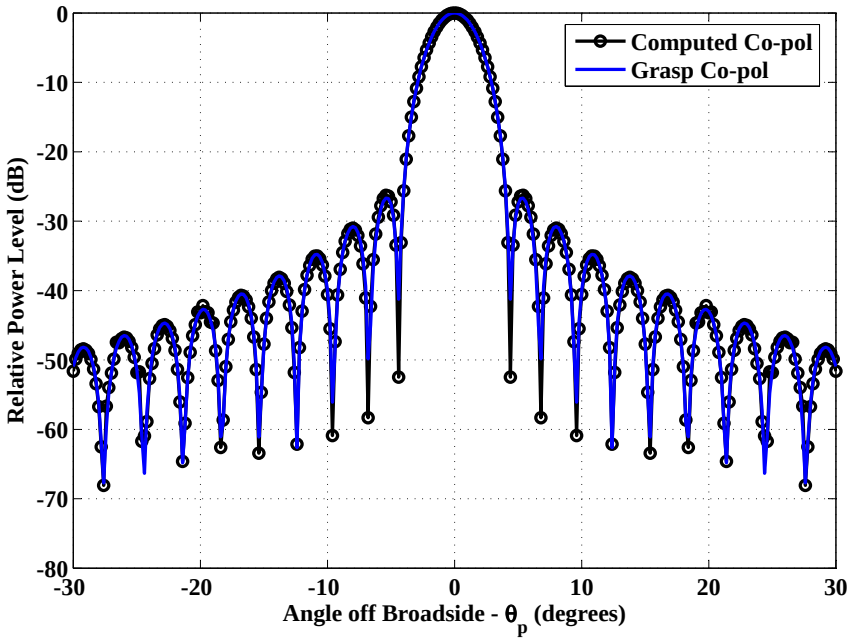

Figure 9: *Radiation Pattern for Reflector #1, MATLAB Numerical Integration, Error Tolerance* $= 1 \times 10^{-8}$, $\phi_p = 90^\circ$

Table 1: *Computation Time (3.0GHz Pentium IV, 512MB RAM)*

Error Tolerance	HPBW	CPU Time
1×10^{-4}	3.38°	18sec
1×10^{-5}	3.38°	21sec
1×10^{-6}	3.38°	49sec
1×10^{-7}	3.38°	2min11sec
1×10^{-8}	3.38°	5min22sec

for the half power beam width for the different error tolerances. Notice that the HPBW is unchanged for the different error tolerances, this is because the PO/Surface Current Method is quite accurate for the main beam and the first couple of sidelobes. However, if we wish to use this technique to calculate the further out sidelobes we require a much smaller error tolerance, which is clearly depicted here. In Figures 10 and 11 we see the E and H-plane radiation patterns respectively. Notice that the cross-polarization field does not appear in these figures, recall that the feed model we are using is a balanced feed where we have a very low cross-polarized field level. Hence the reflector radiation pattern will have a very low cross-pol level.

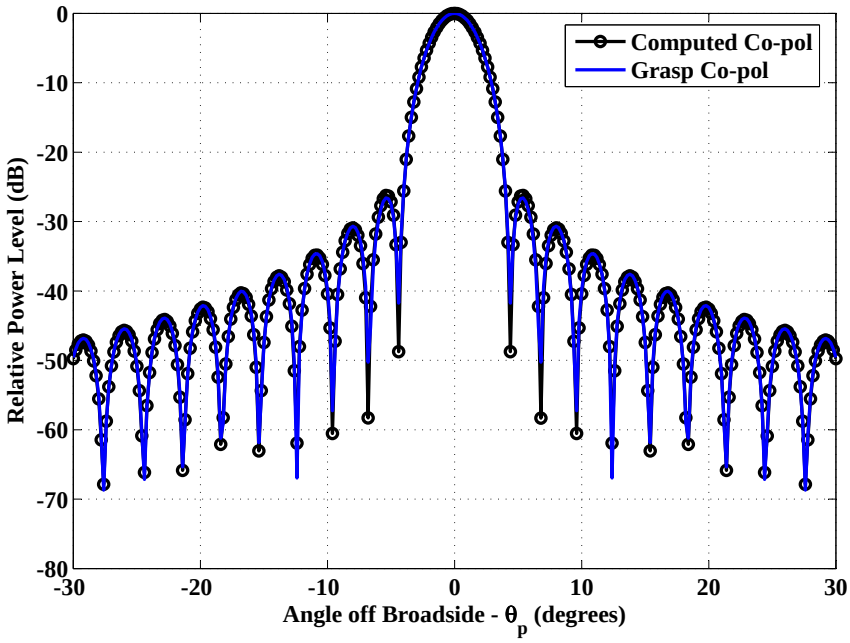

Figure 10: *Radiation Pattern for Reflector #1,* $\phi_p = 0^{\circ}$*, MATLAB Numerical Integration*

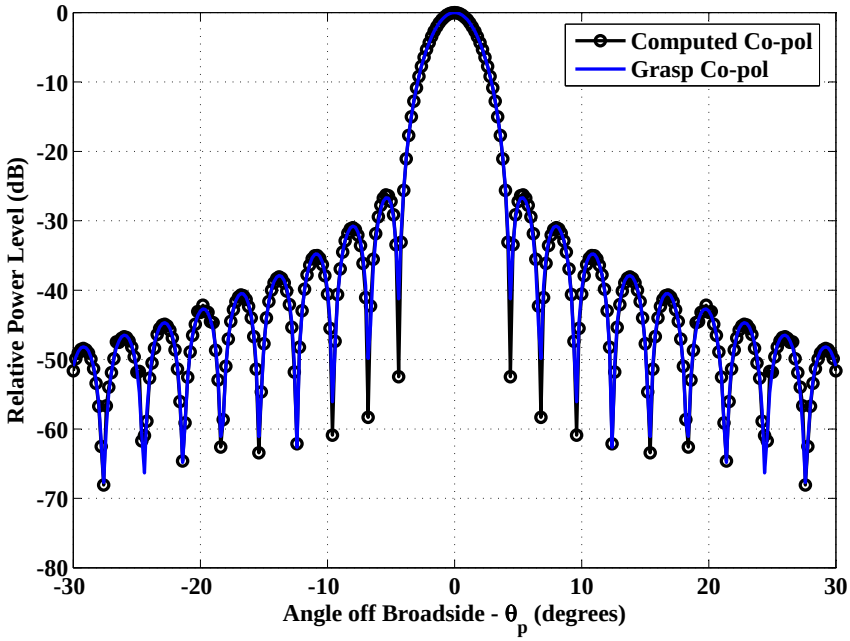

Figure 11: *Radiation Pattern for Reflector #1,* $\phi_p = 90^\circ$*, MATLAB Numerical Integration*

3.2.2 Reflector #2

We will now look at the results of this model with a displace feed. The feed is being displaced along the x-axis or alternatively in the E-plane ($\phi_p = 0^\circ$). We would then expect to the main beam of the reflector to have a squint in the E-plane. This amount of squint can be quantified with the beam deviation factor [5] (BDF):

$$BDF = \frac{\theta_B}{\theta_F} = \frac{1 + 0.36\left[4\frac{F}{D}\right]^{-2}}{1 + \left[4\frac{F}{D}\right]^{-2}} \tag{60}$$

where θ_B is the reflector main beam scan angle and θ_F is the feed displacement angle. For our present reflector configuration we would have a $BDF = 0.943$ which would yield a main beam squint of 0.844°. Figures 12 and 13 show the E and H-plane radiation patterns for this reflector configuration respectively. We can see from Figure 12 that there is a beam

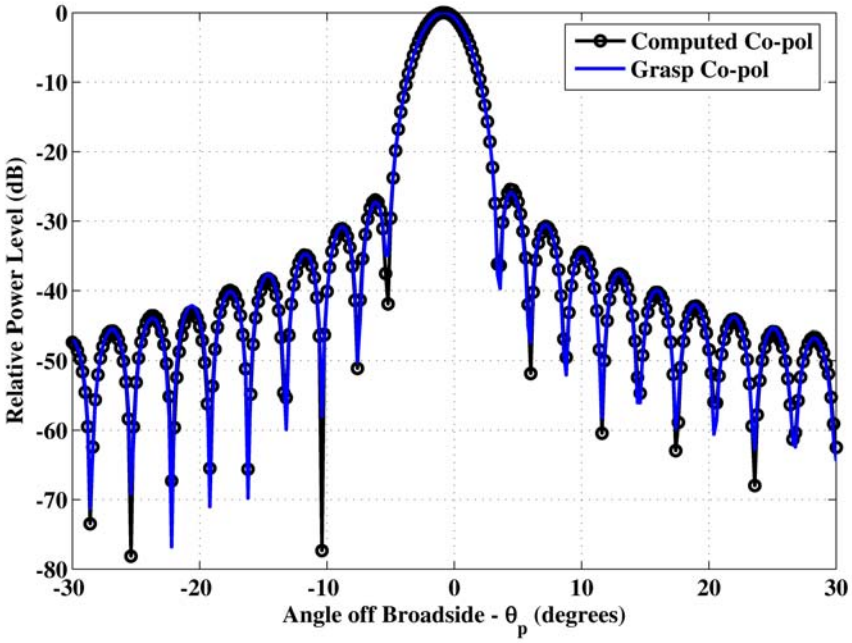

Figure 12: *Radiation Pattern for Reflector #2, $\phi_p = 0^\circ$, MATLAB Numerical Integration*

squint, however on this scale it is not apparent on the exact amount of squint. Figure 14 is a zoomed-in view of the main beam peak, we can see here that the amount of beam squint predicted using equation (60) is quite accurate, the model predicted a main beam at -0.84° off broadside. We can also see in Figure 13 that the cross-pol level has increased. This is expected because the of the feed displacement. We can however see that there is a significant difference between the computed cross-pol level and what is predicted in GRASP.

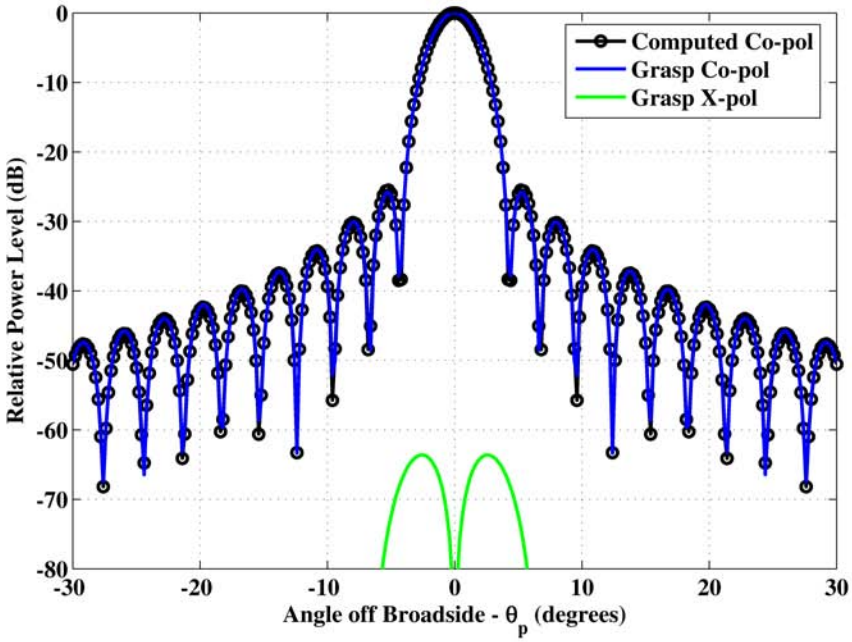

Figure 13: *Radiation Pattern for Reflector #2, $\phi_p = 90^\circ$, MATLAB Numerical Integration*

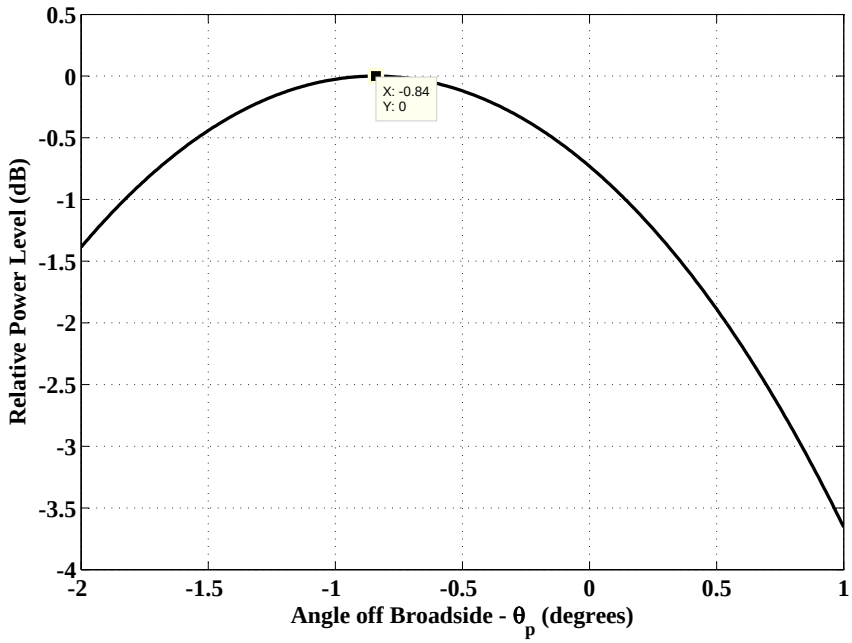

Figure 14: *Radiation Pattern Beam Squint (Reflector #2), MATLAB Numerical Integration*

3.2.3 Reflector #3

We will now look at the final reflector configuration where we have a feed displacement along the y-axis. As opposed to the previous case, we will now see a main beam squint in the H-plane pattern. As before the BDF remains the same ($BDF = 0.943$), however with the feed displaced by $\lambda/2$ we should see a main beam squint of 1.69°. In Figures 15 and 16 show the E and H-plane patterns respectively. Figure 17 is a zoomed in view of the main

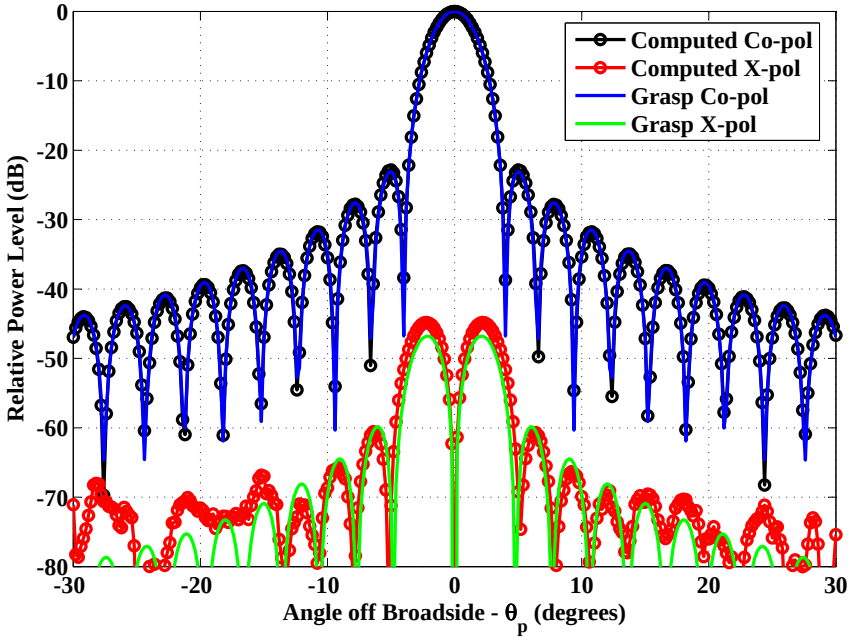

Figure 15: *Radiation Pattern for Reflector #3, MATLAB Numerical Integration, Error Tolerance* $= 1 \times 10^{-8}$, $\phi_p = 0^\circ$

beam peak, we can see here that the amount of beam squint predicted is accurate, the model predicted a main beam at 1.69° off broadside. It should also be noted that because we have a feed displacement the integrand of the radiation integral contains additional amplitude and phase terms. Therefore the *dblquad* routine in MATLAB does not converge as quickly as it does when there is no feed displacement. As a result of this the patterns shown in Figures 15 and 16 took approximately 16 minutes of CPU time to compute. We should also notice that the cross-pol levels are ”noisy” in Figures 15 and 16. To achieve more accurate cross-pol patterns we would have to decrease the error tolerance. Figures 18 and 19 show the principal plane patterns computed using an error tolerance of 1×10^{-9}. We can see that the co and cross-pol patterns are predicted accurately. However because of the decreased error tolerance the CPU time required to compute each of these patterns was well over one hour.

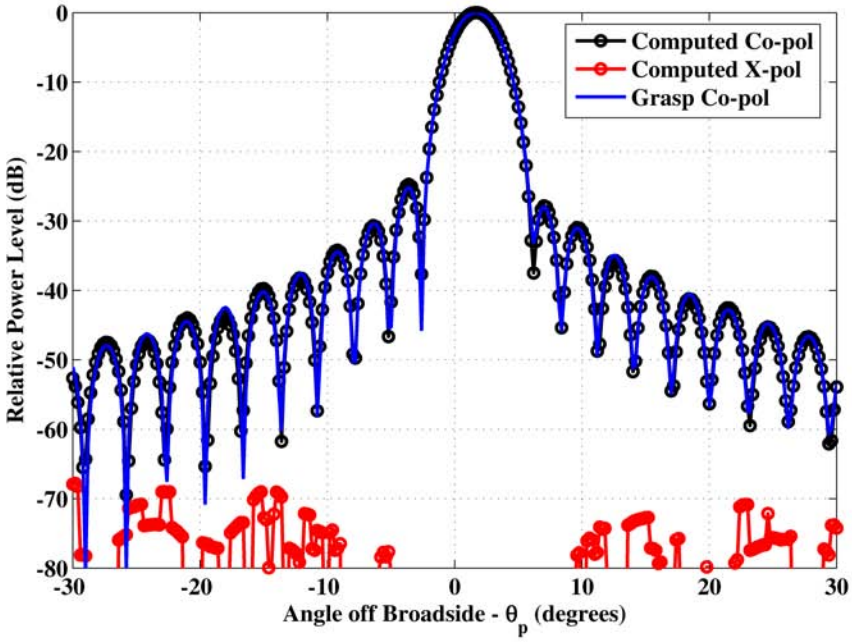

Figure 16: *Radiation Pattern for Reflector #3, MATLAB Numerical Integration, Error Tolerance* $= 1 \times 10^{-8}$, $\phi_p = 90^\circ$

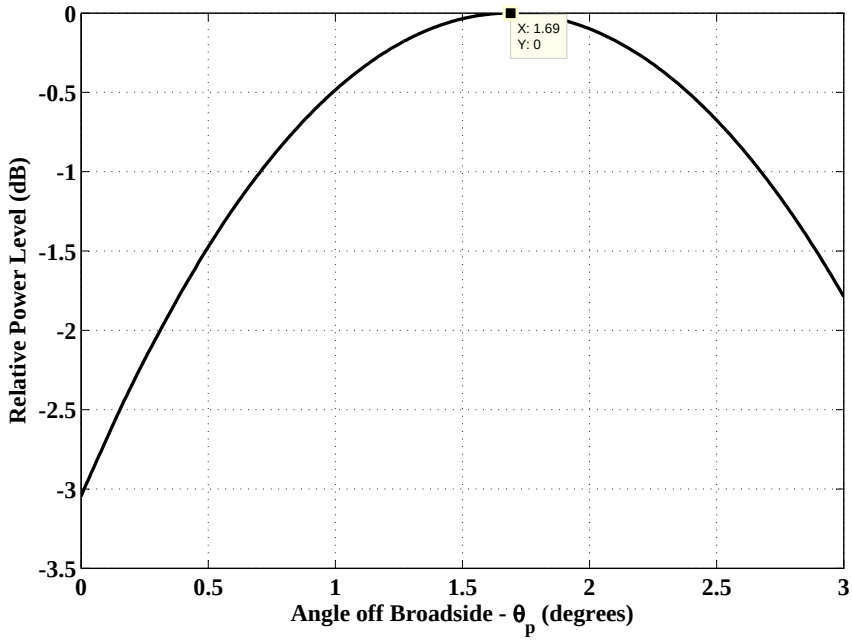

Figure 17: *Radiation Pattern Beam Squint (Reflector #3), MATLAB Numerical Integration*

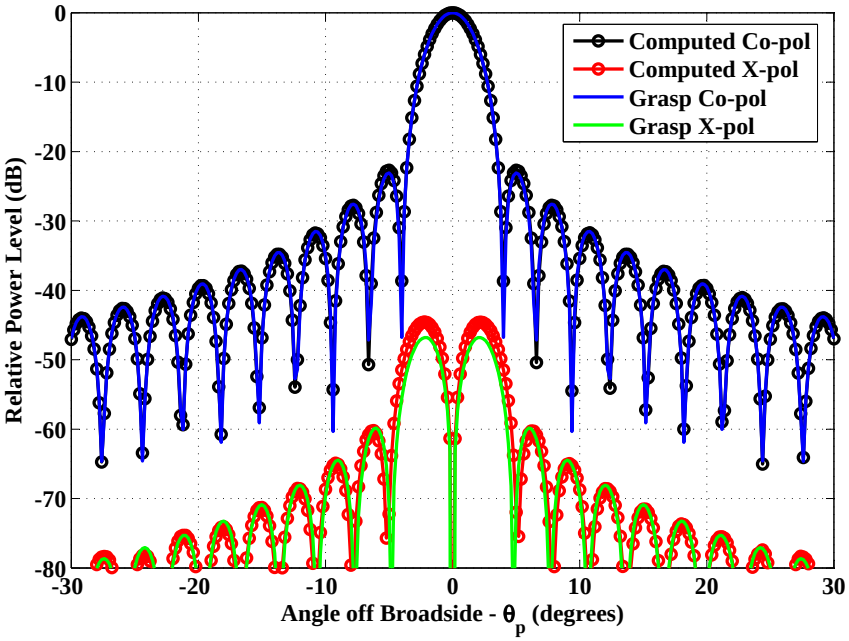

Figure 18: *Radiation Pattern for Reflector #3, MATLAB Numerical Integration, Error Tolerance* $= 1 \times 10^{-9}$, $\phi_p = 0^\circ$

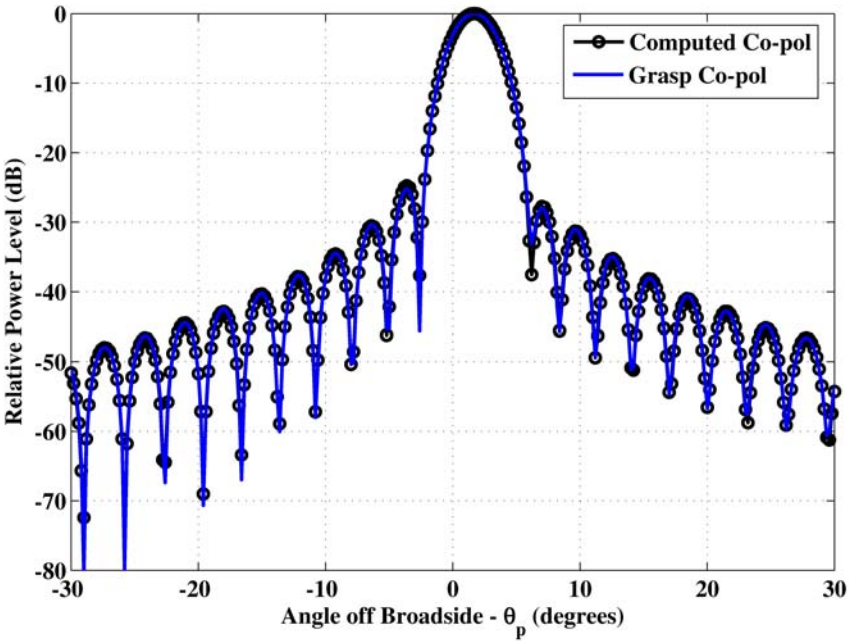

Figure 19: *Radiation Pattern for Reflector #3, MATLAB Numerical Integration, Error Tolerance* $= 1 \times 10^{-9}$, $\phi_p = 90^\circ$

3.3 Ludwig's Method

As stated previously this method divides the integration surface into a grid ($M \times N$). Each cell (incremental surface area) of this grid has a dimension on the order of $\lambda \times \lambda$. The amplitude and phase of the integrand of the radiation integral is then approximated by a plane over this incremental surface area. If we select the size of this incremental area such that its dimensions are on the order of one wavelength, we should have accurate results. To do this we need to determine $\Delta\theta_s^m$ and $\Delta\phi_s^n$. Consider Figure 20. Figure 20 (a) shows the

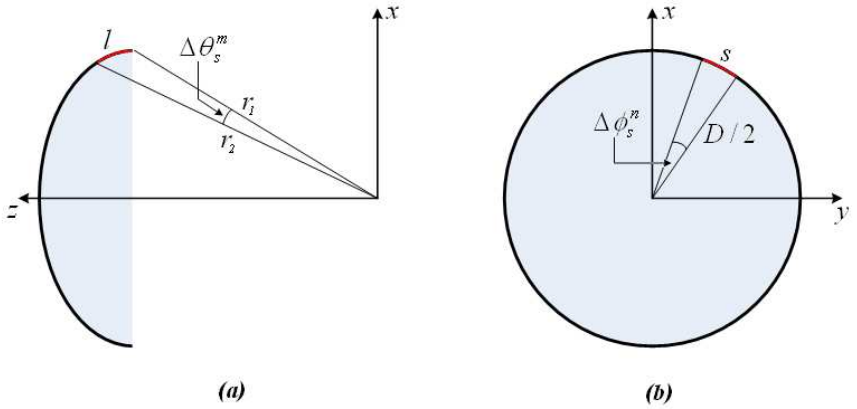

Figure 20: *Reflector Geometry (a) Projected x-z plane (b) Projected x-y plane*

projected x-z plane. If l is small enough we can safely approximate that $r_1 = r_2$. We can then determine $\Delta\theta_s^m$ by setting $l = \lambda$ and using the formula for an arc length, given as

$$l = r\Delta\theta_s^m \tag{61}$$

and

$$r = \frac{2F}{1+cos\theta_o} \tag{62}$$

θ_o was determined in Section 3.1. We can then determine that $\Delta\theta_s^m \simeq 3.4708^\circ$, this gives that $M = 10$. Figure 20 (b) shows the projected x-y plane. $\Delta\phi_s^n$ can be determined in much the same way as $\Delta\theta_s^m$. If we set $s = \lambda$ and using the equation for an arc length and solving for $\Delta\phi_s^n$, we get $\Delta\phi_s^n = 5.73^\circ$. In order to have a round number for N we select $N = 72$ which gives $\Delta\phi_s^n = 5^\circ$ and an arc length of $s = 0.873\lambda$. Now that M and N are determined we can proceed to the analysis of the three reflector configurations. As stated in section 2.2 there is a need for large amounts of memory for the storage of coefficients that are independent of the output observation angle. With this in mind it would be advantageous for this routine to be coded as a function call in MATLAB with the feed location and the pattern cut angle (ϕ_p) as input variables. Having this routine coded as a function will significantly increase the computational efficiency because all the variables of the script, including the large variables containing the coefficient information, are local to the function. The fact that these large variables are not part of the MATLAB workspace will decrease the time required to run this script.

3.3.1 Reflector #1

Figures 21 and 22 show the E and H-plane patterns respectively. Both these patterns were computed in only 7 seconds. Note that both these patterns closely match the patterns com-

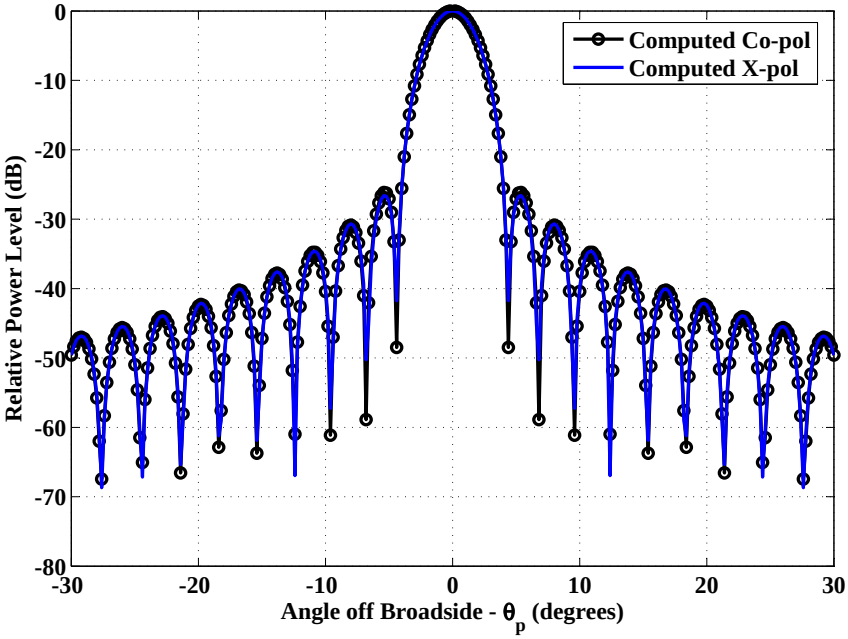

Figure 21: *Radiation Pattern for Reflector #1, $\phi_p = 0^\circ$, Ludwig Method*

puted with GRASP. We should also notice in Figure 21 that the computed cross-pol level can be seen at approximately -80dB even though we have a balanced feed and should have zero cross-pol. The appearance of the cross-pol pattern here is most likely due to numerical errors introduced by approximating the amplitude and phase terms of the integrand of the radiation integral by a first degree polynomial (i.e. plane) over the incremental surface area.

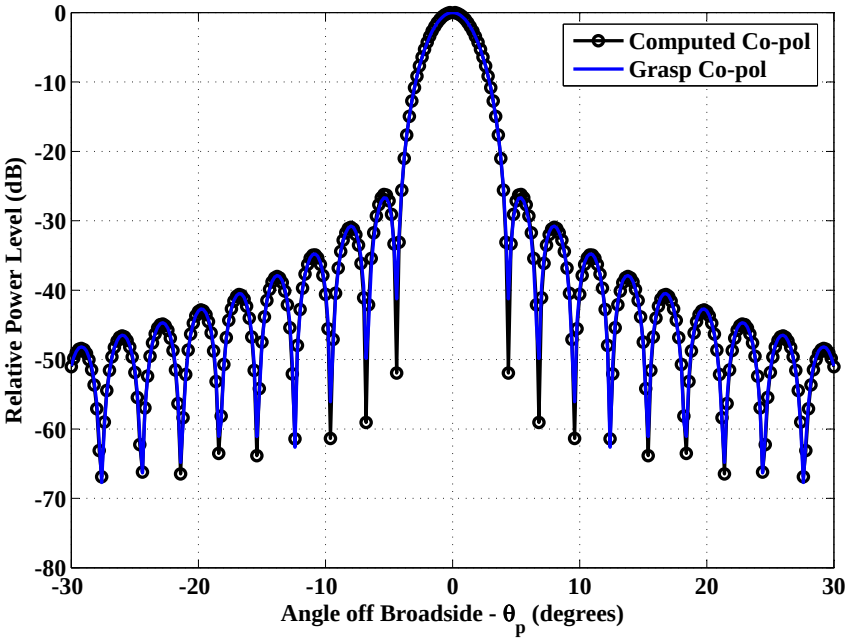

Figure 22: *Radiation Pattern for Reflector #1,* $\phi_p = 90^\circ$*, Ludwig Method*

3.3.2 Reflector #2

We will now look at how well the Ludwig method works with a displaced feed. Figures 23 and 24 show the E and H-plane patterns respectively with a displaced feed. Once again even though we are introducing a more complicated integrand, both these patterns were computed in 7 seconds. Again with the displaced feed we should see a squint in the main

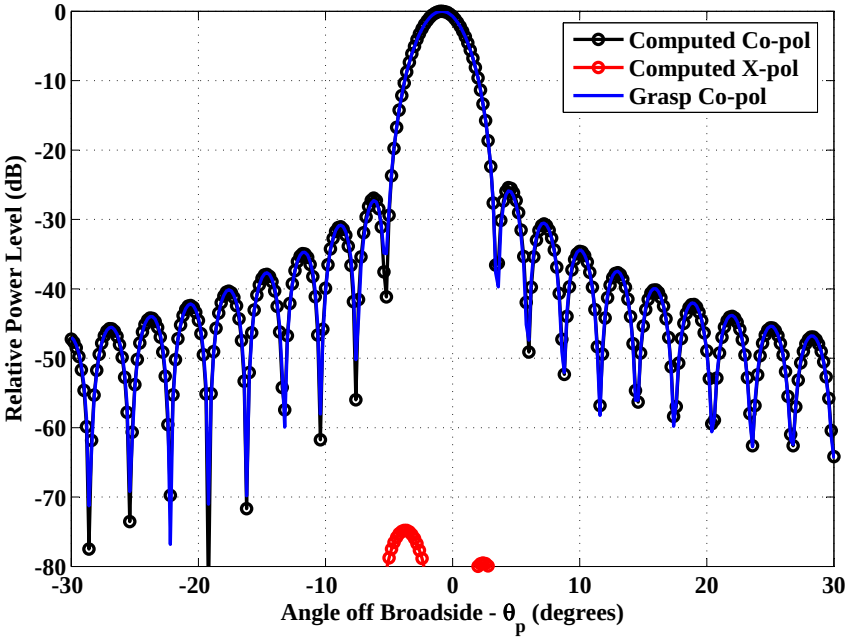

Figure 23: *Radiation Pattern for Reflector #2, $\phi_p = 0°$, Ludwig Method*

beam of $0.84°$. Figure 25 shows a zoomed in view of the squinted main beam. We can see that the Ludwig method accurately models the squint in the main beam.

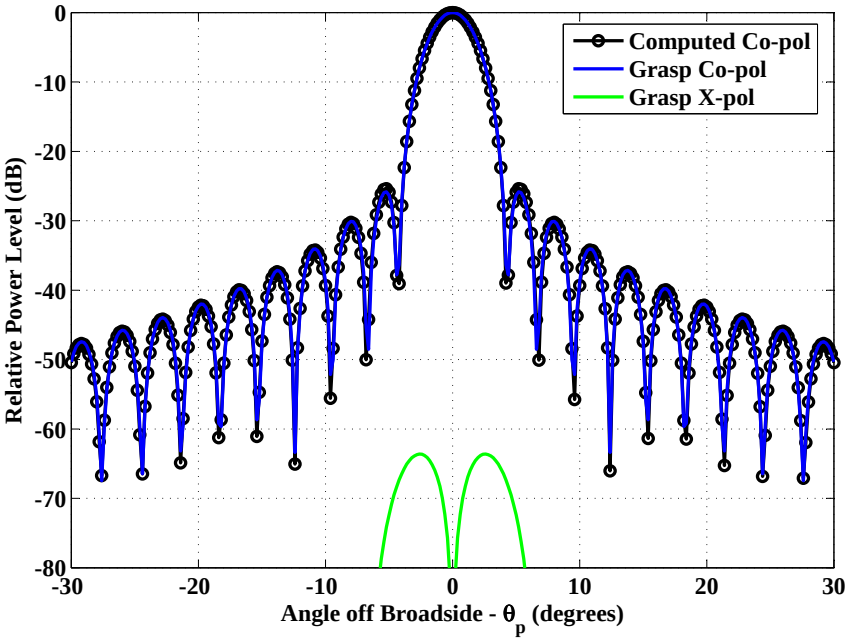

Figure 24: *Radiation Pattern for Reflector #2,* $\phi_p = 90^\circ$*, Ludwig Method*

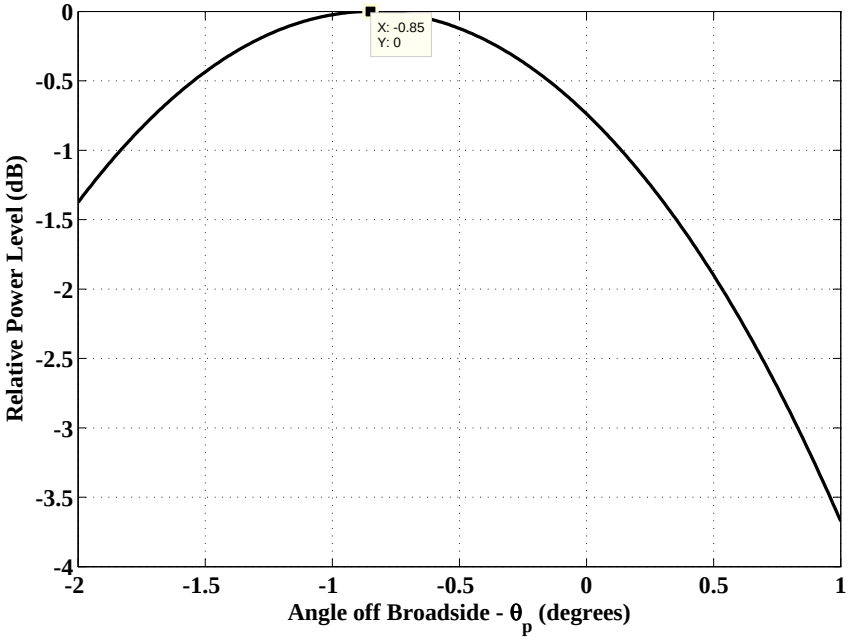

Figure 25: *Radiation Pattern Beam Squint(Reflector #2), Ludwig Method*

3.3.3 Reflector #3

We will now look at the final reflector configuration. Figures 26 and 27 show the E and H-plane patterns respectively. As it was the case for the other two configurations each of these pattern cuts were produced in only 7 seconds. Notice here that cross-pol patterns are

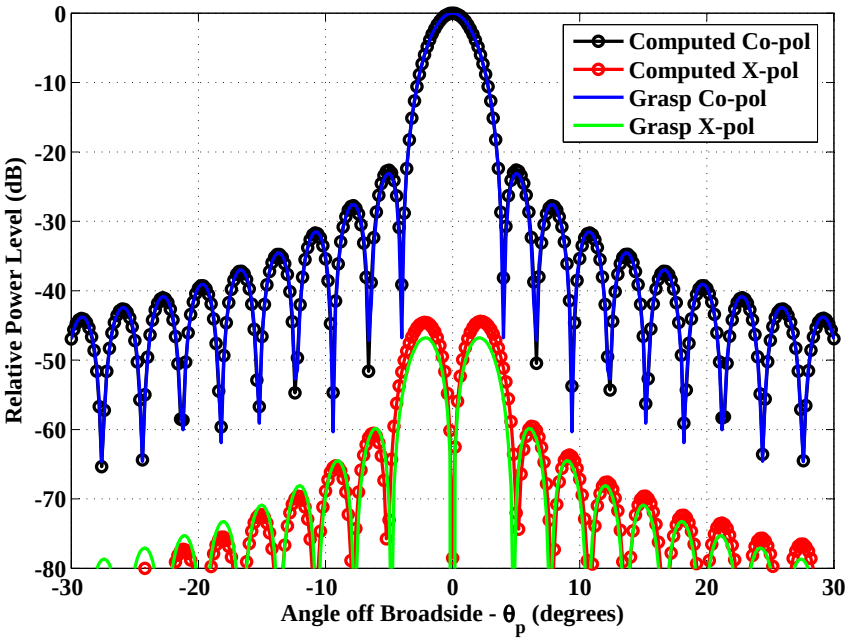

Figure 26: *Radiation Pattern for Reflector #3, $\phi_p = 0°$, Ludwig Method*

modelled fairly accurately. The numerical errors introduced because of the approximation made by the Ludwig method does not seem to be a significant issue when an actual cross-pol pattern is modelled. Again we see a main beam squint, Figure 28 shows an zoomed in view of the main beam and we see again that the Ludwig method accurately models the squint in the main beam.

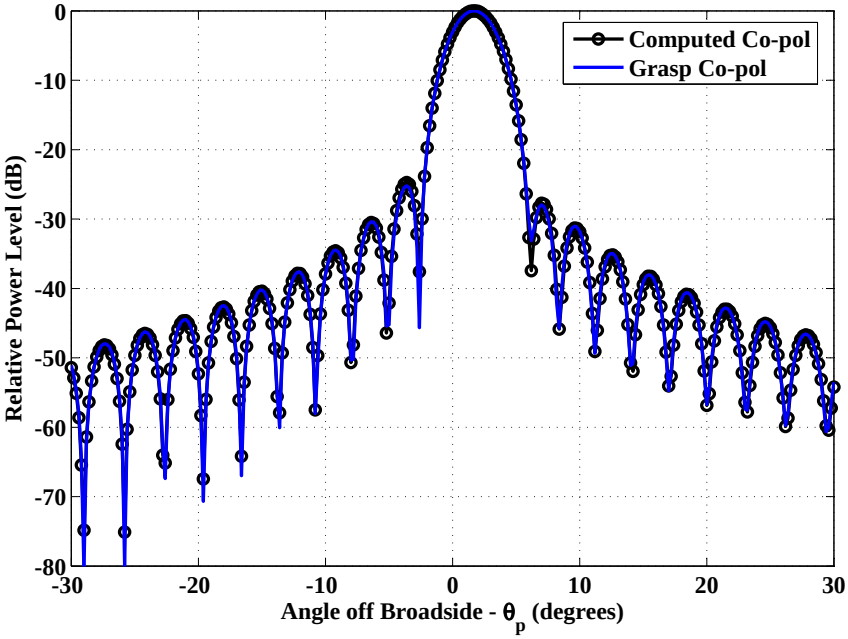

Figure 27: *Radiation Pattern for Reflector #3, $\phi_p = 90^\circ$, Ludwig Method*

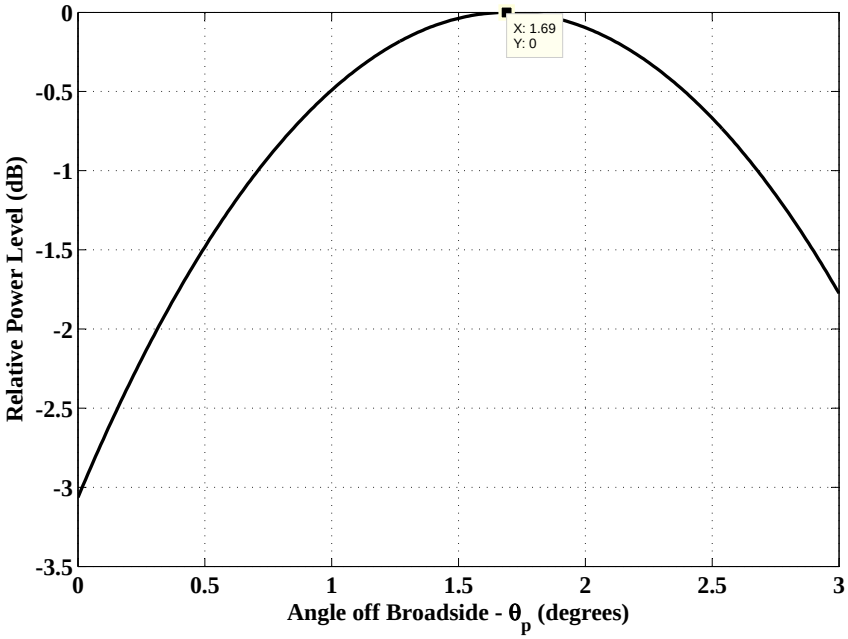

Figure 28: *Radiation Pattern Beam Squint (Reflector #3), Ludwig Method*

4 Concluding Remarks

We can see from the results presented in the previous section that both of the methods discussed (PO/Surface Current Method and Ludwig's Method) give accurate results when compared to patterns computed using commercially available reflector modelling software. Where these two methods differ is in the amount of time required to computed these results. We have seen that with the PO/SC Method, the integration of the surface currents induced on the reflector surface due to the feed is performed using a numerical integration routine built into MATLAB. We have shown that, with a laterally displaced feed, where the integrand of the radiation integral increases in complexity, the evaluation of one principal plane pattern can take up to one hour to compute. Incorporating the PO/SC method into a seeker model would therefore be impractical.

As an alternative to the PO/SC Method we looked at Ludwig's Method for the analysis of reflector patterns. We have shown that this method, regardless of the lateral displacement of the feed, can accurately compute the principal plane pattern in under 10 seconds. It was also shown that this method did introduce some numerical errors, specifically in the cross polarization pattern, due to the approximations made. Further investigation will be required to see if these numerical errors cause significant inaccuracies and if so how to eliminate or reduce these errors so that they do not cause major issues.

Future work will involve applying these modelling techniques to reflector antennas which have a compound feed for use in monopluse systems. Finally a monopulse antenna model, based on the Ludwig method, could be incorporated into a missile seeker model.

References

1. Silver, S. (1949). Microwave Antenna Theory and Design, McGraw-Hill. pp. 146-149.

2. Ludwig, A.C. (1973). The Definition of Cross Polarization. *IEEE Trans. Antenna & Prop.*, **Vol. AP-21**, pp. 116–119.

3. Ludwig, A.C. (1968). Computation of Radiation Patterns Involving Numerical Double Integration. *IEEE Trans. Antenna & Prop.*, **Vol. AP-16**, pp. 767–769.

4. GRASP - General reflector and antenna farm analysis software. Ticra. Læderstræde 34, Copenhagen, Denmark.

5. Stutzman, W.L. and Thiele, G.A. (1997). Antenna Theory and Desing, John Wiley & Sons. pp. 338-347.

6. Clarricoats, P.J.B. and Olver, A.D. (1984). Corrugated Horns for Microwave Antennas, Peter Peregrinus Ltd. pp. 101.

Annex A
MATLAB Code

```
clear clc
%************************************************************************%
%
%File name: Reflector_Rad_Pattern_PO.m
%
%This MATLAB routine will compute the radiation pattern of a parabolic
%reflector using the PO/Surface Current Method.
%
%************************************************************************%

%Global variables
global EO UO ZO C FREQ LAM W KO D F

%Start time to determine run time
S_time = clock;

%Defines constants
EO = 8.8542e-12; UO = 4*pi*1e-7; ZO = sqrt(UO/EO); C =
(sqrt(EO*UO))^-1;

%Model electrical parameters
FREQ = 30e9; LAM = C/FREQ; W = 2*pi*FREQ; KO = W*sqrt(UO*EO);

%Reflector physical parameters
D = 20*LAM; FoD = 0.8; F = FoD*D;

%Far-Field distance
R = ((2*D^2)/LAM);

%Determines the subtended angle of the parabaloid
theta_o_s = 2*atan2(D,4*F);

%Vector which defines the feed position
F_v = [0;0;0];

%Feed raised cosine factor
qe = 6.5; qh = 6.5;

%Radiation pattern cut
phi_p_deg = 0; phi_p = rad(phi_p_deg);
```

```
%Determines repeated cos and sin functions of phi_p
cpp = cos(phi_p); spp = sin(phi_p); cpp2 = cpp^2; spp2 = spp^2;

%Constant definition
const = (-j*KO*ZO*exp(-j*KO*R))/(4*pi*R);

%Index initiallization
ii = 1;

%For loop which will calculate the radiation pattern as a function of the
%angle off broadside (theta_p)
for theta_p_deg = -30:0.2:30
    theta_p = rad(theta_p_deg);

    %Determines repeated cos and sin functions of theta_p
    stp = sin(theta_p);
    ctp = cos(theta_p);
    stp2 = stp^2;
    ctp2 = ctp^2;

    %Function calls (dblquad) that will numerically evaluate the integrand
    %of the radiation integral for the 3 cartesian coordinates.
    Fx = dblquad(@Rad_int_x_rev4,0,2*pi,0,theta_o_s,1e-8,@quad,F_v,...
        qe,qh,stp,ctp,spp,cpp);
    Fy = dblquad(@Rad_int_y_rev4,0,2*pi,0,theta_o_s,1e-8,@quad,F_v,...
        qe,qh,stp,ctp,spp,cpp);
    Fz = dblquad(@Rad_int_z_rev4,0,2*pi,0,theta_o_s,1e-8,@quad,F_v,...
        qe,qh,stp,ctp,spp,cpp);

    %Computes the cartesian components of the far-zone electic field
    %(radiation pattern)
    Ex = const*(Fx - Fx*stp2*cpp2 + Fy*stp2*spp*cpp + Fz*ctp*stp*cpp);
    Ey = const*(-Fy - Fx*stp2*cpp*spp + Fy*stp2*spp2 + Fz*ctp*stp*spp);
    Ez = const*(-Fz - Fx*stp*cpp*ctp + Fy*stp*spp*ctp + Fz*ctp2);

    %Computes the co and x-pol patterns bassed on the cartessian components
    %determined previously
    E_co_p(ii) = -(1 - ctp)*spp*cpp*Ex + (1 - spp2*(1 - ctp))*Ey...
        - stp*spp*Ez;
    E_x_p(ii) = + (1 - cpp2*(1 - ctp))*Ex - (1 - ctp)*spp*cpp*Ey...
        - stp*cpp*Ez;

    %Increment index
    ii = ii + 1;
end
```

```
%Defines theta_p to plot pattern
theta = -30:0.2:30;

%Determines the co and x-pol normalized patterns in dB
m_E_co = abs(E_co_p); m_E_x = abs(E_x_p); peaks = [max(m_E_co)
max(m_E_x)]; max_val = max(peaks);

n_E_co = m_E_co/max_val; n_E_x = m_E_x/max_val;

db_E_co = 20*log10(n_E_co); db_E_x = 20*log10(n_E_x);

%Plots normalized radiation patterns
figure plot(theta,db_E_co,'-k','LineWidth',2) hold on
plot(theta,db_E_x,'-r','LineWidth',2)

%Defines figure parameters
ylim ([-80 0]) xlabel('Angle off Broadside - \theta_p
(degrees)','FontName',...
    'Times New Roman','FontSize',14,'FontWeight','bold')
ylabel('Relative Power Level (dB)','FontName',...
    'Times New Roman','FontSize',14,'FontWeight','bold')
grid on

%Routine that stops timer and converts elapsed time to h:m:s format
time(S_time)
```

```
function [E_temp] =
Rad_int_x_rev4(phi_s,theta_s,F_v,qe,qh,stp,ctp,spp,cpp)
%************************************************************************%
%
%File name: Rad_int_x_rev4.m
%
%This function will define the x-component of the integrand which will be
%numerically integrated using dblquad in Reflector_Rad_Pattern_PO.m
%
%************************************************************************%

%Global variables
global EO UO ZO C FREQ LAM W KO D F

%Defines repeated sinusoidal terms
sts = sin(theta_s); cts = cos(theta_s); sps = sin(phi_s); cps =
cos(phi_s); sts2 = sin(theta_s/2); cts2 = cos(theta_s/2); tts2 =
tan(theta_s/2);

%Defines rho vector from origin to reflector surface
r_s = (2*F)./(1 + cts); r_sx = r_s.*sts*cps; r_sy = r_s.*sts*sps;
r_sz = r_s.*cts;

%Defines angles of primed coor sys.
term1 = sqrt((r_s.*sts*cps - F_v(1)).^2 + (r_s.*sts*sps -
F_v(2)).^2); term2 = r_s.*cts; theta_s_p = atan2(term1,term2);

term3 = (r_s.*sts*sps - F_v(2)); term4 = (r_s.*sts*cps - F_v(1));
phi_s_p = atan2(term3,term4);

%Defines repeated sinusoidal terms
stsp = sin(theta_s_p); ctsp = cos(theta_s_p); spsp = sin(phi_s_p);
cpsp = cos(phi_s_p);

%Defines the magnitude of rho prime vector which extends from the feed to
%the surface of the reflector
r_s_p = sqrt(r_s.^2 + F_v(1)^2 + F_v(2)^2 ...
    - 2*r_s.*sts.*(F_v(1)*cps + F_v(2)*sps));
const = (2*exp(-j*KO*r_s_p))/(r_s_p*ZO);

%Defines the feed pattern coefficients
Ce = (ctsp).^qe; Ch = (ctsp).^qh;

%Defines the x component of the surface current induced on the reflector
%surface
```

```
Js_x = const*(-Ch.*cpsp.*stsp.*sts2.*sps...
    -Ch.*cpsp.*ctsp.*spsp.*cts2...
    +Ce.*spsp.*cpsp.*cts2);

%Defines exponential term of the integrand
temp1 = r_s.*(sts*cps*stp*cpp - sts*sps*stp*spp - cts*ctp); expo =
exp(j*KO*temp1);

%Defines differential surface term
ds = r_s.^2.*sts.*sec(theta_s/2);

%Defines the integrand to be numerically integrated
E_temp = Js_x.*expo.*ds;
```

```
function [E_temp] =
Rad_int_y_rev4(phi_s,theta_s,F_v,qe,qh,stp,ctp,spp,cpp)
%************************************************************************%
%
%File name: Rad_int_y_rev4.m
%
%This function will define the y-component of the integrand which will be
%numerically integrated using dblquad in Reflector_Rad_Pattern_PO.m
%
%************************************************************************%

%Global variables
global EO UO ZO C FREQ LAM W KO D F

%Defines repeated sinusoidal terms
sts = sin(theta_s); cts = cos(theta_s); sps = sin(phi_s); cps =
cos(phi_s); sts2 = sin(theta_s/2); cts2 = cos(theta_s/2); tts2 =
tan(theta_s/2);

%Defines rho vector from origin to reflector surface
r_s = (2*F)./(1 + cts); r_sx = r_s.*sts*cps; r_sy = r_s.*sts*sps;
r_sz = r_s.*cts;

%Defines angles of primed coor sys.
term1 = sqrt((r_s.*sts*cps - F_v(1)).^2 + (r_s.*sts*sps -
F_v(2)).^2); term2 = r_s.*cts; theta_s_p = atan2(term1,term2);

term3 = (r_s.*sts*sps - F_v(2)); term4 = (r_s.*sts*cps - F_v(1));
phi_s_p = atan2(term3,term4);

%Defines repeated sinusoidal terms
stsp = sin(theta_s_p); ctsp = cos(theta_s_p); spsp = sin(phi_s_p);
cpsp = cos(phi_s_p);

%Defines the magnitude of rho prime vector which extends from the feed to
%the surface of the reflector
r_s_p = sqrt(r_s.^2 + F_v(1)^2 + F_v(2)^2 ...
    - 2*r_s.*sts.*(F_v(1)*cps + F_v(2)*sps));
const = (2*exp(-j*KO*r_s_p))/(r_s_p*ZO);

%Defines the feed pattern coefficients
Ce = (ctsp).^qe; Ch = (ctsp).^qh;

%Defines the y component of the surface current induced on the reflector
%surface
```

```
Js_y = const*(Ch.*(cpsp).^2.*ctsp.*cts2...
    + Ce.*(spsp).^2.*cts2...
    + Ch.*cpsp.*stsp.*sts2.*cps);

%Defines exponential term of the integrand
temp1 = r_s.*(sts*cps*stp*cpp - sts*sps*stp*spp - cts*ctp); expo =
exp(j*KO*temp1);

%Defines differential surface term
ds = r_s.^2.*sts.*sec(theta_s/2);

%Defines the integrand to be numerically integrated
E_temp = Js_y.*expo.*ds;
```

```
function [E_temp] =
Rad_int_z_rev4(phi_s,theta_s,F_v,qe,qh,stp,ctp,spp,cpp)
%************************************************************************%
%
%File name: Rad_int_z_rev4.m
%
%This function will define the z-component of the integrand which will be
%numerically integrated using dblquad in Reflector_Rad_Pattern_PO.m
%
%************************************************************************%

%Global variables
global EO UO ZO C FREQ LAM W KO D F

%Defines repeated sinusoidal terms
sts = sin(theta_s); cts = cos(theta_s); sps = sin(phi_s); cps =
cos(phi_s); sts2 = sin(theta_s/2); cts2 = cos(theta_s/2); tts2 =
tan(theta_s/2);

%Defines rho vector from origin to reflector surface
r_s = (2*F)./(1 + cts); r_sx = r_s.*sts*cps; r_sy = r_s.*sts*sps;
r_sz = r_s.*cts;

%Defines angles of primed coor sys.
term1 = sqrt((r_s.*sts*cps - F_v(1)).^2 + (r_s.*sts*sps -
F_v(2)).^2); term2 = r_s.*cts; theta_s_p = atan2(term1,term2);

term3 = (r_s.*sts*sps - F_v(2)); term4 = (r_s.*sts*cps - F_v(1));
phi_s_p = atan2(term3,term4);

%Defines repeated sinusoidal terms
stsp = sin(theta_s_p); ctsp = cos(theta_s_p); spsp = sin(phi_s_p);
cpsp = cos(phi_s_p);

%Defines the magnitude of rho prime vector which extends from the feed to
%the surface of the reflector
r_s_p = sqrt(r_s.^2 + F_v(1)^2 + F_v(2)^2 ...
    - 2*r_s.*sts.*(F_v(1)*cps + F_v(2)*sps));
const = (2*exp(-j*KO*r_s_p))/(r_s_p*ZO);

%Defines the feed pattern coefficients
Ce = (ctsp).^qe; Ch = (ctsp).^qh;

%Defines the z component of the surface current induced on the reflector
%surface
```

```
Js_z = const*(Ch.*cpsp.*ctsp.*spsp.*sts2.*cps...
    -Ce.*spsp.*cpsp.*sts2.*cps...
    -Ch.*(cpsp).^2.*ctsp.*sts2.*sps...
    -Ce.*(spsp).^2.*sts2.*sps);

%Defines exponential term of the integrand
temp1 = r_s.*(sts*cps*stp*cpp - sts*sps*stp*spp - cts*ctp); expo =
exp(j*KO*temp1);

%Defines differential surface term
ds = r_s.^2.*sts.*sec(theta_s/2);

%Defines the integrand to be numerically integrated
E_temp = Js_z.*expo.*ds;
```

```
function [E_co_p,E_x_p] =
Reflector_Rad_Pattern_Ludwig_FC(F_v,phi_p_deg)
%************************************************************************%
%
%File name: Reflector_Rad_Pattern_Ludwig_FC.m
%
%This MATLAB function will compute the radiation pattern of a parabolic
%reflector using the Ludwig Method.
%
%************************************************************************%
global EO UO ZO C FREQ LAM W KO D F

%Far-Field
R = (2*D^2)/LAM;

%Determines the subtended angle of the parabaloid
theta_o_s = 2*atan2(D,4*F);

%Feed pattern raised cosine coefficients
qe = 6.5; qh = 6.5;

%Defines the grid on the reflecotr surface which defines the incremental
%surface area
theta_seg = 11; del_theta = theta_o_s/(theta_seg-1); theta_t = 0;
for mm = 1:1:theta_seg
    theta_s_i(mm) = theta_t;
    theta_t = theta_s_i(mm) + del_theta;
end

phi_seg = 73; del_phi = 2*pi/(phi_seg-1); phi_t = 0; for nn =
1:1:phi_seg
    phi_s_i(nn) = phi_t;
    phi_t = phi_s_i(nn) + del_phi;
end

%Computes repeated sinusoidal terms
stsi = sin(theta_s_i); ctsi = cos(theta_s_i); spsi = sin(phi_s_i);
cpsi = cos(phi_s_i);

%Computes length from the origine to reflector surface
r_s = (2*F)./(1 + cos(theta_s_i));

%Radiation pattern cut in radians
phi_p = rad(phi_p_deg);
```

```
%Computes repeated sinusoidal terms
spp = sin(phi_p); cpp = cos(phi_p); cpp2 = cpp^2; spp2 = spp^2;

%Preallocation of storage vectors
r_s_p1 = zeros((theta_seg-1)*(phi_seg-1),1); Fxmn =
zeros((theta_seg-1)*(phi_seg-1),1); Fymn =
zeros((theta_seg-1)*(phi_seg-1),1); Fzmn =
zeros((theta_seg-1)*(phi_seg-1),1); temp_mn1 =
zeros((theta_seg-1)*(phi_seg-1),1); temp_mn2 =
zeros((theta_seg-1)*(phi_seg-1),1); temp_mn3 =
zeros((theta_seg-1)*(phi_seg-1),1);

r_s_p2 = zeros((theta_seg-1)*(phi_seg-1),1); Fxmpn =
zeros((theta_seg-1)*(phi_seg-1),1); Fympn =
zeros((theta_seg-1)*(phi_seg-1),1); Fzmpn =
zeros((theta_seg-1)*(phi_seg-1),1); temp_mpn1 =
zeros((theta_seg-1)*(phi_seg-1),1); temp_mpn2 =
zeros((theta_seg-1)*(phi_seg-1),1); temp_mpn3 =
zeros((theta_seg-1)*(phi_seg-1),1);

r_s_p3 = zeros((theta_seg-1)*(phi_seg-1),1); Fxmnp =
zeros((theta_seg-1)*(phi_seg-1),1); Fymnp =
zeros((theta_seg-1)*(phi_seg-1),1); Fzmnp =
zeros((theta_seg-1)*(phi_seg-1),1); temp_mnp1 =
zeros((theta_seg-1)*(phi_seg-1),1); temp_mnp2 =
zeros((theta_seg-1)*(phi_seg-1),1); temp_mnp3 =
zeros((theta_seg-1)*(phi_seg-1),1);

r_s_p4 = zeros((theta_seg-1)*(phi_seg-1),1); Fxmpnp =
zeros((theta_seg-1)*(phi_seg-1),1); Fympnp =
zeros((theta_seg-1)*(phi_seg-1),1); Fzmpnp =
zeros((theta_seg-1)*(phi_seg-1),1); temp_mpnp1 =
zeros((theta_seg-1)*(phi_seg-1),1); temp_mpnp2 =
zeros((theta_seg-1)*(phi_seg-1),1); temp_mpnp3 =
zeros((theta_seg-1)*(phi_seg-1),1);

ax_mn = zeros((theta_seg-1)*(phi_seg-1),1); bx_mn =
zeros((theta_seg-1)*(phi_seg-1),1); cx_mn =
zeros((theta_seg-1)*(phi_seg-1),1); ay_mn =
zeros((theta_seg-1)*(phi_seg-1),1); by_mn =
zeros((theta_seg-1)*(phi_seg-1),1); cy_mn =
zeros((theta_seg-1)*(phi_seg-1),1); az_mn =
zeros((theta_seg-1)*(phi_seg-1),1); bz_mn =
zeros((theta_seg-1)*(phi_seg-1),1); cz_mn =
zeros((theta_seg-1)*(phi_seg-1),1);
```

```
%Index initiallization
jj = 1;

%This routine will calculate all the terms that are independant on the
%output pattern angle and stores them in vectors.  Namely it will compute
%the amplitude term at all the corners of the incremental surface area for
%the three cartesian components it also computes the coefficients of the
%linear function which approximates the amplitude over the incremental
%surface area.
for mm = 1:1:theta_seg-1
    for nn = 1:1:phi_seg-1
        [r_s_p1(jj),Fxmn(jj),Fymn(jj),Fzmn(jj),temp_mn1(jj),...
            temp_mn2(jj),temp_mn3(jj)] = Ludwig_terms(mm,nn,theta_s_i,...
            stsi,ctsi,spsi,cpsi,F_v,r_s,qe,qh);
        [r_s_p2(jj),Fxmpn(jj),Fympn(jj),Fzmpn(jj),temp_mpn1(jj),...
            temp_mpn2(jj),temp_mpn3(jj)] = Ludwig_terms((mm+1),nn,...
            theta_s_i,stsi,ctsi,spsi,cpsi,F_v,r_s,qe,qh);
        [r_s_p3(jj),Fxmnp(jj),Fymnp(jj),Fzmnp(jj),temp_mnp1(jj),...
            temp_mnp2(jj),temp_mnp3(jj)] = Ludwig_terms(mm,(nn+1),...
            theta_s_i,stsi,ctsi,spsi,cpsi,F_v,r_s,qe,qh);
        [r_s_p4(jj),Fxmpnp(jj),Fympnp(jj),Fzmpnp(jj),temp_mpnp1(jj),...
            temp_mpnp2(jj),temp_mpnp3(jj)] = Ludwig_terms((mm+1),...
            (nn+1),theta_s_i,stsi,ctsi,spsi,cpsi,F_v,r_s,qe,qh);

        ax_mn(jj) = (1/4)*(3*Fxmn(jj) - Fxmpnp(jj)...
            + Fxmpn(jj) + Fxmnp(jj));
        bx_mn(jj) = (1/(2*del_theta))*(Fxmpn(jj)...
            - Fxmn(jj) + Fxmpnp(jj) - Fxmnp(jj));
        cx_mn(jj) = (1/(2*del_phi))*(Fxmnp(jj)...
            - Fxmn(jj) + Fxmpnp(jj) - Fxmpn(jj));

        ay_mn(jj) = (1/4)*(3*Fymn(jj) - Fympnp(jj)...
            + Fympn(jj) + Fymnp(jj));
        by_mn(jj) = (1/(2*del_theta))*(Fympn(jj)...
            - Fymn(jj) + Fympnp(jj) - Fymnp(jj));
        cy_mn(jj) = (1/(2*del_phi))*(Fymnp(jj)...
            - Fymn(jj) + Fympnp(jj) - Fympn(jj));

        az_mn(jj) = (1/4)*(3*Fzmn(jj) - Fzmpnp(jj)...
            + Fzmpn(jj) + Fzmnp(jj));
        bz_mn(jj) = (1/(2*del_theta))*(Fzmpn(jj)...
            - Fzmn(jj) + Fzmpnp(jj) - Fzmnp(jj));
        cz_mn(jj) = (1/(2*del_phi))*(Fzmnp(jj)...
            - Fzmn(jj) + Fzmpnp(jj) - Fzmpn(jj));
```

```
        jj = jj +1;
    end
end

%Defines the angles off broadside for which the radiation pattern is
%computed
theta = -30:0.2:30;

%Preallocation of storage vectors
L = length(theta); E_theta_p = zeros(L,1); E_phi_p = zeros(L,1);
E_co_p = zeros(L,1); E_x_p = zeros(L,1);

%Constant definition
const = (-j*KO*ZO*exp(-j*KO*R))/(4*pi*R);

%Index initiallization
ii = 1;

%For loop which will calculate the radiation pattern as a function of the
%angle off broadside (theta_p)
for theta_p_deg = -30:0.2:30
    theta_p = rad(theta_p_deg);

    %Computes repeated sinusoidal terms
    stp = sin(theta_p);
    ctp = cos(theta_p);
    stp2 = stp^2;
    ctp2 = ctp^2;

    %Variable initialisation
    Fx = 0;
    Fy = 0;
    Fz = 0;

    %Index initiallization
    jj = 1;

    %This routine computes the phase term of the integrand on the
    %integration grid which defines the incremental surface area it also
    %computes the coefficients linear function approximating the phase
    %onver the incremental surface area.  Finally this rouitne also
    %computes the intgral (closed form) over the incremental surface area
    %and sums all these contributions for the 3 cartesian components.
    for mm = 1:1:theta_seg-1
```

```
for nn = 1:1:phi_seg-1

    gmn = temp_mn1(jj)*stp*cpp...
        - temp_mn2(jj)*stp*spp...
        - temp_mn3(jj)*ctp - r_s_p1(jj);

    gmpn = temp_mpn1(jj)*stp*cpp...
        - temp_mpn2(jj)*stp*spp...
        - temp_mpn3(jj)*ctp - r_s_p2(jj);

    gmnp = temp_mnp1(jj)*stp*cpp...
        - temp_mnp2(jj)*stp*spp...
        - temp_mnp3(jj)*ctp - r_s_p3(jj);

    gmpnp = temp_mpnp1(jj)*stp*cpp...
        - temp_mpnp2(jj)*stp*spp...
        - temp_mpnp3(jj)*ctp - r_s_p4(jj);

    alpha_mn = (1/4)*(3*gmn - gmpnp + gmpn + gmnp);
    beta_mn = (1/(2*del_theta))*(gmpn - gmn + gmpnp - gmnp);
    xi_mn = (1/(2*del_phi))*(gmnp - gmn + gmpnp - gmpn);

    term1 = exp(j*KO*alpha_mn);
    if beta_mn == 0 || xi_mn == 0
        term2 = 0;
        term3 = 0;
        term4 = 0;
    else
        term2 = ((exp(j*KO*beta_mn*del_theta)-1)/(j*KO*beta_mn))...
            *((exp(j*KO*xi_mn*del_theta)-1)/(j*KO*xi_mn));
        term3 = (((del_theta)/(j*KO*beta_mn))...
            *exp(j*KO*beta_mn*del_theta)...
            - ((exp(j*KO*beta_mn*del_theta)-1)/(j*KO*beta_mn)^2))...
            *((exp(j*KO*xi_mn*del_theta)-1)/(j*KO*xi_mn));
        term4 = ((exp(j*KO*beta_mn*del_theta)-1)/(j*KO*beta_mn))...
            *(((del_phi)/(j*KO*xi_mn))*exp(j*KO*xi_mn*del_phi)...
            - ((exp(j*KO*xi_mn*del_phi)-1)/(j*KO*xi_mn)^2));
    end

    del_Fx = term1*(ax_mn(jj)*term2 + bx_mn(jj)*term3...
        + cx_mn(jj)*term4);
    del_Fy = term1*(ay_mn(jj)*term2 + by_mn(jj)*term3...
        + cy_mn(jj)*term4);
    del_Fz = term1*(az_mn(jj)*term2 + bz_mn(jj)*term3...
        + cz_mn(jj)*term4);
```

```
            Fx = Fx + del_Fx;
            Fy = Fy + del_Fy;
            Fz = Fz + del_Fz;

            jj = jj + 1;

        end
    end

    %Computes the cartesian components of the far-zone electic field
    %(radiation pattern)
    Ex = const*(Fx - Fx*stp2*cpp2 + Fy*stp2*spp*cpp + Fz*ctp*stp*cpp);
    Ey = const*(-Fy - Fx*stp2*cpp*spp + Fy*stp2*spp2 + Fz*ctp*stp*spp);
    Ez = const*(-Fz - Fx*stp*cpp*ctp + Fy*stp*spp*ctp + Fz*ctp2);

    %Computes the co and x-pol patterns bassed on the cartessian components
    %determined previously
    E_co_p(ii) = -(1 - ctp)*spp*cpp*Ex + (1 - spp2*(1 - ctp))*Ey...
        - stp*spp*Ez;
    E_x_p(ii) = + (1 - cpp2*(1 - ctp))*Ex - (1 - ctp)*spp*cpp*Ey...
        - stp*cpp*Ez;

    %Increments index
    ii = ii + 1;
end
```

```
function [r_s_p,Fx,Fy,Fz,temp_1,temp_2,temp_3]...
    = Ludwig_terms(mm,nn,theta_s_i,stsi,ctsi,spsi,cpsi,F_v,r_s,qe,qh)
%************************************************************************%
%
%File name: Ludwig_terms.m
%
%This MATLAB routine will compute several terms used in the Ludwig Method,
%specifically the terms that independant on the output pattern angle
%direction. This is computed for differenc m and n indices.
%
%************************************************************************%

%Global variables
global EO UO ZO C FREQ LAM W KO D F

%Defines angles of primed coor sys. (Feed coor sys.)
term1 = sqrt((r_s(mm)*stsi(mm)*cpsi(nn) - F_v(1))^2 ...
    + (r_s(mm)*stsi(mm)*spsi(nn) - F_v(2))^2);
term2 = r_s(mm)*ctsi(mm); theta_s_p_i = atan2(term1,term2);

term3 = (r_s(mm)*stsi(mm)*spsi(nn) - F_v(2)); term4 =
(r_s(mm)*stsi(mm)*cpsi(nn) - F_v(1)); phi_s_p_i =
atan2(term3,term4);

%Defines repeated sinusoidal terms
stspi = sin(theta_s_p_i); ctspi = cos(theta_s_p_i); spspi =
sin(phi_s_p_i); cpspi = cos(phi_s_p_i);

%Defines the magnitude of rho prime vector which extends from the feed to
%the surface of the reflector
r_s_p = sqrt(r_s(mm)^2 + F_v(1)^2 + F_v(2)^2 ...
    - 2*r_s(mm)*stsi(mm)*(F_v(1)*cpsi(nn) + F_v(2)*spsi(nn)));

%Computes the amplitude term of the integrand for the 3 cartesian coord.
Fx = r_s(mm)^2*sec(theta_s_i(mm)/2)*stsi(mm)...
    *(-ctspi^qh*sin(theta_s_i(mm)/2)*spsi(nn)*stspi*cpspi...
    -ctspi^qh*cos(theta_s_i(mm)/2)*ctspi*cpspi*spspi...
    +ctspi^qe*cos(theta_s_i(mm)/2)*cpspi*spspi)*(2/(ZO*r_s_p));

Fy = r_s(mm)^2*sec(theta_s_i(mm)/2)*stsi(mm)...
    *(ctspi^qh*cpspi^2*ctspi*cos(theta_s_i(mm)/2)...
    + ctspi^qe*spspi^2*cos(theta_s_i(mm)/2)...
    + ctspi^qh*cpspi*stspi*sin(theta_s_i(mm)/2)*cpsi(nn))*(2/(ZO*r_s_p));

Fz = r_s(mm)^2*sec(theta_s_i(mm)/2)*stsi(mm)...
```

```
    *(ctspi^qh*cpspi*ctspi*spspi*sin(theta_s_i(mm)/2)*cpsi(nn)...
    -ctspi^qe*spspi*cpspi*sin(theta_s_i(mm)/2)*cpsi(nn)...
    -ctspi^qh*cpspi^2.*ctspi*sin(theta_s_i(mm)/2)*spsi(nn)...
    -ctspi^qe*spspi^2.*sin(theta_s_i(mm)/2)*spsi(nn))*(2/(ZO*r_s_p));

%Computes the cartesian components of the point on the reflector where the
%amplitude and phase terms of the integrands will be calculated (i.e.
%corners of the incremental surface area.
temp_1 = r_s(mm)*stsi(mm)*cpsi(nn); temp_2 =
r_s(mm)*stsi(mm)*spsi(nn); temp_3 = r_s(mm)*ctsi(mm);
```

DOCUMENT CONTROL DATA

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

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence R&D Canada – Ottawa
3701 Carling Ave., Ottawa, ON, Canada K1A 0Z4

2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable).

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

Parabolic Reflector Modelling Techniques with a Laterally Displaced Feed

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Arpin, Frédéric

5. DATE OF PUBLICATION (month and year of publication of document)

July 2005

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc).

60

6b. NO. OF REFS (total cited in document)

6

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

Technical Memorandum

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include address).

Defence R&D Canada – Ottawa
3701 Carling Ave., Ottawa, ON, Canada K1A 0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Specify whether project or grant).

11at14

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written).

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique.)

DRDC Ottawa TM 2005-098

10b. OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Defence departments and defence contractors; further distribution only as approved
( ) Defence departments and Canadian defence contractors; further distribution only as approved
( ) Government departments and agencies; further distribution only as approved
( ) Defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution beyond the audience specified in (11) is possible, a wider announcement audience may be selected).

UNLIMITED

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

This document looks at two techniques for modelling the radiation pattern of a parabolic reflector antenna. Both these methods use physical optics to determine the induced currents on the reflector due to the feed. The first method uses MATLAB's numerical integration routine to compute the far-field radiation pattern. The second method involves segmenting the reflector surface and approximating the amplitude and phase of the surface currents by a first degree polynomial. This approximation allows the integration to be performed analytically in closed-form. Three different reflector configurations are investigated including two configurations with a laterally displaced feed. Both these techniques yield accurate results but the latter technique had a significant improvement in computational efficiency.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

Parabolic Reflector, Antenna Modelling, Radiation Patterns, Physical Optics

Defence R&D Canada

Canada's leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

www.drdc-rddc.gc.ca