User Guide for Q-par PO reflector analysis software (QPORAS)

Q-par/QPORAS/TR-PO1/1.1

Q-par Angus Ltd

IDEAS ENGINEERED

Technical Report

Cover + ix + 98 pages

April 2014

Barons Cross Laboratories,
Leominster, Herefordshire, HR6 8RS, UK.
Tel: +44 (0) 1568 612138 Fax : +44 (0) 1568 616373
Web: www.q-par.com E-mail : sales@q-par.com

© Copyright 2014
Q-par Angus Ltd., U.K.

Although the author has made every reasonable effort to ensure the accuracy of the contents of this document, neither the author nor Q-par Angus Ltd. will be held liable for damages resulting from the use or application of the material contained, to the extent permitted by international law.

This document may be distributed freely subject to the following conditions:

The document may not be changed or modified or the contents altered. It must remain complete without selective removal of material. It may not be sold or reproduced for commercial gain, without the written approval of an authorised representative of Q-par Angus Ltd. (UK).

Q-par Angus Ltd.
Barons Cross Lodge, Leominster,
Herefordshire, HR6 8RS, UK.

Author Dr A. J. Mackay

Date April 2014

Issued by Q-par Angus Ltd.
Barons Cross Laboratories
Leominster
Herefordshire HR6 HRS
UK.

Document changes record

Issue	Date	Change summary
Issue 0.1	September 2008	In progress.
Issue 0.2	May 2010	Various extensions. Include *ANGLECUT* key word and theory.
Issue 0.3	November 2012	Include *GAUSSIAN* key word and theory.
Issue 0.4	December 2012	Include various additional shaped reflector forms.
Issue 0.5	December 2012	Replace GAUSSIAN $\rightarrow$ MGAUSSIAN and introduce the standard PGAUSSIAN paraxial form. Modify MGAUSSIAN formulation.
Issue 0.6	January 2013	Include validation examples and use of CST macros for CST comparison predictions.
Issue 0.7	January 2013	Extra parameter required for offset shaped reflectors. Described.
Issue 0.8	February 2013	Extra key word ROTSURF for rigid rotation of main reflector. Extra key word DISTORTSURF with two distortion types for shaped reflectors. More clarity in definition of rotation matrix $\underline{\underline{R}}$.
Issue 0.9	February 2013	Extra key word VARIPAR2D2 and other minor changes.
Issue 1.0	February 2013	Modification to VARIPAR2D2 to reflect change in software.
Issue 1.1	April 2014	Minor correction to description of syntax for flat plate reflector.

Abstract

This guide provides a technical overview and description of the use of the Q-par Physical Optics Reflector Analysis Software (QPORAS). The software uses a meshed representation of a general reflector under illumination by a plane wave or feed to provide gain and radiation patterns. It is based on the use of Physical Optics applied to a triangulated regular meshed reflector of arbitrary shape.

List of contents

Document changes record **iv**

Abstract **v**

List of contents **vi**

List of figures **viii**

1 Introduction **1**
1.1 Background 1
1.2 Overview of the physical model 1

2 The mathematical formulation **3**
2.1 The scattered far field 3
2.2 Scattered field directions and definitions 5
2.3 The Physical Optics integration 8
2.4 The incident fields 11
2.5 The scattered near field 19
2.6 Blockage effects 20

3 Use of the software **21**
3.1 Introduction 21
3.2 The *CALCOPTS* key word 23
3.3 The *FILENAME* key word 24
3.4 The *PLOTFILE* key word 25
3.5 The *ROTSURF* key word 26
3.6 The *ANGLES* key word 27
3.7 The *ANGLECUT* key word 28
3.8 The *FREQS* key word 28
3.9 The *FEEDCEN* key word 29
3.10 The *FEEDROT* key word 29
3.11 The *PLANEWAVE* key word 30
3.12 The *DIPOLE* key word 31
3.13 The *MGAUSSIAN* key word 31
3.14 The *PGAUSSIAN* key word 32
3.15 The *RECTHORN* key word 32
3.16 The *GEOMFILE* key word 34

3.17 The *SURFACE* key word 36
3.18 The *BOUNDARY* key word 57
3.19 The *DISTORTSURF* key word 59
3.20 The *BLOCKAGE* key word 63
3.21 The *NEARFIELD* key word 64
3.22 The *GRASPOUT* key word 66
3.23 The *FILEREFL* key word 66
3.24 The *FEEDFILE* key word 67
3.25 The *FARPOL* key word 69

4 Examples and CST comparison 70
4.1 Introduction 70
4.2 CST Macro "Shaped_Reflector_A" 70
4.3 CST Macro "Far field Gaussian beam source" 71
4.4 Use of an optimiser 73
4.5 Example 1. Reflection of Gaussian beams from a flat reflector 73
4.6 Example 2. A large shaped reflector 76
4.7 Example 3. A smaller shaped reflector with a real feed 79
4.8 Example 4. Use of an optimiser 92

5 References 98

List of figures

2-1 Definition of surface normal 4
2-2 Arbitrary cuts on the surface of a sphere, specified by *ANGLECUT* 6
2-3 Angle and coordinate definitions (primed feed coordinates) in the modified Gaussian beam 16
3-1 Main reflector global coordinate system 21
3-2 The feed coordinate system and Euler angles 22
3-3 Aperture excitation types for a rectangular horn 33
3-4 An ideal cosec-squared distribution and a far-field match using a shaping function. 40
3-5 A profile plot of the reflector shape and Gaussian beam origin and angle lines 40
3-6 An ideal cosec-squared distribution and a far-field match using a shaping function. 43
3-7 A near-rectangular azimuth distribution using a shaping function. 43
3-8 A profile plot of the reflector shape and Gaussian beam origin and angle lines 44
3-9 A second profile plot of the reflector shape and Gaussian beam origin and angle lines 44
3-10 Hyperbolas of constant Θ 47
3-11 Profile plots of an offset shaped reflector as defined (top to bottom) by 'sideview1', 'sideview2' and 'sideview3' 48
3-12 A target and predicted gain pattern in azimuth. 49
3-13 A target and predicted gain pattern in elevation. 49
3-14 Boundary rectangle and its projection, shown for the case $s_x = X$ and $s_z = Z$. 58
3-15 Shaping requirements for an offset shaped reflector with towards-feed directed power 59
3-16 Near field region, shadow masking and far field 64
4-1 Standard paraxial Gaussian beam reflection (absolute gain in dB), x-polarised, azimuth cut. 75
4-2 Modified non-paraxial Gaussian beam reflection (absolute gain in dB), x-polarised, azimuth cut. 75
4-3 Screen dump of CST geometry, example 2. 77
4-4 Screen dump of CST predicted 3D field magnitude 77
4-5 Scattered field comparison (absolute gain in dB), x-polarised, principal azimuth $(x - y)$ plane 78
4-6 Scattered field comparison (absolute gain in dB), x-polarised, principal elevation $(y - z)$ plane 78
4-7 CST screen dump of the feed in profile, local (feed) coordinate system. 81
4-8 CST screen dump of the feed in cross section, local (feed) coordinate system. 81

4-9 CST view of 3D far-field at 23 GHz in local (feed) coordinate system. 82
4-10 CST view of 3D far-field at 29 GHz in local (feed) coordinate system. 82
4-11 CST predicted feed gain in principal planes at 23 GHz, local (feed) coordinate system. 83
4-12 CST predicted feed gain in principal planes at 25 GHz, local (feed) coordinate system. 83
4-13 CST predicted feed gain in principal planes at 27 GHz, local (feed) coordinate system. 84
4-14 CST predicted feed gain in principal planes at 29 GHz, local (feed) coordinate system. 84
4-15 Screen dump of CST geometry, example 3. 86
4-16 Absolute gain comparison, 23 GHz, vertically polarised feed, principal horizontal plane 87
4-17 Absolute gain comparison, 23 GHz, vertically polarised feed, principal vertical plane 87
4-18 Absolute gain comparison, 23 GHz, horizontally polarised feed, principal horizontal plane 88
4-19 Absolute gain comparison, 23 GHz, horizontally polarised feed, principal vertical plane 88
4-20 Absolute gain comparison, 29 GHz, vertically polarised feed, principal horizontal plane 89
4-21 Absolute gain comparison, 29 GHz, vertically polarised feed, principal vertical plane 89
4-22 Absolute gain comparison, 29 GHz, horizontally polarised feed, principal horizontal plane 90
4-23 Absolute gain comparison, 29 GHz, horizontally polarised feed, principal vertical plane 90
4-24 Absolute gain comparison PO using real and Gaussian beam feeds, 23 GHz, vertically polarised feed, principal horizontal plane 91
4-25 Absolute gain comparison PO using real and Gaussian beam feeds, 23 GHz, vertically polarised feed, principal vertical plane 91
4-26 Cross section of the shaped reflector, horizontal cut. 96
4-27 Cross section of the shaped reflector, vertical cut. 96
4-28 Predicted PO gain in dBi based on horizontal plane optimisation at 25 GHz. Vertically polarised feed, horizontal plane 97
4-29 Predicted PO gain in dBi based on horizontal plane optimisation at 25 GHz. Vertically polarised feed, vertical plane 97

1 Introduction

1.1 Background

Although there are many powerful and efficient software packages available for the accurate analysis of reflector antennas, these tend to be expensive commercial packages which are not readily affordable. For example a full version of GRASP [1], often cited as an industry standard, costs several tens of thousands of pounds. Unfortunately, free software is generally not available to analyse general shaped reflectors with standard feeds.

This report documents our own in-house software (which, in this report, we will call QPO-RAS) intended for this purpose. This is not intended to compete with GRASP or similar general purpose analysis software, but it is adaptable and can be added and modified as and when desired. It is written in FORTRAN for use on a Linux platform, but may be ported to other systems since no system-dependent calls are implemented.

1.2 Overview of the physical model

A general shaped curved reflecting surface is assumed topologically open with no part shadowing any other part under a point illuminator or incident plane wave. The surface can be described in several different ways either accepting a previously meshed structure or meshing the structure itself. The mesh is regular and composed of triangular facets.

Unlike moment method or finite element analysis methods there are no hard restrictions on how large each facet may be compared to a wavelength. The mesh size is primarily governed by geometrical requirements. For example, a flat rectangular plate reflector could be modelled by only two triangular facets with no loss of accuracy, independent of the wavelength. However, any representation of a curved surface by a faceted (polygonal) surface involves a difference in its electromagnetic properties so in such cases the maximum departure of a flat triangle from the curved surface should be small compared to a wavelength. It is still therefore a rule for a general curved surface that a triangle mesh cell should be small compared to a wavelength.

The main faceted reflector is assumed to be perfectly conducting (as of version 0.0.012) modelled using the version of physical optics that assumes that the induced electric currents at a point $\underline{r}'$ on the surface, $\mathcal{S}$, $\underline{J}_e(\underline{r}')$ is given by the approximation,

$$\underline{J}_e(\underline{r}') = 2\hat{\underline{n}}(\underline{r}') \times \underline{H}_{inc}(\underline{r}') \text{ for } \underline{r}' \in \mathcal{S} \tag{1-1}$$

where $\hat{\underline{n}}(\underline{r}')$ is an outward pointing surface normal and $\underline{H}_{inc}(\underline{r}')$ is the incident magnetic field.

This physical optics (PO) assumption is an approximation which is generally considered accurate if, These requirements are not necessarily independent and may sometimes be relaxed under special conditions. Here they should be considered as rules of thumb. It is,

1. The radius of curvature of the surface is everywhere large compared to a wavelength except possibly over a region which is small compared to the rest of the structure.
2. Every point of the surface is under direct illumination of the source without self-shadowing of one part of the structure by another.
3. The incident field is locally planar.
4. The angle of incidence made between a locally planar wave and the surface normal should not be close to grazing incidence except possibly over a region which is small compared to the rest of the structure.
5. The surface should be much larger than a wavelength.

in fact, extremely difficult to obtain precise conditions that determine the accuracy of the physical optics method and under what conditions. To our knowledge, this remains an open question in mathematical physics.

In the discussion to date, *shadowing* refers to a purely geometrical effect. This is properly defined in the high frequency limit in terms of ray tracing. If a source is represented by a wavefront which is modelled by a collection of ray bundles, then if a given ray intersects the surface more than once the surface at the second intersection point is said to be in the shadow of the surface at the first intersection point. Shadowing, within physical optics, can only be modelled by making secondary approximations. Generally (though not always) this is accomplished by assuming an incident field that is defined by the value of the field within a ray bundle at the point of intersection with the surface. Thus, if a point of the surface is in shadow, the incident field is there assumed to be zero. In this approximation we assume sudden transitions of electric current; i.e. the electric current on the surface varies continuously until a region of shadow where the induced electric current drops to zero.

Although theoretically admissible, we always assume no self-shadowing of the main reflector. Shadowing is permissible only when considering blockage effects.

Various sources are permitted to illuminate the main reflector. These may be placed at any position and orientation. Feed types include: a standard rectangular aperture horn, a dipole source, a plane wave illuminator.

The main purpose of the software is to compute radiation patterns in the far-field of the antenna. However there are applications where the near field is also required and, if there is blockage, a means to re-compute the far field from the (possibly blocked) near field. These are all options within the software.

2 The mathematical formulation

2.1 The scattered far field

The radiation pattern and gain are determined from the scattered far-field. We define the far-field electric field scattering coefficient by $\underline{E}_s$, where

$$\underline{E}_s = \lim_{r\to\infty}(k_0 r e^{jk_0 r}\underline{E}(r)) \tag{2-1}$$

where k_0 is the free space wave number, $r = |\underline{r}|$, $\underline{r}$ is a point in space and $\underline{E}(\underline{r})$ is the electric field at $\underline{r}$. The coordinate origin is assumed at the reflector and we may use a standard definition for $\underline{E}_s$ (e.g. [1]) given by,

$$\begin{aligned}\underline{E}_s = \frac{-jZ_0}{4\pi}\iint_{\mathcal{S}}(\underline{J}_e - [\underline{J}_e.\underline{\hat{r}}]\underline{\hat{r}})e^{jk_0\underline{r}'.\underline{\hat{r}}}\,k_0^2\,ds' \\ + \frac{j}{4\pi}\underline{\hat{r}}\times\iint_{\mathcal{S}}\underline{J}_m e^{jk_0\underline{r}'.\underline{\hat{r}}}\,k_0^2\,ds'\end{aligned} \tag{2-2}$$

where Z_0 is the free space impedance, $\underline{r} = r\underline{\hat{r}}$ is the observation vector and $\underline{r}'$ is a point in the surface $\mathcal{S}$ over which the integration is performed. $\underline{J}_e$ is the equivalent electric current and $\underline{J}_m$ is the equivalent magnetic current on the surface defined by,

$$\underline{J}_e(\underline{r}') = \underline{\hat{n}}\times(\underline{H}_{inc} + \underline{H}_{ref} - \underline{H}_{tran}) \tag{2-3}$$

and

$$\underline{J}_m(\underline{r}') = -\underline{\hat{n}}\times(\underline{E}_{inc} + \underline{E}_{ref} - \underline{E}_{tran}) \tag{2-4}$$

where $\underline{\hat{n}}$ is the outward pointing surface normal at the point $\underline{r}'$ (see figure 2-1), $\underline{H}_{inc}$ is the incident magnetic field in the absence of the scatterer, $\underline{H}_{ref}$ is the scattered magnetic field just off the scatter in the direction of positive $\underline{\hat{n}}$ and $\underline{H}_{tran}$ is the transmitted magnetic field (non-zero only if the surface represents an interface between two regions of dielectric). Similar definitions apply to the electric field terms. Under the PO approximation for a perfect conductor,

$$\underline{J}_e = 2\underline{\hat{n}}\times\underline{H}_{inc} \tag{2-5}$$

$$\underline{J}_m = 0 \tag{2-6}$$

so far our purpose we may ignore the second integral term in (2-2)

It may be shown that the gain of the antenna, in the direction $\underline{\hat{r}}$, is given by

$$G = \frac{|\underline{E}_s|^2}{2k_0^2 Z_0}.\frac{4\pi}{P_{inc}} \tag{2-7}$$

where P_{inc} is the incident power. For a feed, this may be defined as the power radiated by the feed. For an incident plane wave, this may be defined as the product of the projected area of the reflector in the direction of the incident plane wave and the intensity of the plane wave.

Since we assume the source field from the feed does not contribute to the scattered far field we will also assume that P_{inc} is the radiated power from the feed, radiated in all directions. This definition is especially significant for a dipole feed source.

We now assume that the curved surface can be approximately represented by a polygonal surface consisting of contiguous non-overlapping flat facets whose maximum deviation from the curved surface that they represent is small. Over the i_{th} such flat facet, $\underline{\hat{n}}(\underline{r}') = \underline{\hat{n}}_i$ is a constant. In addition, if we further assume that each such facet is small enough to lie in the far field of the source then the direction of incidence does not vary over the facet and $\underline{H}_{inc}(\underline{r}')$ may be approximated by,

$$\underline{H}_{inc}(\underline{r}') \approx \underline{H}_i e^{-j\underline{k}_i \cdot \underline{r}'} \text{ over the } i_{th} \text{ facet} \tag{2-8}$$

where $\underline{k}_i = k_0 \underline{\hat{k}}_i$, $\underline{\hat{k}}_i$ is the direction of the incident wave on the i_{th} facet and $\underline{H}_i$ is the constant vector amplitude of the magnetic field over the i_{th} facet.

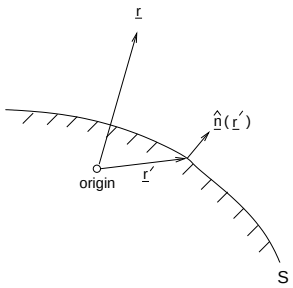

Figure 2-1: Definition of surface normal

It then follows that,

$$\underline{E}_s(\underline{\hat{k}}_s) = \frac{-jZ_0 k_0^2}{4\pi} \sum_i \underline{J}_i^{(p)} \iint_{S_i} e^{jk_0 \underline{r}' \cdot (\underline{\hat{k}}_s - \underline{\hat{k}}_i)} \, ds' \tag{2-9}$$

where the projected current amplitude, $\underline{J}_i^{(p)}$, is constant over each facet;

$$\underline{J}_i^{(p)} = \underline{J}_i - [\underline{J}_i . \underline{\hat{k}}_s] \underline{\hat{k}}_s \tag{2-10}$$

where

$$\underline{J}_i = 2\underline{\hat{n}}_i \times \underline{H}_i \tag{2-11}$$

and $\underline{\hat{k}}_s = \underline{\hat{r}}$ is the scatter direction. The sum over i is over all the flat facets, S_i, representing the surface.

2.2 Scattered field directions and definitions

The direction of the required scattered field is specified within the software using spherical polar coordinates, θ and ϕ. These angles are illustrated in figure 3-1. The field directions are specified with respect to the directions of increasing θ, defined by the unit vector $\underline{\hat{\theta}}$ and the direction of increasing ϕ, defined by the unit vector $\underline{\hat{\phi}}$. Expressed in Cartesian coordinates,

$$\begin{aligned} \underline{\hat{\theta}} &= \cos\phi\cos\theta\underline{\hat{x}} + \sin\phi\cos\theta\underline{\hat{y}} - \sin\theta\underline{\hat{z}} \\ \underline{\hat{\phi}} &= -\sin\phi\underline{\hat{x}} + \cos\phi\underline{\hat{y}} \end{aligned} \tag{2-12}$$

We may write the components of the scattered electric field vector,

$$\underline{E}_s(\underline{\hat{k}}_s) = E_1\underline{\hat{\theta}} + E_2\underline{\hat{\phi}} \tag{2-13}$$

The coefficients E_1 and E_2 are the (complex) scattering amplitudes in these directions. The program permits a rotation of this polarisation base by a constant angle ζ_s so that final outputs may be expressed at an angle ζ_s to the spherical angle directions, i.e. in the directions,

$$\begin{aligned} \underline{\hat{\theta}}' &= \cos\zeta_s\underline{\hat{\theta}} - \sin\zeta_s\underline{\hat{\phi}} \\ \underline{\hat{\phi}}' &= \sin\zeta_s\underline{\hat{\theta}} + \cos\zeta_s\underline{\hat{\phi}} \end{aligned} \tag{2-14}$$

Writing,

$$\underline{E}_s(\underline{\hat{k}}_s) = E_1'\underline{\hat{\theta}}' + E_2'\underline{\hat{\phi}}' \tag{2-15}$$

we therefore have,

$$\begin{aligned} E_1' &= \cos\zeta_s\, E_1 - \sin\zeta_s\, E_2 \\ E_2' &= \sin\zeta_s\, E_1 + \cos\zeta_s\, E_2 \end{aligned} \tag{2-16}$$

The default angle for $\zeta_s = 0$, but may be changed using the input tag command *FARPOL*, as given later. Note that the sense of the angle for ζ_s is chosen so that a positive angle ζ_s is equivalent to a positive rotation of the feed by an angle ψ_f in an axisymmetric system. Thus in a system where the reflector is axisymmetric and the feed lies on this axis, choosing $\psi_f = \zeta_s$ implies that E_1' and E_2' remain invariant of ζ_s when $\theta = \phi = 90°$.

The key word *ANGLES* defines a block of required θ and ϕ angles directly. However, it is also required to specify an arbitrary angular cut which cannot be achieved for constant θ or constant ϕ. This is accomplished using the *ANGLECUT* key word. In this case, we define a particular direction specified by $\theta = \theta_0$ and $\phi = \phi_0$. We then specify a scan direction angle, η_0, made with respect to the direction $\underline{\hat{\theta}}_0$ and vary the angle ν along this cut. This is illustrated in figure 2-2 below.

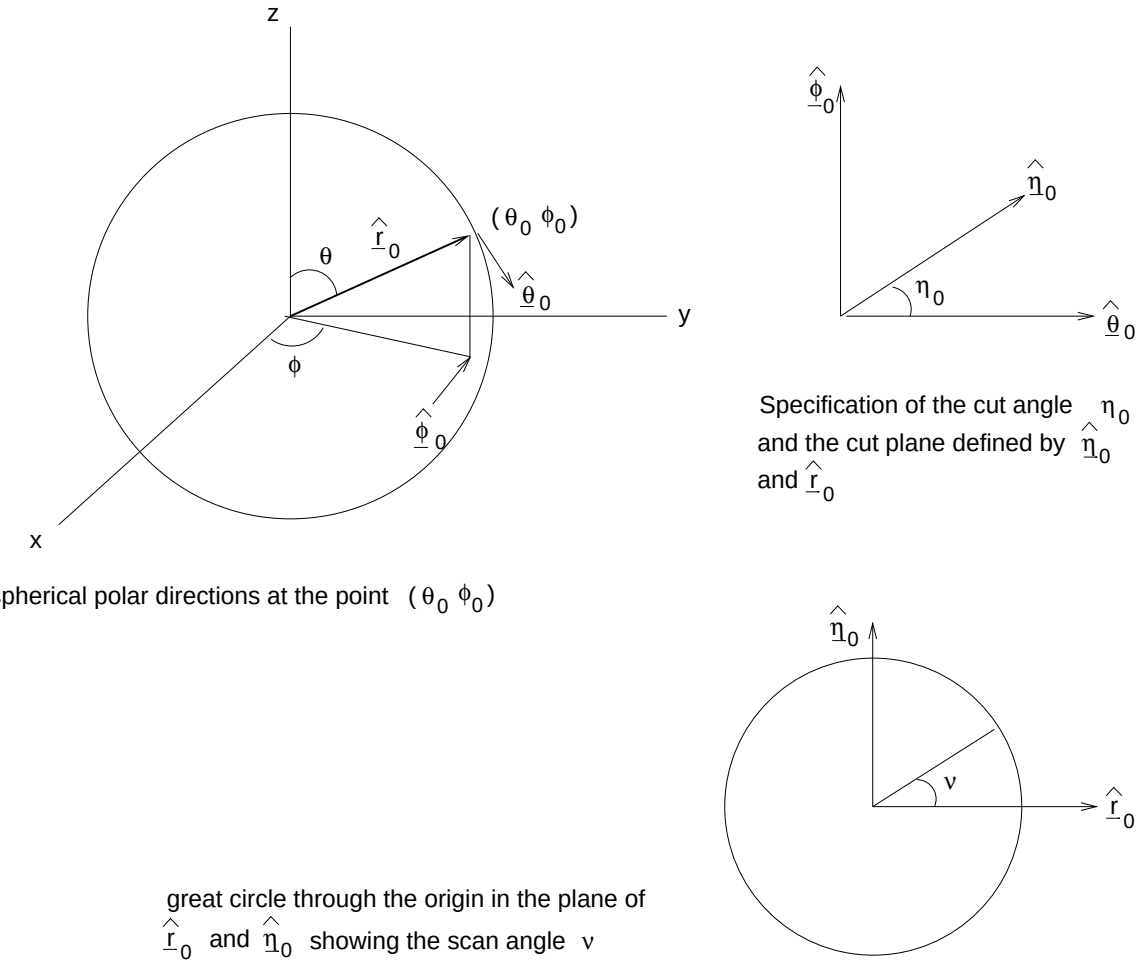

Figure 2-2: Arbitrary cuts on the surface of a sphere, specified by $ANGLECUT$

The angle η_0 at the point (θ_0, ϕ_0) defines the direction,

$$\underline{\hat{\eta}}_0 = \cos\eta_0\,\underline{\hat{\theta}}_0 + \sin\eta_0\,\underline{\hat{\phi}}_0 \tag{2-17}$$

where

$$\begin{aligned}\underline{\hat{\theta}}_0 &= \cos\phi_0\cos\theta_0\,\underline{\hat{x}} + \sin\phi_0\cos\theta_0\,\underline{\hat{y}} - \sin\theta_0\,\underline{\hat{z}} \\ \underline{\hat{\phi}}_0 &= -\sin\phi_0\,\underline{\hat{x}} + \cos\phi_0\,\underline{\hat{y}}\end{aligned} \tag{2-18}$$

We also require the direction vector at (θ_0, ϕ_0), defined by,

$$\underline{\hat{r}}_0 = \cos\phi_0\sin\theta_0\,\underline{\hat{x}} + \sin\phi_0\sin\theta_0\,\underline{\hat{y}} + \cos\theta_0\,\underline{\hat{z}} \tag{2-19}$$

The direction vector under the scan is now given by,

$$\underline{\hat{r}} = \cos\nu\,\underline{\hat{r}}_0 + \sin\nu\,\underline{\hat{\eta}}_0 \tag{2-20}$$

but we also have,

$$\underline{\hat{r}} = \cos\phi\sin\theta\,\underline{\hat{x}} + \sin\phi\sin\theta\,\underline{\hat{y}} + \cos\theta\,\underline{\hat{z}} \tag{2-21}$$

so equating terms, we obtain expressions for θ and ϕ in terms of θ_0, ϕ_0, η_0 and ν. The Cartesian components of $\underline{\hat{r}}$ are given by,

$$\begin{aligned}r_x(\nu) &= \cos\nu\cos\phi_0\sin\theta_0 + \sin\nu\cos\eta_0\cos\phi_0\cos\theta_0 - \sin\nu\sin\eta_0\sin\phi_0 \\ r_y(\nu) &= \cos\nu\sin\phi_0\sin\theta_0 + \sin\nu\cos\eta_0\sin\phi_0\cos\theta_0 + \sin\nu\sin\eta_0\cos\phi_0 \\ r_z(\nu) &= \cos\nu\cos\theta_0 - \sin\nu\cos\eta_0\sin\theta_0\end{aligned} \tag{2-22}$$

so,

$$\cos\theta = r_z \tag{2-23}$$

from which we may obtain θ, under the assumption $0 \le \theta \le \pi$, and

$$\phi = arctan_2(r_y, r_x) \tag{2-24}$$

where the $arctan_2$ function is the angle-unambiguous arc-tangent of the ratio r_y/r_x.

2.3 The Physical Optics integration

In evaluating the PO integrals, we assume the facets are triangular in which case the integrals are of Gorden's generic form,

$$I_G(\underline{w})_i = \iint_{\mathcal{A}_i} e^{jk_0(\underline{w}.\underline{x})}\, dx \tag{2-25}$$

over a general triangle $\mathcal{A}_i$ with area A_i. If the triangle is defined by the coordinates $\underline{x}_1$, $\underline{x}_2$ and $\underline{x}_3$ of the vertices, then it can be shown after some algebra that,

$$I_G = \frac{2Ae^{jk_0\underline{x}_1.\underline{w}}}{js_{31}}\left(\frac{e^{js_{31}}(e^{js_{23}}-1)}{js_{23}} - \frac{(e^{js_{21}}-1)}{js_{21}}\right) \tag{2-26}$$

where the i subscripts are no longer shown explicitly in the notation and

$$\begin{aligned} s_{21} &= k_0(\underline{x}_2 - \underline{x}_1).\underline{w} \\ s_{31} &= k_0(\underline{x}_3 - \underline{x}_1).\underline{w} \\ s_{23} &= k_0(\underline{x}_2 - \underline{x}_3).\underline{w} \end{aligned} \tag{2-27}$$

In the limit that one or more of the s terms tends to zero, the expression (2-26) must be modified to permit computation. To do this, define

$$P_0(\delta) = \int_0^1 e^{j\eta\delta} d\eta \tag{2-28}$$

$$P_1(\delta) = \int_0^1 \eta e^{j\eta\delta} d\eta \tag{2-29}$$

$$P_2(\delta) = \int_0^1 \eta^2 e^{j\eta\delta} d\eta \tag{2-30}$$

which may be evaluated by,

$$P_0(\delta) = \begin{cases} \frac{1}{j\delta}(e^{j\delta}-1) & \text{for } \delta \neq 0 \\ (1-\delta^2/6) + \frac{j\delta}{2}(1-\delta^2/12) & \text{for } \delta \to 0 \end{cases} \tag{2-31}$$

$$P_1(\delta) = \begin{cases} \frac{e^{j\delta}}{j\delta} - P_0(\delta)/(j\delta) & \text{for } \delta \neq 0 \\ (1/2 - \delta^2/8) + j\delta/3 & \text{for } \delta \to 0 \end{cases} \tag{2-32}$$

$$P_2(\delta) = \begin{cases} \frac{e^{j\delta}}{j\delta} - 2P_1(\delta)/(j\delta) & \text{for } \delta \neq 0 \\ 1/3 + j\delta/4 & \text{for } \delta \to 0 \end{cases} \tag{2-33}$$

Now, for $s_{31} \to 0$,

$$I_G = 2Ae^{jk_0\underline{x}_1.\underline{w}}[(1 - js_{31}/2)P_0(s_{21}) + (-1 + js_{31})P_1(s_{21}) - j(s_{31}/2)P_2(s_{21})] \tag{2-34}$$

For $s_{31} \neq 0$ we must consider the cases $s_{23} \to 0$ and/or $s_{21} \to 0$ in which case,

$$I_G = \frac{2Ae^{jk_0\underline{x}_1.\underline{w}}}{js_{31}}[e^{js_{31}}P_0(s_{23}) - P_0(s_{21})] \tag{2-35}$$

The area of each triangle may be conveniently evaluated using Heron's formula,

$$A = \sqrt{s(s-a)(s-b)(s-c)} \tag{2-36}$$

where

$$s = \frac{a+b+c}{2} \text{ and } a = |\underline{x}_3 - \underline{x}_2|,\ b = |\underline{x}_3 - \underline{x}_1|,\ c = |\underline{x}_2 - \underline{x}_1| \tag{2-37}$$

If a surface is represented by triangles which are in turn defined by nodes representing the triangle vertices, there is a numerical dichotomy; the fields and directions of waves are naturally defined at the nodes, but they are required on the triangle surfaces where the surface normals are defined. However, since it is assumed that these quantities are slowly varying, using (2-8) we may take,

$$\underline{H}_i = \frac{1}{3}(e^{j\underline{k}_i.\underline{x}_{i1}}\underline{H}_{inc}(\underline{x}_{i1}) + e^{j\underline{k}_i.\underline{x}_{i2}}\underline{H}_{inc}(\underline{x}_{i2}) + e^{j\underline{k}_i.\underline{x}_{i3}}\underline{H}_{inc}(\underline{x}_{i3})) \tag{2-38}$$

where $\underline{x}_{ij}$ are the vertices $\underline{x}_j$ of the i_{th} triangle. The incident wave directions $\underline{k}_i$ must also be approximated. We permit two ways of doing this within the software, though in practise there is very little difference between them if our assumptions on slow variations are correct.

The first method assumes that the phase centre of the source is specified, and that the field radiated by the source may be considered to radiate from this phase centre at all points over the reflector surface. In this case, if the phase centre is specified at the coordinates $\underline{r}_0$, then the vector from the phase centre to the triangle centroid is given by,

$$\underline{r}_i^{(c)} = \frac{1}{3}(\underline{x}_{i1} + \underline{x}_{i2} + \underline{x}_{i3}) - \underline{r}_0 \tag{2-39}$$

hence

$$\underline{k}_i = k_0 \underline{r}_i^{(c)} / |\underline{r}_i^{(c)}| \tag{2-40}$$

The second method makes no assumption on the position of the phase centre, but assumes the wave direction is that of the Poynting vector. This basically assumes that the field is locally planar with the E and H-fields orthogonal and in-phase. The direction of the wave at a general point in space may be defined by,

$$\hat{\underline{k}}^{(p)}(\underline{x}) = \frac{\Re(\underline{E}(\underline{x}) \times \underline{H}^{\star}(\underline{x}))}{|\Re(\underline{E}(\underline{x}) \times \underline{H}^{\star}(\underline{x}))|} \tag{2-41}$$

which is required when $\underline{E}(\underline{x}) = \underline{E}_{inc}(\underline{x})$ and $\underline{H}(\underline{x}) = \underline{H}_{inc}(\underline{x})$. Evaluation of the incident fields at the nodal coordinates depends on the method employed (see below).

Again, since the fields are only available at the nodes, we employ an average defined by,

$$\underline{k}_i = \frac{k_0(\hat{\underline{k}}^{(p)}(\underline{x}_{i1}) + \hat{\underline{k}}^{(p)}(\underline{x}_{i2}) + \hat{\underline{k}}^{(p)}(\underline{x}_{i3}))}{|\hat{\underline{k}}^{(p)}(\underline{x}_{i1}) + \hat{\underline{k}}^{(p)}(\underline{x}_{i2}) + \hat{\underline{k}}^{(p)}(\underline{x}_{i3})|} \tag{2-42}$$

Finally, the surface normals $\underline{\hat{n}}_i$ must be computed. There is a possible sign ambiguity that must be resolved since we require the surface normals always to be pointed in the "outward" direction, remembering that the surface interface represents the surface of a solid region of conductor. We define this such that $\underline{\hat{n}}_i.\underline{k}_i < 0$ for all points on the scattering surface. Note that transition regions, where there may be a required change in sign of $\underline{\hat{n}}_i.\underline{k}_i$, are ruled out by our assumption that no self-shadowing of the reflector is permitted.

We therefore define,

$$\underline{\hat{n}}_i' = \frac{(\underline{x}_{i2} - \underline{x}_{i1}) \times (\underline{x}_{i3} - \underline{x}_{i2})}{|(\underline{x}_{i2} - \underline{x}_{i1}) \times (\underline{x}_{i3} - \underline{x}_{i2})|} \tag{2-43}$$

and

$$\underline{\hat{n}}_i = \begin{cases} \underline{\hat{n}}_i' & \text{if } \underline{\hat{n}}_i'.\underline{r}_i^{(c)} < 0 \\ -\underline{\hat{n}}_i' & \text{if } \underline{\hat{n}}_i'.\underline{r}_i^{(c)} > 0 \end{cases} \tag{2-44}$$

2.4 The incident fields

In the previous section it was assumed that the incident fields, $\underline{E}_{inc}(\underline{x})$ and $\underline{H}_{inc}(\underline{x})$ are available at the coordinates specifying the triangular nodes. The software has various feed options which calculate or estimate these fields in different manners. An electric dipole may be specified using the *DIPOLE* key word, or a plane wave using the *PLANEWAVE* key word, a modified non-paraxial Gaussian beam using the *MGAUSSIAN* key word and a standard paraxial Gaussian beam using the *PGAUSSIAN*. These latter three types provide an analytic determination of these fields. More generally, however, no analytic formulation is available.

A rectangular horn or a user specified aperture distribution may be provided, too. The latter allows either the electric field or both the electric and the magnetic field to be given on a plane in space. Once these aperture fields are specified the far-fields $\underline{E}_{inc}(\underline{x})$ and $\underline{H}_{inc}(\underline{x})$ may be determined by integration.

The basic formulas for the E and H radiated fields, evaluated at a point $\underline{R}_p$, in terms of the aperture fields E_r and H_r are given by Silver (section 5-11, of [5]),

$$\underline{E}(\underline{R}_p) = \frac{1}{4\pi j\omega\epsilon_0}\int_A \left(k_0^2(\underline{n}\times\underline{H}_r)\Psi + (\underline{n}\times\underline{H}_r).\nabla(\nabla\Psi) + j\omega\epsilon_0(\underline{n}\times\underline{E}_r)\times\nabla\Psi\right) ds \quad (2\text{-}45)$$

and

$$\underline{H}(\underline{R}_p) = \frac{-1}{4\pi j\omega\mu_0}\int_A \left(k_0^2(\underline{n}\times\underline{E}_r)\Psi + (\underline{n}\times\underline{E}_r).\nabla(\nabla\Psi) - j\omega\mu_0(\underline{n}\times\underline{H}_r)\times\nabla\Psi\right) ds \quad (2\text{-}46)$$

where the surface integral is over the aperture plane, A, and $\underline{E}_r(\underline{x}_s)$ and $\underline{H}_r(\underline{x}_s)$ are evaluated on this plane and $\underline{n}(\underline{x}_s)$ is the outward pointing surface normal in the direction of the radiating field, taken as a constant unit vector for this application. ϵ_0 and μ_0 are the permittivity and permeability of free space. Ψ is defined by,

$$\Psi(\underline{R}_p, \underline{x}_s) = \frac{e^{-jk_0 r}}{r} \quad (2\text{-}47)$$

where

$$r^2 = |\underline{R}_p - \underline{x}_s|^2 \quad (2\text{-}48)$$

Notice that only the transverse fields with components in the plane normal to $\underline{n}$ are required in the formulation. Without loss of generality, assume $\underline{n} = \underline{\hat{z}}$ where $\underline{\hat{z}}$ is the unit vector normal to the plane surface and we have,

$$\underline{n}\times\underline{E}_r = \underline{\hat{z}}\times\underline{E}_t \quad (2\text{-}49)$$

and

$$\underline{n}\times\underline{H}_r = \underline{\hat{z}}\times\underline{H}_t \quad (2\text{-}50)$$

where $\underline{E}_t$ and $\underline{H}_t$ are the tangential field components of the electric and magnetic fields. Specification of $\underline{E}_t$ and $\underline{H}_t$ depends on the feed method, as outlined shortly.

In (2-45) and (2-46) the "∇" gradient terms are interpreted as acting on r, i.e.

$$\nabla\Psi = \nabla\left(e^{-jk_0 r}/r\right) = \left(jk_0 + \frac{1}{r}\right)\frac{e^{-jk_0 r}}{r}\,\underline{\hat{r}} \tag{2-51}$$

and

$$(\underline{J}.\nabla)\nabla\Psi = \left(-k_0^2(\underline{J}.\underline{\hat{r}})\underline{\hat{r}} + \frac{3}{r}(jk_0 + 1/r)(\underline{J}.\underline{\hat{r}})\underline{\hat{r}} - \frac{\underline{J}}{r}(jk_0 + 1/r)\right)\frac{e^{-jk_0 r}}{r} \tag{2-52}$$

where

$$\underline{\hat{r}} = \frac{\underline{R}_p - \underline{x}_s}{|\underline{R}_p - \underline{x}_s|} \tag{2-53}$$

For a rectangular feed, we assume a feed which flares from waveguide with dimensions d_x by d_y to an aperture of dimensions D_x by D_y over a length H, where $D_x > d_x$ and $D_y > d_y$. The flare introduces a phase distortion. We assume operation in either the TE01 or TE10 mode in which case the aperture field takes the approximate form,

$$\underline{E}_t = E_a\underline{\hat{y}}\cos\left(\frac{\pi x}{D_x}\right)\exp\left(-j(\alpha(2x/D_x)^2 + \beta(2y/D_y)^2)\right) \tag{2-54}$$

or the alternative mode with interchange of x and y coordinates. The values for α and β may be approximated by,

$$\begin{aligned} \alpha &= k_0(\sqrt{r_x^2 + (D_x/2)^2} - r_x) \\ \beta &= k_0(\sqrt{r_y^2 + (D_y/2)^2} - r_y) \end{aligned} \tag{2-55}$$

where

$$r_x \approx r_y \approx H \tag{2-56}$$

The coefficient E_a is a normalisation constant which is adjusted so that the feed always radiates exactly 1 watt. We assume that the admittance at the aperture is approximately that of free space, in which case it can be shown that,

$$E_a = \frac{2}{\sqrt{D_x D_y}}\left(\frac{\epsilon_0}{\mu_0}\right)^{-1/4} \tag{2-57}$$

The magnetic field and the electric field are related in a waveguide by the modal admittance. At the aperture of a flared horn this relation is only approximate. We have attempted to use numerical differentiation on the assumed electric field to generate a more consistent approximation but generally this seems to be less reliable. Our most robust approximation is simply to assume that at the aperture,

$$\underline{H}_t = \underline{\hat{z}} \times \underline{E}_t/Z_0 \tag{2-58}$$

using the free space admittance. More accurate field distributions require the export of aperture field distribution files, e.g. from CST [4] using the *FEEDFILE* option.

When importing aperture field distributions we have two options; either we specify both the electric and magnetic fields in which case the approximation (2-58) is not made, or we specify the E-field and use (2-58) to estimate the H-field on the aperture.

Finally, we note that in the formulation (2-45) is not required to generate the electric field scattered by the reflector in (2-9), only (2-46). However we calculate both since the overhead in doing so is small and this allows us to determine Poynting vectors and local directions of incidence $\underline{k}_i$ (the wave is assumed locally planar over a facet of the reflector) when the phase centre of the source field is not defined.

2.4.1 Plane waves incident

An incident plane wave is specified in the antenna reflector (global) coordinates. Its general form is given by,

$$\underline{E}(\underline{r}) = \left(\cos\chi_\alpha \underline{\hat{\theta}}_p + e^{j\chi_\beta} \sin\chi_\alpha \underline{\hat{\phi}}_p \right) \exp(-jk_0 \underline{\hat{k}}_p.(\underline{r} - \underline{r}_0)) \tag{2-59}$$

where $\underline{r}_0$ is the position of the phase centre (defined by the *FEEDCEN* key word), and,

$$\underline{\hat{k}}_p = \underline{\hat{\phi}}_p \times \underline{\hat{\theta}}_p \tag{2-60}$$

is the direction of the incident wave, pointing towards the global origin, and $\underline{\hat{\theta}}_p$ and $\underline{\hat{\phi}}_p$ are the directions of increasing θ and ϕ for a wave incident at these angles. The angles χ_α and χ_β are the ellipticity and phase polarisation angles.

2.4.2 Dipole incident field

A point source Hertzian dipole field is defined by an electric Hertzian dipole at the phase centre $\underline{r}_0$ in the local feed coordinate system with axis aligned with the z' axis. It may be rotated to arbitrary orientation using the *FEEDROT* key word. We define in terms of Cartesian components in the primed local coordinates. We have the magnetic field,

$$\underline{H}(\underline{r}') = \frac{E_0}{Z_0} \sin\theta' e^{-jk_0 r'} f_1(\underline{r}') \, \underline{\hat{\phi}}' \tag{2-61}$$

where

$$\underline{\hat{\phi}}' = -\sin\phi' \, \underline{\hat{x}}' + \cos\phi' \, \underline{\hat{y}}' \tag{2-62}$$

and Z_0 is the impedance of free space, θ' and ϕ' are the spherical polar angles, r' is the distance from the source in the local coordinate system and E_0 is the normalised field strength which may be defined in terms of the total radiated power P_0 as,

$$E_0 = k_0 \sqrt{\frac{3 Z_0 P_0}{4\pi}} \tag{2-63}$$

and

$$f_1(\underline{r}') = \frac{1}{k_0 r'} - \frac{j}{(k_0 r')^2} \tag{2-64}$$

The electric field is given by,

$$\underline{E}(\underline{r}') = E_0 e^{-jk_0 r'} \left[\sin\theta' f_2(\underline{r}')\, \hat{\underline{\theta}}' + \cos\theta' f_3(\underline{r}')\, \hat{\underline{r}}' \right] \tag{2-65}$$

where

$$\hat{\underline{\theta}}' = \cos\theta' \cos\phi'\, \hat{\underline{x}}' + \cos\theta' \sin\phi'\, \hat{\underline{y}}' - \sin\theta'\, \hat{\underline{z}}' \tag{2-66}$$

$$\hat{\underline{r}}' = \sin\theta' \cos\phi'\, \hat{\underline{x}}' + \sin\theta' \sin\phi'\, \hat{\underline{y}}' + \cos\theta'\, \hat{\underline{z}}' \tag{2-67}$$

$$f_2(\underline{r}') = f_1(\underline{r}') - \frac{1}{(k_0 r')^3} \tag{2-68}$$

and

$$f_3(\underline{r}') = -2\left(\frac{1}{(k_0 r')^3} - \frac{j}{(k_0 r')^2} \right) \tag{2-69}$$

2.4.3 Modified Gaussian beam feed field

The theory of scalar Gaussian beams is standard and is readily available in the literature. It is, however, a paraxial approximation only accurate when the beam angle is relatively small (e.g. with semi-angle between axis and e-field e^{-1} points rather less than 30 degrees). The theory has been extended to provide much greater accuracy for much wider angles [6] and it is a slight variation of this theory that we will apply. It is implemented using the *MGAUSSIAN* key word.

As for the dipole feed field, we will assume specification in local coordinates so that the orientation and polarisation (assumed linear) can be arbitrarily changed. In local coordinates the beam axis is the $\hat{\underline{z}}'$ direction, with electric field polarised in the $\hat{\underline{y}}'$ direction.

Let

$$\underline{\rho}' = \underline{r}' - (\underline{r}'.\hat{\underline{z}}')\, \hat{\underline{z}}' \tag{2-70}$$

be the axial radial vector about the $\hat{\underline{z}}'$ axis with $\rho' \equiv |\underline{\rho}'|$. In [6],the non-paraxial Gaussian electric field vector may be approximated as [6],

$$\underline{E}(\underline{\rho}', z') = E_0(\underline{\rho}', z') \left[\sin\phi'\, \hat{\underline{\theta}}' + \cos\Theta' \cos\phi'\, \hat{\underline{\phi}}' \right] \tag{2-71}$$

where $\hat{\underline{\theta}}'$ and $\hat{\underline{\phi}}'$ are defined in (2-66) and (2-62). Note that Θ' is the angle made with the phase centre O_R of the beam, illustrated in figure 2-3 and defined below. However, the direction $\hat{\underline{\theta}}'$ is here defined, as in [6], as the direction of increasing θ'. This is the angle made with the coordinate origin and not the phase centre. While this is a good approximation well away from the beam waste, it is poor near the beam waste and completely fails on the beam waste.[1] We therefore modify the function so that the directions are referenced with respect to angles made with the phase centre,

$$\underline{E}(\underline{\rho}', z') = E_0(\underline{\rho}', z') \left[\sin\phi'\, \hat{\underline{\Theta}}' + \cos\Theta' \cos\phi'\, \hat{\underline{\phi}}' \right] \tag{2-72}$$

[1] For example when $\phi' = \pi/2$ and $z' = 0$ a $y-$polarised electric field on the aperture would not be defined since the described field would be polarised in the $z-$ direction.

where

$$\underline{\hat{\Theta}}' = \cos\Theta' \cos\phi' \, \underline{\hat{x}}' + \cos\Theta' \sin\phi' \, \underline{\hat{y}}' - \sin\Theta' \, \underline{\hat{z}}' \tag{2-73}$$

The azimuthal direction $\underline{\hat{\phi}}'$ is defined as above (2-62). We will also define the radial direction to the phase centre,

$$\underline{\hat{R}}' = \sin\Theta' \cos\phi' \, \underline{\hat{x}}' + \sin\Theta' \sin\phi' \, \underline{\hat{y}}' + \cos\Theta' \, \underline{\hat{z}}' \tag{2-74}$$

The accuracy of this modification is untested but agrees with [6] in the regions where [6] has provided numerical tests and is valid on the beam waste. Proceeding with the definitions, let the Rayleigh range,

$$z_R = \frac{k_0 w_0^2}{2} \tag{2-75}$$

where w_0 is the beam waste radius (see below). We further define the beam width $w(z)$ and axial distance to phase centre $R(z)$ by,

$$w(z') = w_0 \sqrt{1 + (z'/z_R)^2} \tag{2-76}$$

and

$$R(z') = z' \left(1 + \left(\frac{z_R}{z'} \right)^2 \right) \tag{2-77}$$

Rather than employing $R(z')$ we will define its reciprical,

$$p(z') = \frac{1}{R(z')} = \frac{z'}{(z')^2 + (z_R)^2} \tag{2-78}$$

This is because at the beam waste $R(z') \to \infty$ which causes numerical problems.

The distance between the points P and P' in the figure (and in [6]) is given by,

$$d \equiv \overline{PP'} = \sqrt{R^2(z') + (\rho')^2} - R(z') = R(z') \left(\frac{1}{\cos\Theta'} - 1 \right) \tag{2-79}$$

This is redefined in terms of $p(z')$ by,

$$d(z') = \begin{cases} \dfrac{1}{p(z')} \left(\sqrt{1 + (\rho' p(z'))^2} - 1 \right) & \text{for } |p(z')| z_R > \epsilon \\ (\rho')^2 \, p(z')/2 & \text{for } |p(z')| z_R \leq \epsilon \end{cases} \tag{2-80}$$

where $0 < \epsilon \ll 1$ is some very small number consistent with computational precision (e.g. $\epsilon = 10^{-6}$ in double precision arithmetic).

The angle Θ' is defined by $\tan\Theta' = \rho'/R(z')$ so that,

$$\cos\Theta' = \frac{1}{\sqrt{1 + (\rho' p(z'))^2}} \tag{2-81}$$

and

$$\sin\Theta' = \frac{\rho' p(z')}{\sqrt{1 + (\rho' p(z'))^2}} \tag{2-82}$$

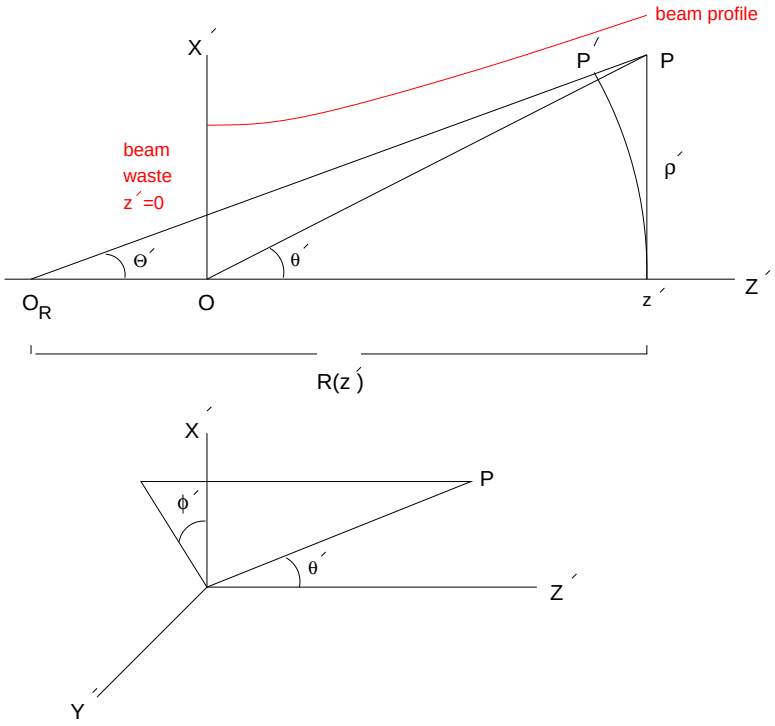

Figure 2-3: Angle and coordinate definitions (primed feed coordinates) in the modified Gaussian beam

We may now define the electric field magnitude as in [6] by,

$$E_0(\underline{\rho}', z') = A_0 \frac{w_0}{w(z')} \cos\Theta' \exp\left(\frac{-(\rho' \cos\Theta')^2}{\big(w(z')\big)^2}\right) \exp\big(-j\Phi(\underline{\rho}', z')\big) \tag{2-83}$$

with beam phase, $\Phi(\underline{\rho}', z')$ given by,

$$\Phi(\underline{\rho}', z') = -\arctan\left(\frac{z'}{z_R}\right) + k_0 z' + k_0 d(z') \tag{2-84}$$

The Gaussian beam shape is controlled by the single parameter w_0. In the paraxial approximation w_0 represents the radius at the beam waste at which the electric field magnitude falls to e^{-1} of the value at the origin, A_0. In the non-paraxial form, we apply the same criterion to the field amplitude. Thus as $z' \to \infty$, for which $\Theta' \to \theta'$, we require the value of θ', θ_0, such that

$$\lim_{z'\to\infty}\left(\frac{A_0(w_0/w(z'))\cos\Theta' \exp\big(-(\rho')^2\cos^2\Theta'/w^2(z')\big)}{A_0(w_0/w(z'))\cos\Theta'}\right) = e^{-1} \tag{2-85}$$

This occurs when $\rho' \cos\theta_0 = w(z')$ as $z' \to \infty$ or when

$$w_0 = \frac{2}{k_0 \sin\theta_0} \tag{2-86}$$

We will take (2-86) as a definition of θ_0. We specify θ_0 and determine w_0 accordingly. To a good approximation θ_0 is the angle between the beam axis and the point at which the intensity falls to e^{-2} of its axial value[2] (equivalent to the angle at which the magnitude of the electric field falls to e^{-1} of its axial value). This is the semi-angle to the -8.69 dB point.

The exact relationship between A_0 and the total radiated power P_0 is similarly not available (though it could be computed with some dedicated analysis). When $z = 0$ the electric field under the modified Gaussian beam and the standard paraxial Gaussian beam are identical, but this may not be exactly true of the magnetic field. However, any differences are expected to be small. In the paraxial approximation, standard theory gives,

$$P_0 = \frac{1}{2}\pi w_0^2 \left(\frac{|A_0|^2}{2Z_0}\right) \tag{2-87}$$

For present purposes we will assume it holds for the non-paraxial case to a level of approximation consistent with the use of our numerical implementation of physical optics. Thus we take,

$$A_0 \approx \frac{2}{w_0}\sqrt{\frac{Z_0 P_0}{\pi}} \tag{2-88}$$

The accuracy of this approximation is only important in determining the accuracy of the power spill-over at the reflector.

Next we need to determine the associated magnetic field. Exact evaluation requires us to calculate $\underline{H}(\underline{\rho}', z')$ as

$$\underline{H}(\underline{\rho}', z') = -\frac{1}{jk_0 Z_0}\nabla \times \underline{E}(\underline{\rho}', z')$$

which results in a very complicated expression which carries terms to orders not required in the expression for $\underline{E}(\underline{\rho}', z')$. This is not calculated in [6]. Instead, we will simply assume that the incident field at the reflector is locally planar, using the phase centre point O_R as the origin. In this case we may take,

$$\underline{H}(\underline{\rho}', z') \approx \frac{1}{Z_0}\hat{\underline{R}}' \times \underline{E}(\underline{\rho}', z') \tag{2-89}$$

where $\hat{\underline{R}}'$ is given by (2-74).

Expressed in Cartesian coordinates we find the electric and magnetic fields may be given by,

$$\underline{E}(\rho', z') = E_0(\rho', z')\Big(\cos\Theta'\,\underline{\hat{y}}' - \sin\phi'\sin\Theta'\,\underline{\hat{z}}'\Big) \tag{2-90}$$

and

$$\underline{H}(\rho', z') = \frac{E_0(\rho', z')}{Z_0}\Big(-(\sin^2\Theta'\sin^2\phi' + \cos^2\Theta')\,\underline{\hat{x}}' + \sin^2\Theta'\sin\phi'\cos\phi'\,\underline{\hat{y}}' + \sin\Theta'\cos\Theta'\cos\phi'\,\underline{\hat{z}}'\Big) \tag{2-91}$$

[2] It is not exact because in general $|\underline{E}(\underline{\rho}', z')| \neq |E_0(\underline{\rho}', z')|$.

Note that this formulation represents a $y'-$polarised electric field at the beam waste, for which $z' = 0$ and $\Theta' = 0$ so we obtain the expected correct electric field polarisation here. Similarly the magnetic field is $x'-$polarised, as expected. More interestingly, away from the beam axis and away from the principal planes ($\phi' \neq n\pi/2$ for integer n) the magnetic field has a $y'-$polarised component which does not occur under the paraxial approximation.

2.4.4 The standard paraxial Gaussian beam

The standard paraxial Gaussian beam is provided here. This is mostly intended for diagnostic purposes and because it is a common method of excitation in many reflector analysis models. According to [6] the standard form does not provide as accurate a solution of Maxwell's equations as the modified form. It is implemented under the key word *PGAUSSIAN*. Under the standard model we define the electric field intensity,

$$\underline{E}_0(\rho', z') = A_0 \frac{w_0}{w(z')} \exp\left(-\frac{(\rho')^2}{w^2(z')}\right) \exp(-j\Phi(z')) \tag{2-92}$$

where

$$\Phi(\underline{\rho}', z') = -\arctan\left(\frac{z'}{z_R}\right) + k_0 z' + k_0\ (\rho')^2 p(z')/2 \tag{2-93}$$

where the expressions for A_0, z_R, $p(z')$, w_0 and $w(z')$ are defined as for the modified Gaussian beam in the previous section. The relation between θ_0 and w_0 is now different, however. Here, following standard practise, we define

$$w_0 = \frac{2}{k_0 \theta_0} \tag{2-94}$$

The analysis suggest we should take $w_0 = 2/(k_0 \tan\theta_0)$. While $\theta_0 \sim \tan\theta_0$ under the paraxial approximation one might expect greater accuracy using the tangent form. However, this appears not to be the case. For example, if we assume a paraxial Gaussian beam normally incident on a flat reflector, the PO approximation should generate a scattered field whose far-field semi-beam width is also given by θ_0.[3] This forms a useful test on the accuracy of the numerical implementation. We find better agreement with the expected value of θ_0 when (2-94) is assumed.

The vector electric and magnetic fields are just defined by,

$$\underline{E}(\rho', z') = E_0(\rho', z')\, \underline{\hat{y}}' \tag{2-95}$$

and

$$\underline{H}(\rho', z') = \frac{1}{Z_0} \underline{\hat{z}}' \times \underline{E}(\rho', z') = -\frac{E_0(\rho', z')}{Z_0}\, \underline{\hat{x}}' \tag{2-96}$$

[3] The scattered wave is defined to satisfy Maxwell's equations (unlike the paraxial incident field) if the tangential magnetic field on the reflector is assumed correct. Since any incident field can be generated as a result of *some* incident distribution it can be argued, if there is some flexibility in the definition, that it is better to obtain a relation between w_0 and θ_0 based on the scattered field rather than the incident field.

2.5 The scattered near field

We assume that there are no dielectrics present between the perfectly conducting reflector and a required set of points where the scattered near field is required. Let $\underline{r}$ be an observation point where the field is required and let $\mathcal{S}$ be the scatterer (reflector). The scattered electric field $\underline{E}(\underline{r})$ is given (see, e.g. [1]) by,

$$\underline{E}(\underline{r}) = \frac{Z_0 k_0^2}{4\pi} \int\int_{\mathcal{S}} \left[\underline{J}_e(\underline{r}') \left(-\frac{j}{k_0 R} - \frac{1}{k_0^2 R^2} + \frac{j}{k_0^3 R^3} \right) + \hat{\underline{R}} (\underline{J}_e . \hat{\underline{R}}) \left(\frac{j}{k_0 R} + \frac{3}{k_0^2 R^2} - \frac{3j}{k_0^3 R^3} \right) \right] e^{-jk_0 R} ds' \tag{2-97}$$

where, $R = |\underline{r} - \underline{r}'|$ and $\hat{\underline{R}} = (\underline{r} - \underline{r}')/R$.

Similarly, the magnetic field is given by,

$$\underline{H}(\underline{r}) = \frac{k_0^2}{4\pi} \int\int_{\mathcal{S}} \underline{J}_e(\underline{r}') \times \hat{\underline{R}} \frac{1 + jk_0 R}{k_0^2 R^2} e^{-jk_0 R} ds' \tag{2-98}$$

Again, under the PO approximation, we assume

$$\underline{J}_e = 2\hat{\underline{n}}(\underline{r}') \times \underline{H}_{inc}(\underline{r}') \text{ for } \underline{r}' \in \mathcal{S} \tag{2-99}$$

Using the same faceted representation of the surface as above, we assume that each facet is small enough to be (individually) in the far-field of a fictitious elemental dipole source at $\underline{r}$. We may then safely assume that R varies very little over any given facet and all terms other than the phase variation involving R may be taken outside the integral. It may then be shown that,

$$\underline{E}(\underline{r}) = \frac{Z_0 k_0^2}{4\pi} \sum_i I_G(\underline{w})_i \underline{Q}_i e^{-jk_0 (R_i^{(c)} + \hat{\underline{R}}_i^{(c)} . \underline{r}_i^{(c)})} \tag{2-100}$$

where

$$\underline{w} = \hat{\underline{R}}_i^{(c)} - \hat{\underline{k}}_i \tag{2-101}$$

$$\hat{\underline{R}}_i^{(c)} = \underline{R}_i^{(c)} / R_i^{(c)} \tag{2-102}$$

$$R_i^{(c)} = |\underline{R}_i^{(c)}| \tag{2-103}$$

$$\underline{R}_i^{(c)} = \underline{r} - \underline{r}_i^{(c)} \tag{2-104}$$

where I_G, $\hat{\underline{k}}_i = \underline{k}_i / k_0$ and $\underline{r}_i^{(c)}$ are defined in the previous section. The vector coefficient $\underline{Q}_i$ is given by,

$$\underline{Q}_i = \underline{J}_i \left(-\frac{j}{k_0 R_i^{(c)}} - \frac{1}{k_0^2 (R_i^{(c)})^2} + \frac{j}{k_0^3 (R_i^{(c)})^3} \right) + \hat{\underline{R}}_i^{(c)} (\underline{J}_i . \hat{\underline{R}}_i^{(c)}) \left(\frac{j}{k_0 R_i^{(c)}} + \frac{3}{k_0^2 (R_i^{(c)})^2} - \frac{3j}{k_0^3 (R_i^{(c)})^3} \right) \tag{2-105}$$

Similarly, it may be shown that

$$\underline{H}(\underline{r}) = \frac{k_0^2}{4\pi} \sum_i I_G(\underline{w})_i \underline{P}_i e^{-jk_0(R_i^{(c)} + \hat{\underline{R}}_i^{(c)} \cdot \underline{r}_i^{(c)})} \tag{2-106}$$

where the vector coefficient,

$$\underline{P}_i = \left(\frac{1 + jk_0 R_i^{(c)}}{k_0^2 (R_i^{(c)})^2} \right) \underline{J}_i \times \hat{\underline{R}}_i^{(c)} \tag{2-107}$$

2.6 Blockage effects

There are a large number of logically and computationally different methods for approximating the effect of blockages under high frequency and/or physical optics approximations. In our implementation we employ a combination of ray-tracing and PO on the main reflector, avoiding the need to recompute far fields at other than the main reflector. In outline, we assume a general blocking surface either specified by another triangulated surface represented as a user specified "points and joins" file or that the feed aperture is blocking the reflected wave. This latter option is relevant only to certain feed types (i.e. it is not relevant to an infinitesimally small dipole or incident plane wave).

The primary method follows:

1. Determine the incident wave direction $\hat{\underline{k}}_i$ at the centroid of each facet.
2. Determine the reflected wave direction $\hat{\underline{k}}_r$ at the centroid of each facet.
3. Trace a ray from the centroid of each facet with wave direction $\hat{\underline{k}}_r$.
4. Determine whether any such reflected ray intersects any part of the blocking surface.
5. If such a ray does intersect, mark the facet on the reflector, from which the ray originates, as "shadowed". It is assumed that the energy associated with this ray is totally absorbed, i.e. that the blocking structure is "black".
6. Assume any "shadowed" facet does not contribute to the far field at any scatter angle or at any frequency. Exclude "shadowed" facets from the PO sum over facets.

This is one of the simplest and fastest algorithms to establish the effects of blockage. It is probably also one of the least accurate and must be treated with care until we are able to establish the level of accuracy that may be expected. A second method is also implemented, which requires the computation of the near field. This is described in more detail in the user section on the *NEARFIELD* command.

3 Use of the software

3.1 Introduction

The software is driven from an input script file using a command line console. The name of the executable is arbitrary, so let us assume here it is called **testpo**. The command would thus read:

testpo <filename>

where **<filename>** is the name of the script file. The program searches for recognised commands designated by a key word followed by a list of parameters. Unrecognised words are ignored, but it is good practise to comment out any unused commands with a symbol such as '%'. Generally (and as of version 0.0.012) of the software, any commands may be entered in any order.

The commands specify the calculation options, required frequencies and scan angles, geometry designations, coordinate rotations, etc. There are two coordinate systems used within the software. The first is the "global coordinate system" in which the main reflector is defined and to which the scan angles are referenced. The simplest structures assume a reflector that is pointing forwards in the direction of the y-axis. This is illustrated in figure 3-1, together with the definitions of elevation scan angle θ and azimuth angle ϕ. There are commands, however, to read in surfaces defined by an arbitrary list of coordinates and which change the orientation and method of parameterising the surface.

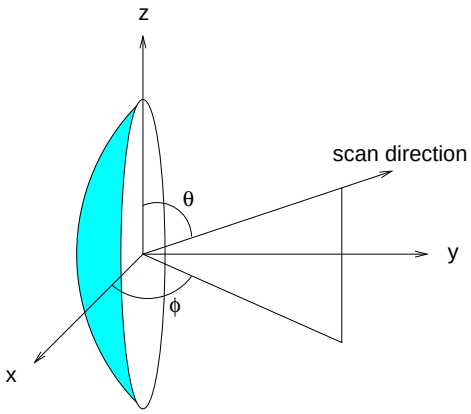

Figure 3-1: Main reflector global coordinate system

There also exists a separate coordinate system which defines how the feed is orientated. This is not relevant to an incident plane wave, which is specified in the global coordinate system.

The feed coordinate system is specified by the coordinates x', y' and z' such that the feed aperture is normal to the local z' axis. The nominal phase centre of the feed is assumed to

lie on the origin of the primed coordinate system. The orientation of the feed coordinate system with respect to the global coordinate system is controlled by user-specified Euler angles, θ_f, ϕ_f and ψ_f. This is illustrated in figure 3-2. The illustration of the Euler angles

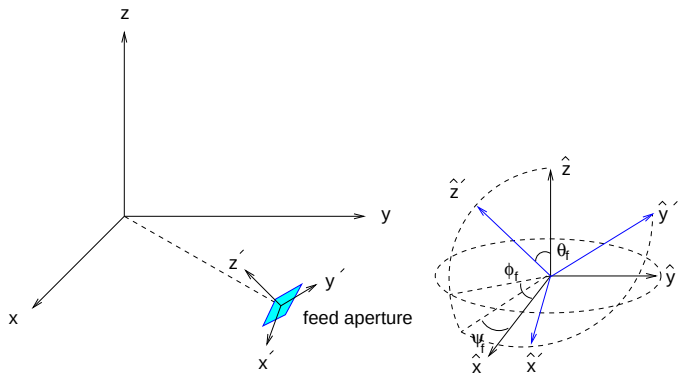

Figure 3-2: The feed coordinate system and Euler angles

is a little difficult; we define them as in [2]. If $\underline{\hat{x}}$, $\underline{\hat{y}}$ and $\underline{\hat{z}}$ are unit vectors in direction of increasing x, y and z and similarly, $\underline{\hat{x}}'$, $\underline{\hat{y}}'$ and $\underline{\hat{z}}'$ are unit vectors in direction of increasing x', y' and z', we have

$$\begin{aligned} \underline{\hat{x}}' &= R_{11}\underline{\hat{x}} + R_{12}\underline{\hat{y}} + R_{13}\underline{\hat{z}} \\ \underline{\hat{y}}' &= R_{21}\underline{\hat{x}} + R_{22}\underline{\hat{y}} + R_{23}\underline{\hat{z}} \\ \underline{\hat{z}}' &= R_{31}\underline{\hat{x}} + R_{32}\underline{\hat{y}} + R_{33}\underline{\hat{z}} \end{aligned} \tag{3-1}$$

where R_{ij} are the terms for i_{th} row and j_{th} column of the Euler rotation matrix $\underline{\underline{R}}$ defined as in [2] by,

$$\underline{\underline{R}} = \begin{bmatrix} -\sin\phi_f \sin\psi_f + \cos\theta_f \cos\phi_f \cos\psi_f & \cos\phi_f \sin\psi_f + \cos\theta_f \sin\phi_f \cos\psi_f & -\sin\theta_f \cos\psi_f \\ -\sin\phi_f \cos\psi_f - \cos\theta_f \cos\phi_f \sin\psi_f & \cos\phi_f \cos\psi_f - \cos\theta_f \sin\phi_f \sin\psi_f & \sin\theta_f \sin\psi_f \\ \sin\theta_f \cos\phi_f & \sin\theta_f \sin\phi_f & \cos\theta_f \end{bmatrix} \tag{3-2}$$

If a point in the global coordinate system is specified by $\underline{r} = (x, y, z)$ and in the feed coordinate system by $\underline{r}' = (x', y', z')$ then the transformation between the two is given by,

$$\begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix} = \underline{\underline{R}} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{3-3}$$

where $\underline{r}_0 = (x_0, y_0, z_0)$ is the position of the phase centre of the feed.

A typical script file might take the form;

```
CALCOPTS 1
ANGLES  90.0, 0.0, 1, -0.0, 0.25, 361
FREQS 77000.0 0.0 1
FILENAME test_out junk_out
FEEDCEN  0.0 0.15 0.0
%%FEEDROT  0.000 0.000  0.0
%%RECTHORN 0.012 0.012 0.0300 0.0300  Y  0.10
PLANEWAVE 90.0 -45.0 0.0 0.0
PLOTFILE junk.ps 90.00  45.0 180.0  0.4 0.2 0.0 T F F F
SURFACE SPLINE1D spline1.dat
BOUNDARY ellipse 0.106, 0.075, 0.0 0.00 00.0 0.001
GEOMFILE qqq_pj.dat  RW
```

where we have assumed illumination by a plane wave. The instructions to define a rectangular feed horn and its Euler rotation angles have been commented out. A definition of all the instructions now follows.

3.2 The *CALCOPTS* key word

Description

Defines one of several types of calculation options in the method of evaluation of the incident fields at the reflector. As of version 0.1.012) there are just two options. Option 1 uses both the incident electric and magnetic fields at each point on the reflector to determine the apparent direction of incidence using the Poynting vector method as described earlier. Option 2 uses only the incident electric fields together with a knowledge of the phase centre (or in the case of a plane wave, the defined direction of incidence) to determine the direction of incidence. The list below summarises its use.

Number of parameters	1
Key word requirement	Optional. May occur anywhere in a composite. If not present, Option 1 is assumed.
Number of occurrences	At most one.
File location	Anywhere within input file.
Parameter #1	Integer. Its value is 1 for option 1 and 2 for option 2.

3.3 The *FILENAME* key word

Description

Defines the name of the main output files. As of version 0.1.013 this takes two file names, though only the first is used. (The second is reserved for further software enhancements). The first file name is the name of the output file containing the predicted far-field gains.

The list below summarises its use.

Number of parameters	2
Key word requirement	Mandatory.
Number of occurrences	One.
File location	Anywhere within input file.
Parameter #1	Character string containing no spaces or separators. Name of output gain file.
Parameter #2	Character string containing no spaces or separators. Currently unused.

3.4 The *PLOTFILE* key word

Description

Defines the name of optional output POSTSCRIPT files showing the reflector and feed. Intended for diagnostic purposes. There may be any number of these. The parameters control the Euler view angles and magnification as well as some logical flags to switch on and off various features.

The list below summarises its use.

Number of parameters	11
Key word requirement	Optional.
Number of occurrences	Any number.
File location	Anywhere within input file.
Parameter #1	Character string containing no spaces or separators. Name of output POSTSCRIPT file.
Parameter #2	Real number, θ_v. The $\theta-$ view rotation angle in global coordinate system, specified in degrees.
Parameter #3	Real number, ϕ_v. The $\phi-$ view rotation angle in global coordinate system, specified in degrees.
Parameter #4	Real number, ψ_v. The $\psi-$ view rotation angle in global coordinate system, specified in degrees.
Parameter #5	Real number, magnification (scale) factor.
Parameter #6	Real number, x-coordinate display offset in meters.
Parameter #7	Real number, y-coordinate display offset in meters.
Parameter #8	Logical 'T' or 'F'. If 'T' displays reflector.
Parameter #9	Logical 'T' or 'F'. If 'T' displays shadow structure (when there is a blockage).
Parameter #10	Logical 'T' or 'F'. If 'T' displays incident wave directions on each facet of reflector.
Parameter #11	Logical 'T' or 'F'. If 'T' displays reflected wave directions on each facet of reflector.

3.5 The *ROTSURF* key word

Description

This is optional. If it occurs it can occur only once. It is used to define a rigid rotation of the main reflector about a specified point while keeping the feed orientation unchanged. If the new coordinates of the reflector are designated by the vector $\underline{x}'$ expressed by the coordinate nodes $\underline{x}'_i$ on the reflector and the original coordinates by the vector $\underline{x}$, then we define

$$\underline{x}' = \underline{x}_c + \underline{\underline{R}}(\underline{x} - \underline{x}_c) \tag{3-4}$$

where $\underline{x}_c$ is the user specified centre of rotation and $\underline{\underline{R}}$ is the Euler rotation matrix as defined in (3-2). This is specified by the three angles θ_c, ϕ_c amd ψ_c. For example, if $\phi_c = \psi_c = 0^o$ then the angle θ_c describes a rotation about a line parallel to the $y-$axis. If $\theta_c = \psi_c = 0^o$ then the angle ϕ_c describes a rotation about a line parallel to the $z-$axis and if $\phi_c = -90^o$, $\psi = 90^o$ then θ_c describes a rotation about a line parallel to the $x-$axis.

The list below summarises its use.

Number of parameters	6
Key word requirement	Optional.
Number of occurrences	At most one.
File location	Anywhere within input file.
Parameter #1	Real number, θ_c. The θ rotation angle, specified in degrees.
Parameter #2	Real number, ϕ_c. The ϕ rotation angle, specified in degrees.
Parameter #3	Real number, ψ_c. The ψ rotation angle, specified in degrees.
Parameter #4	Real number, x-coordinate of centre of rotation $\underline{x}_c$, in metres.
Parameter #5	Real number, y-coordinate of centre of rotation $\underline{x}_c$, in meters.
Parameter #6	Real number, z-coordinate of centre of rotation $\underline{x}_c$, in meters.

3.6 The *ANGLES* key word

Description

This is mandatory and defines the angular range at which the far-field gain is required. This defines a uniformly sampled block of solid angle defined by a range of θ and ϕ.

The list below summarises its use.

Number of parameters	6
Key word requirement	Mandatory.
Number of occurrences	1.
File location	Anywhere within input file.
Parameter #1	Real number. Defines the first value of θ in degrees.
Parameter #2	Real number. Defines the θ increment in degrees.
Parameter #3	Integer. Defines the total number of θ values.
Parameter #4	Real number. Defines the first value of ϕ in degrees.
Parameter #5	Real number. Defines the ϕ increment in degrees.
Parameter #6	Integer. Defines the total number of ϕ values.

Note that we may set Parameter #3 and Parameter #6 to zero, in which case ANGLES does not specify any requested directions. Angles may also be requested using *ANGLECUT* any number of times (see below). If no angles are specified either by *ANGLES* or by *ANGLECUT* then the program will terminate with an error message.

3.7 The *ANGLECUT* key word

Description

This is optional. It may be used any number of times. It defines a single cut (angular sweep) about any great plane of the direction sphere. Data output from each *ANGLECUT* is appended sequentially to the output file in order of their appearance in the input file, after the data generated by the *ANGLES* definition. The output file (as of version 0.1.004 of the software, in which this tag is introduced) contains a field position 10 which outputs the scan angle ν. See the previous documentation concerning the definitions of angles here.

The list below summarises its use.

Number of parameters	5
Key word requirement	Optional.
Number of occurrences	Any number.
File location	Anywhere within input file.
Parameter #1	Real number, θ_0, defining the value for θ of the centre point of the cut in degrees.
Parameter #2	Real number, ϕ_0, defining the value for ϕ of the centre point of the cut in degrees.
Parameter #3	Real number, η_0, defining the angle of the cut in degrees, with respect to the direction $\underline{\hat{\theta}}_0$.
Parameter #4	Real number, $\Delta\nu$, defining the angular increment along the cut, in degrees.
Parameter #5	Integer. Defines the total number of increments such that the sweep is made, $-n\Delta\nu \leq \nu \leq n\Delta\nu$.

3.8 The *FREQS* key word

Description

This is mandatory and defines the frequency range for which predictions are required. Frequencies are entered in MHz.

The list below summarises its use.

Number of parameters	3
Key word requirement	Mandatory.
Number of occurrences	1.
File location	Anywhere within input file.
Parameter #1	Real number. Defines the first value of frequency in MHz.
Parameter #2	Real number. Defines the frequency increment in MHz.
Parameter #3	Integer. Defines the total number of frequency values.

3.9 The *FEEDCEN* key word

Description

This is mandatory and defines the phase centre origin of the excitation. For a plane wave this defines the reference point for which the phase is zero and defines a front/back reference to the reflector surface normal. For a *FEEDFILE* specification using near fields, this defines the centre point of the rectangular array of points at which the near field is provided. For other feeds it defines the phase centre of the feed.

The list below summarises its use.

Number of parameters	3
Key word requirement	Mandatory.
Number of occurrences	1.
File location	Anywhere within input file.
Parameter #1	Real number. Defines the x-coordinate of phase centre or near field centre point in metres in global coordinate system.
Parameter #2	Real number. Defines the y-coordinate of phase centre or near field centre point in metres in global coordinate system.
Parameter #3	Real number. Defines the z-coordinate of phase centre or near field centre point in metres in global coordinate system.

3.10 The *FEEDROT* key word

Description

This is mandatory if a real feed is used, such as a dipole or horn. It must not be present if an incident plane wave is employed in order to avoid possible confusion, since a plane wave is defined with respect to the global coordinate system. For a feed, this defines the Euler rotation angles of the feed, θ_f, ϕ_f and ψ_f. See earlier notes for a definition of these.

The list below summarises its use.

Number of parameters	3
Key word requirement	Special. Mandatory if a real feed is employed, otherwise must not be present.
Number of occurrences	1 or 0.
File location	Anywhere within input file.
Parameter #1	Real number. Defines the Euler rotation angle θ_f in degrees.
Parameter #2	Real number. Defines the Euler rotation angle ϕ_f in degrees.
Parameter #3	Real number. Defines the Euler rotation angle ψ_f in degrees.

3.11 The *PLANEWAVE* key word

Description

One of several methods to illuminate the reflector and is logically considered as a feed type. See section 2.4.1 for the full description. This defines an incident plane wave of arbitrary polarisation and ellipticity. One and only one feed type must be used. A plane wave is specified by an incoming wave along the direction specified by θ_p and ϕ_p, with a polarisation defined by the angles χ_α and χ_β. The former defines the angle of the principal axis of the polarisation ellipse with respect to the electric field vectors. When $\chi_\beta = 0$, this refers to the polarisation angle of a linearly polarised wave. When $\chi_\alpha = 45$ degrees, the two states of circular polarisation are defined when $\chi_\beta = \pm 90$ degrees. More generally, the plane wave is elliptically polarised with the electric vector amplitude taking the form,

$$\underline{E}_{inc} = \cos\chi_\alpha \, \underline{\hat{\theta}}_p + e^{j\chi_\beta} \sin\chi_\alpha \, \underline{\hat{\phi}}_p \tag{3-5}$$

When the *PLOTFILE* command is used, the direction of the plane wave is represented by an arrow through the phase centre point specified by *FEEDCEN*. The principal direction (shown as $\cos\chi_\alpha \, \underline{\hat{\theta}}_p + \sin\chi_\alpha \, \underline{\hat{\phi}}_p$) is also illustrated through the phase centre point by an un-arrowed line.

The list below summarises its use.

Number of parameters	5
Key word requirement	Special. Represents one of several types of 'feed'. One and only one 'feed' type must be present.
Number of occurrences	1 or 0.
File location	Anywhere within input file.
Parameter #1	Real number. Defines the incidence angle θ_p in degrees.
Parameter #2	Real number. Defines the incidence angle ϕ_p in degrees.
Parameter #3	Real number. Defines the polarisation angle χ_α in degrees.
Parameter #4	Real number. Defines the ellipticity angle χ_β in degrees.

3.12 The *DIPOLE* key word

Description

One of several methods to illuminate the reflector and is logically considered as a feed type. This defines an elementary perfect (Hertzian) electric dipole radiator, situated at the phase centre point defined by *FEEDCEN*. See section 2.4.2 for the full description. The electric dipole is assumed to lie in the $\underline{\hat{z}}'$ direction. As of version 0.1.013 of the software, this takes no parameters.

When the *PLOTFILE* command is used, the $\underline{\hat{z}}'$ direction of the dipole is represented by an arrow through the phase centre point specified by *FEEDCEN*.

The list below summarises its use.

Number of parameters	0
Key word requirement	Special. Represents one of several types of 'feed'. One and only one 'feed' type must be present.
Number of occurrences	1 or 0.
File location	Anywhere within input file.

3.13 The *MGAUSSIAN* key word

Description

This is a method to illuminate the reflector and is logically considered as a feed type. It defines a modified Gaussian beam intended to provide greater field accuracy when the beam width is large, for which the standard scalar paraxial theory is not properly valid. For small beam widths it is equivalent to a standard Gaussian beam.

The beam is specified in the local feed coordinate system, with beam axis the z' direction and with electric field polarised in the y' direction at the beam waste. See section 2.4.3 for the full description. It requires the use of the *FEEDCEN* and *FEEDROT* key words to define the local coordinate system. One parameter is required; the semi-angle θ_0 defining the far-field beam angle from axis to -8.69 dB points in the $x'z'$ plane.

When the *PLOTFILE* command is used, a symbol is generated as for the *DIPOLE* command but additionally four lines are shown illustrating the angle θ_0 in both the $x'z'$ and $y'z'$ planes.

The list below summarises its use.

Number of parameters	1
Key word requirement	Special. Represents one of several types of 'feed'. One and only one 'feed' type must be present.
Number of occurrences	1 or 0.
File location	Anywhere within input file.
Parameter #1	Positive real number. Defines the semi-angle θ_0 in degrees to the -8.69 dB points.

3.14 The *PGAUSSIAN* key word

Description

This is syntactically identical to the *MGAUSSIAN* key word. This defines a standard paraxial Gaussian beam which should be equivalent to the extended Gaussian formulation for small angles from beam axis. The parameter θ_0 is here related to the beam waste radius w_0 by the standard approximation, $w_0 = 2/(k_0\theta_0)$. The full formulation is given earlier for reference.

3.15 The *RECTHORN* key word

Description

One of several methods to illuminate the reflector and is logically considered as a feed type. This defines a rectangular aperture horn with a cosine electric field distribution on the aperture. The feed horn is defined by the widths of the aperture in the $\underline{\hat{x}}'$ and $\underline{\hat{y}}'$ directions, as well as offset phase distances that define the distance of the phase centre to the aperture associated with $\underline{\hat{x}}'$ and $\underline{\hat{y}}'$ directions. Also required is a meshing interval in fractions of a wavelength and a 'type' designator to define the manner of polarisation of the horn.

Since the phase centre of the source may not exist at a single point in space (i.e. if the offset phase distances are different) and since the local phase centre of the aperture field may not coincide with the apparent phase centre as seen by the reflector in the intermediate or far field of the horn, it is generally best to assume the physical location of the phase centre defined by *FEEDCEN* lies close to the centre of the aperture of the horn. The offset phase distances required here are principally required to accurately define the aperture field distribution from which the field at the reflector is calculated.

The meshing interval should typically lie between a value of 0.2 and 0.01, depending on the accuracy required and the computation time penalty. We have usually found that a sampling value of 0.1 is quite adequate.

The 'type' designator is a character 'X', 'Y', 'LCP' or 'RCP'. 'X' implies the electric field on the aperture is aligned in the $\underline{\hat{x}}'$ direction. 'Y' implies the electric field on the aperture is aligned in the $\underline{\hat{y}}'$ direction. 'LCP' and 'RCP' imply that the feed is left or right circularly polarised. Note that the main reflector reverses the sense of circularity of the polarisation of the radiated field.

It is assumed here that only a TE01 and a TE10 mode can be excited. When either one or the other is present (but not both), the 'type' is designated by 'Y' or 'X' respectively. The LCP and RCP types involve a suitably phased linear combination of both. Figure 3-3 shows the excitation type with respect to the local feed coordinate system.

When the *PLOTFILE* command is used, the $\underline{\hat{z}}'$ direction of the feed and its aperture are illustrated.

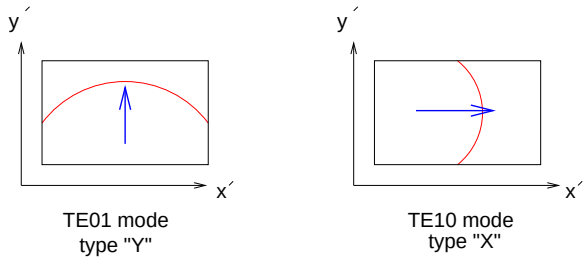

Figure 3-3: Aperture excitation types for a rectangular horn

The list below summarises its use.

Number of parameters	6 or 8
Key word requirement	Special. Represents one of several types of 'feed'. One and only one 'feed' type must be present.
Number of occurrences	1 or 0.
File location	Anywhere within input file.
Parameter #1	Real number. Defines the aperture width in x' direction in metres.
Parameter #2	Real number. Defines the aperture width in y' direction in metres.
Parameter #3	Real number. Defines the distance to the local phase centre for the TE01 (type 'Y') mode.
Parameter #4	Real number. Defines the distance to the local phase centre for the TE10 (type 'X') mode.
Parameter #5	Character string. Defines the 'type' designator, as described above.
Parameter #6	Real number. Defines the aperture sampling interval as a wavelength fraction.
Parameter #7	Character string. Optional. If present, defines the name of an output file defining the aperture electric field on the sampling grid.
Parameter #8	Character string. Optional. Must be present if parameter #7 is present. If present, defines the name of an output file defining the aperture magnetic field on the sampling grid.

Note that parameters #7 and #8 are optional. If present they should contain no separators and define the name of output files containing the transverse electric and magnetic fields on the aperture. The components of the field perpendicular to the aperture are defined as zero. This file is in the same format as a CST file [4] and may, for example, be used in tests of the *FEEDFILE* command.

3.16 The *GEOMFILE* key word

Description

This key word defines the name of a secondary file required by the program and is mandatory. There are two uses of this command. The first parameter defines the name of a "points and joins" file. The second defines a permission and the mode of operation. This takes the form of a two character string; either 'RO' or 'RW'. In 'RO' mode, the defined file can only be read. In 'RW' mode, the designated file can be read and written to.

The effect of this command depends on whether the *SURFACE* and *BOUNDARY* commands are present. *SURFACE* and *BOUNDARY* commands must either both be present or neither must be present. If both are present the points-and-joins file will be written to if 'RW' is designated or there is no effect if 'RO' is designated (i.e. this command is ignored). In both cases, the reflector geometry is defined by the *SURFACE* and *BOUNDARY* commands and not the points-and-joins file.

If neither *SURFACE* or *BOUNDARY* commands are present, the software takes the reflector geometry defined by the points-and-joins file. In this case the permission must be set to 'RO'.

The required format of the points and joins file is shown in the following example (note that the node coordinates are specified in metres):

```
Number of nodes:
16
Node coordinates:
   1     0.80680E-01    0.99296E-04    0.48644E-01
   2     0.34082E-01    0.69546E-04    0.71018E-01
   3    -0.20870E-01    0.24396E-04    0.73532E-01
   4    -0.79174E-01    0.13780E-02    0.49868E-01
   5    -0.10600E+00    0.30000E-02    0.41399E-05
   6    -0.80663E-01    0.15101E-02   -0.48659E-01
   7    -0.31787E-01    0.19570E-03   -0.71548E-01
   8     0.25243E-01   -0.46067E-05   -0.72842E-01
   9     0.80294E-01    0.10700E-03   -0.48964E-01
  10     0.10600E+00    0.22617E-03    0.43846E-13
  11     0.15782E-01    0.48883E-04    0.30572E-01
  12    -0.35072E-01    0.80834E-04    0.21060E-01
  13    -0.34648E-01    0.88332E-04   -0.21409E-01
  14     0.15777E-01    0.48921E-04   -0.30573E-01
  15     0.46000E-01   -0.45332E-05    0.19028E-13
  16     0.00000E+00    0.41190E-04    0.00000E+00
Number of facet elements:
```

```
20
Element reference list:
   1       1       2      11
   2       2       3      11
   3       3       4      12
   4       4       5      12
   5       5       6      13
   6       6       7      13
   7       7       8      14
   8       8       9      14
   9       9      10      15
  10      10       1      15
  11      11      12       3
  12      12      13       5
  13      13      14       7
  14      14      15       9
  15      15      11       1
  16      11      12      16
  17      12      13      16
  18      13      14      16
  19      14      15      16
  20      15      11      16
```

The first list defines the node coordinates of the reflector in metres and assigns a node number to each point. The second list defines the triangles by reference to the node such that each triangle is defined by three nodes.

The list below summarises the command use.

Number of parameters	2
Key word requirement	Mandatory.
Number of occurrences	1
File location	Anywhere within input file.
Parameter #1	Character string containing no separators. Defines the name of the points and joins file.
Parameter #2	Character string. Defines the permission. This must be either 'RO' or 'RW'.

3.17 The *SURFACE* key word

3.17.1 Introduction to surface definitions

Within the formulation the reflector geometry is described by a set of piece-wise flat triangular facets (triangles) which are defined by a three-set of nodal coordinates (nodes). The definition of triangles and nodes is controlled by the BOUNDARY key word with each node residing on the surface. In general a surface may be defined mathematically by a three dimensional vector function of two parameters, $\underline{S}(s_x, s_z)$. The reason for the parameter nomenclature is described below. For our purposes we may limit this definition by further imposing that the parameters s_x and s_z form a compact domain of finite extent. So, for example, s_x and s_z may represent the interior points of an ellipse in (s_x, s_z) space or the interior points of a rectangle $L_1 < s_x < L_2$ and $L_3 < s_z < L_4$.

In early version of the software (up to version 0.1.011), it was sufficient to define $s_x = x$ and $s_z = z$ where x and z were global coordinates and the surface $\underline{S}(s_x, s_z)$ could then be defined by, $\underline{S}(s_x, s_z) = (x, Y(x, z), z)$ where $Y(x, z)$ was an arbitrary single-valued function of x and z, where $s_x = x$ and $s_z = z$. The BOUNDARY key word specified the compact domain of x and z. This is sufficient to define a section of paraboloid and several distortions for shaped reflectors. However, it is not general enough for the specification of offset shaped reflectors so we require a slightly more general notation and formulation. This more advanced characterisation is in place for versions $\geq$ 0.1.012 of the software.

Since a surface is usually parabolic or nearly parabolic, for which we employ $s_x = x$ and $s_z = z$, it makes sense to consider s_x as 'x-like' and s_z as 'z-like', hence the notation. This allows us to maintain the BOUNDARY definitions describing a rectangle or elliptical cut-out. However, formally the BOUNDARY definition defines a compact space on (s_x, s_z) and not on (x, z). For a slightly distorted surface the (x, z) profile will be fairly similar to the (s_x, s_z) one, even when $z(y)$ is not a single-valued function.

The SURFACE types PLANE, PARABOLOID, PARABDISTORT, VARIPARAB1, VARIPARAB2, VARIPARAB3, VARIPAR2D1 and SPLINE1D assume a surface defined by,

$$\underline{S}(s_x, s_z) = (X, Y(s_x, s_z), Z) \tag{3-6}$$

If the new key word DISTORTSURF (see section 3.19) is not present then we take $X = s_x$ and $Z = s_z$.

The key word DISTORTSURF permits a more general distortion such that,

$$\underline{S}(s_x, s_z) = (X(s_x, s_z), Y(s_x, s_z), Z(s_x, s_z)) \tag{3-7}$$

Thus the SURFACE command defines the function $Y(s_x, s_z)$ and the optional DISTORTSURF command defines the functions $X(s_x, s_z)$ and $Z(s_x, s_z)$. Currently (as of version 0.1.013 of the software), the DISTORTSURF options are more specialised and the surface is described by;

$$\underline{S}(s_x, s_z) = (s_x, Y(s_x, s_z), Z(s_z)) \tag{3-8}$$

In summary, a surface is now defined by the three key words, SURFACE, BOUNDARY and DISTORTSURF where DISTORTSURF is optional.

3.17.2 Description and definition of parameters

The SURFACE key word is mandatory if the reflector geometry is not defined by the GEOMFILE command. If present, it must also be accompanied by the BOUNDARY command. The first parameter defines the type of surface. Subsequent parameters depend on the type of surface.

Currently (version 0.1.013) the permitted surfaces are; a plane surface PLANE, a parabolic surface PARABOLOID, a distorted parabolic surface PARABDISTORT, shaped parabolic surfaces VARIPARAB1, VARIPARAB2, VARIPARAB3, VARIPAR2D1 and an extruded one-dimension cubic spline surface SPLINE1D. The parametric boundary of all these surfaces is defined by the BOUNDARY command.

The PLANE surface

This surface is a simple flat mirror whose normal is defined by the vector $\underline{n}_r = (n_x, n_y, n_z)$ and a point $\underline{p} = (p_x, p_y, p_z)$ in the plane. The plane is defined in the (s_x, s_y, s_z) coordinate system with the plane defined by the surface vector $(s_x, Y(s_x, s_z), s_z)$, i.e. where $Y(s_x, s_z) = s_y$. Use of subsequent distortion (i.e. the DISTORTSURF command) can be used to generate a distorted plate as defined in (3-6).

The PARABOLOID surface

This surface is a paraboloid of revolution. It is defined by its focal length, f_l, and the coordinates of the focal point, $\underline{r}_f = (x_f, y_f, z_f)$. It is assumed that the paraboloid is always orientated such that its axis of revolution is parallel to the y-axis. Mathematically the reflector surface is defined by,

$$Y(s_x, s_z) = y_f - y_p(s_x, s_z) \tag{3-9}$$

where

$$y_p(s_x, s_z) = f_l - \frac{r^2}{4f_l} \tag{3-10}$$

and

$$r = \sqrt{(s_x - x_f)^2 + (s_z - z_f)^2} \tag{3-11}$$

We have had occasion to use an alternative representation with

$$y_p(s_x, s_z) = r\cot(\Theta) \equiv \frac{r\cos\Theta}{\sqrt{1-\cos^2\Theta}} \quad \text{where} \quad \cos\Theta = \frac{4f_l^2 - r^2}{4f_l^2 + r^2} \tag{3-12}$$

In the latter representation special care is required for small $r/f_l \ll 1$ where we should revert to (3-10). Note that $Y(s_x, s_z) = 0$ if $y_f = f_l$ when $s_x = s_z = 0$.

The PARABDISTORT surface

This surface is a paraboloid of revolution, as above, with an additive deviation function designated by a type (specified by an integer) and a file name containing the deviation parameters associated with the type. See section 3.17.3.

The VARIPARAB1 surface

This surface represents one model of a distorted paraboloid of revolution intended to permit a profiled beam in one angular direction (e.g. an approximation to a $cosec^2\theta$ distribution in the θ direction) while maintaining a narrow beam in the orthogonal ϕ direction. There are any number of ways in which this can be done. Here we will assume that the reflector surface function $Y(s_x, s_z)$ is given by,

$$Y(s_x, s_z) = y_f - A(s_z)y_p(s_x, s_z) \tag{3-13}$$

$y_p(s_x, s_z)$ is a paraboloid of revolution about the y axis, as defined by the PARABOLOID option. The function $A(s_z)$ defines the distortion. We aim for a function such that $\alpha(s_z)$ is close to one for $s_z > 0$, with progressive deviation for smaller s_z. We choose,

$$A(s_z) = 1 - \Big(\alpha_0 + \sum_{i=1}^{N}[\alpha_i(s_z - z_c)]^i\Big)e^{-\beta(s_z - z_c)} \tag{3-14}$$

where α_0 is dimensionless and β, α_1, $..\alpha_n$ have units of m^{-1}. We will assume a total of N such α_i parameters. z_c is an offset distance with units of m, usually required non-zero for offset shaped reflectors.

The VARIPARAB2 surface

This surface represents an alternative distorted paraboloid of revolution intended to permit a profiled beam in one angular direction (e.g. an approximation to a $cosec^2\theta$ distribution in the θ direction) while maintaining a narrow beam in the orthogonal ϕ direction. As above we will assume that the reflector surface function $Y(s_x, s_z)$ is given by,

$$Y(s_x, s_z) = y_f - A(s_z)y_p(s_x, s_z) \tag{3-15}$$

where $y_p(s_x, s_z)$ is a paraboloid of revolution about the $y-$axis, as defined by the PARABOLOID option. The function $A(s_z)$ is now defined by,

$$A(s_z) = 1 + \left(\frac{1 + \tanh\big(-\beta(s_z - z_c)\big)}{2}\right)\big(\alpha_0 + \sum_{i=1}^{N}[\alpha_i(s_z - z_c)]^i\big) \tag{3-16}$$

where α_0 is dimensionless and β, α_1, $..\alpha_n$ have units of m^{-1}. We will assume a total of N such α_i parameters. z_c is an offset distance with units of m, usually required non-zero for offset shaped reflectors.

The use of the hyperbolic tangent provides a better transition function than the exponential used in (3-14) since for $s_z \ll 0$ the distortion is given by a standard power series and not one with exponential weight. For modelling a far-field *cosec*^{2} distribution, we find experimentally that β should be large and positive, sufficient to give a rapid transition from constancy to the power series over a scale of a few wavelengths. We also find the odd order α_i coefficients should be significant and positive whilst the even order α_i coefficients should be ether zero or small and negative.

For example, consider the following input file, with $\beta = 11.0\ m^{-1}$, $\alpha_0 = \alpha_2 = \alpha_4 = 0$, $\alpha_1 = 0.08\ m^{-1}$, $\alpha_3 = 2.20\ m^{-1}$ and $\alpha_5 = 1.80\ m^{-1}$:

```
CALCOPTS 1
ANGLES 20.0, 0.3, 450, 90.0, 0.0, 1
FREQS 20000.0 0.0 1
FILENAME test_out junk_out
FEEDCEN  0.0 0.58 0.0
FEEDROT  95.000  -90.000  00.0
MGAUSSIAN 30.0
PLOTFILE sideview.ps 90.00  00.0 180.0  0.50 0.1 0.0 T F F F
SURFACE VARIPARAB2 0.60 0.0 0.60 0.0 0.0 5 11.000 0.00 0.08 -0.00 2.2 -0.00  1.80
BOUNDARY ellipse 0.400, 0.32, 0.0 0.02 00.0 0.015
GEOMFILE qqq_pj.dat  RW
```

The parameters are not optimised. The predicted far field is illustrated in figure 3-4, together with an example of an idealised cosec-squared distribution defined by,

$$P_{db}(f) = 40.0 + 10.0\ \log_{10}\left(\frac{\sin^2(\pi/2 - \theta_1)}{\sin^2(\theta - \theta_1)}\right) \tag{3-17}$$

where $\theta_1 = 92^o \approx 1.6057$. A side view of the reflector, as defined in the PLOTFILE command is illustrated in figure 3-5.

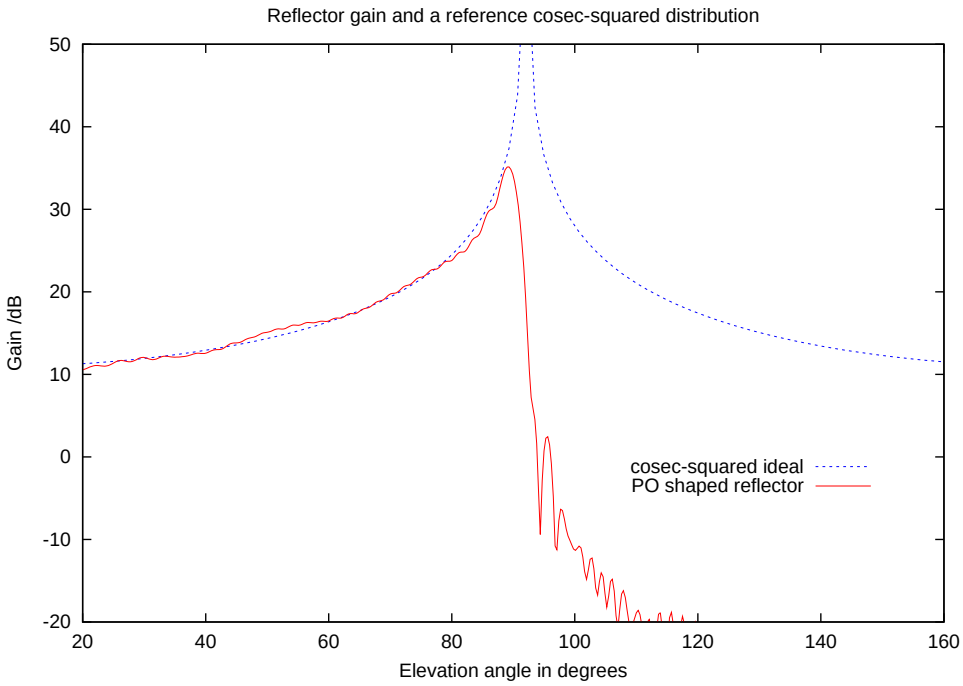

Figure 3-4: An ideal cosec-squared distribution and a far-field match using a shaping function.

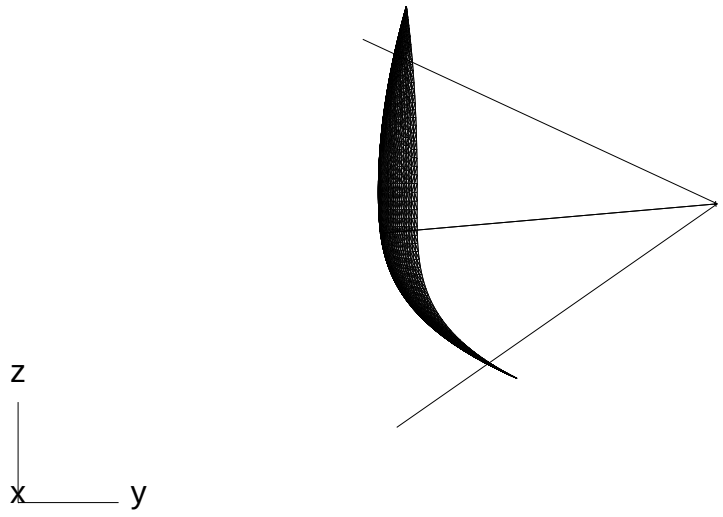

Figure 3-5: A profile plot of the reflector shape and Gaussian beam origin and angle lines

The VARIPARAB3 surface

This surface represents an alternative distorted paraboloid of revolution where the distortion is additive rather than applied as a multiplicative factor. This permits better control of the distortion for some applications. We assume here,

$$Y(s_x, s_z) = y_f - y_p(s_x, s_z) + A(s_z) \tag{3-18}$$

where $y_p(s_x, s_z)$ is a paraboloid of revolution about the $y-$axis, as defined by the PARABOLOID option and the function $A(s_z)$ is now defined by,

$$A(s_z) = f_l \left(\frac{1 + \tanh\big(-\beta(s_z - z_c)\big)}{2} \right) \big(\alpha_0 + \sum_{i=1}^{N} \alpha_i (s_z - z_c)^i\big) \tag{3-19}$$

The f_l focal length pre-multiplier is employed and we have changed the power series definition. Here, α_i have dimensions of $metres^{-i}$. The VARIPARAB2 definition implicitly forces the coefficients of even powers of $(s_z - z_c)$ to be positive, which is not always required.

The VARIPAR2D1 surface

This surface represents an alternative distorted paraboloid with shaping as a function of both s_x and s_z. This is intended to allow symmetric shaping of the beam in the $x - y$ plane and non-symmetric (e.g. a cosec-squared distribution) in the $z - y$ plane. This method has found use for prime-focus (non-offset) reflectors, but does not perform well for offset reflectors. For design of the latter we recommend use of the VARIPAR2D2 surface type, as described below.

We will assume that the reflector surface function $Y(s_x, s_z)$ is given by,

$$Y(s_x, s_z) = y_f - P(s_z)A(s_z)B(s_x) \tag{3-20}$$

where $P(s_z)$ is parabolic as a function only of s_z,

$$P(s_z) = y_f - \frac{(s_z - z_f)^2}{4f_l} \tag{3-21}$$

$A(s_z)$ is defined as in (3-16) for legacy reasons[4] with,

$$A(s_z) = 1 + \left(\frac{1 + \tanh\big(-\beta(s_z - z_c)\big)}{2} \right) \Big(\alpha_0 + \sum_{i=1}^{N} [\alpha_i (s_z - z_c)]^i\Big) \tag{3-22}$$

and $B(s_x)$ is a symmetric function of harmonic form,

$$B(s_x) = 1 + \sum_{i=1}^{n} a_i \left(1 - \cos \frac{i\pi(s_x - x_c)}{x_M}\right) \tag{3-23}$$

[4] The revised power series was introduced after numerous examples were made using the older definition.

We specify n dimensionless coefficients a_1 to a_n together with a scaling factor x_M with units of distance. The parameters x_c and z_c are coordinate offsets for the shaping functions, with units of m. These offsets are usually required to be non-zero for offset shaped reflectors, though this surface type is not recommended if significant offsets are required.

For example, consider the following input file, with $\beta = 8.0\ m^{-1}$, $\alpha_0 = \alpha_2 = \alpha_4 = 0$, $\alpha_1 = 0.05\ m^{-1}$, $\alpha_3 = 0.70\ m^{-1}$ and $\alpha_5 = 1.80\ m^{-1}$ and the secondary coefficients $x_M = 1.0$ m, $a_1 = 0.06$ and $a_2 = 0.02$:

```
CALCOPTS 1
ANGLES  90.0, 0.0, 1, 00.0, 0.5, 900
%%ANGLES 00.0, 0.3, 450, 90.0, 0.0, 1
FREQS 20000.0 0.0 1
FILENAME test_out junk_out
FEEDCEN  0.0 0.58 0.0
FEEDROT  90.000  -90.000  00.0
MGAUSSIAN 35.0
%%PLOTFILE sideview1.ps 00.00  00.0 180.0  0.50 0.1 0.0 T F F F
PLOTFILE sideview2.ps 90.00  00.0 180.0  0.50 0.1 0.0 T F F F
SURFACE VARIPAR2D1 0.60 0.0 0.60 0.0 0.0 5 8.0 0.00 0.05 -0.00 0.7 -0.00 1.80 2 1.00 0.06 0.02
BOUNDARY ellipse 0.500, 0.34, 0.0 0.02 00.0 0.015
GEOMFILE qqq_pj.dat  RW
```

The parameters are not optimised. The predicted far field is illustrated in the two principal planes. Figure 3-6 shows an elevation cut (in the $y-z$ plane) and 3-7 shows an azimuth cut (in the $y-x$ plane). Figure 3-6 also shows an example of an idealised cosec-squared distribution defined by,

$$P_{db}(f) = 30.0 + 10.0\ \log_{10}\left(\frac{\sin^2(\pi/2-\theta_1)}{\sin^2(\theta-\theta_1)}\right) \tag{3-24}$$

where $\theta_1 = 91^o \approx 1.5882$. Side views of the reflector, as defined in the PLOTFILE command (one commented out) are illustrated in figures 3-8 and 3-9.

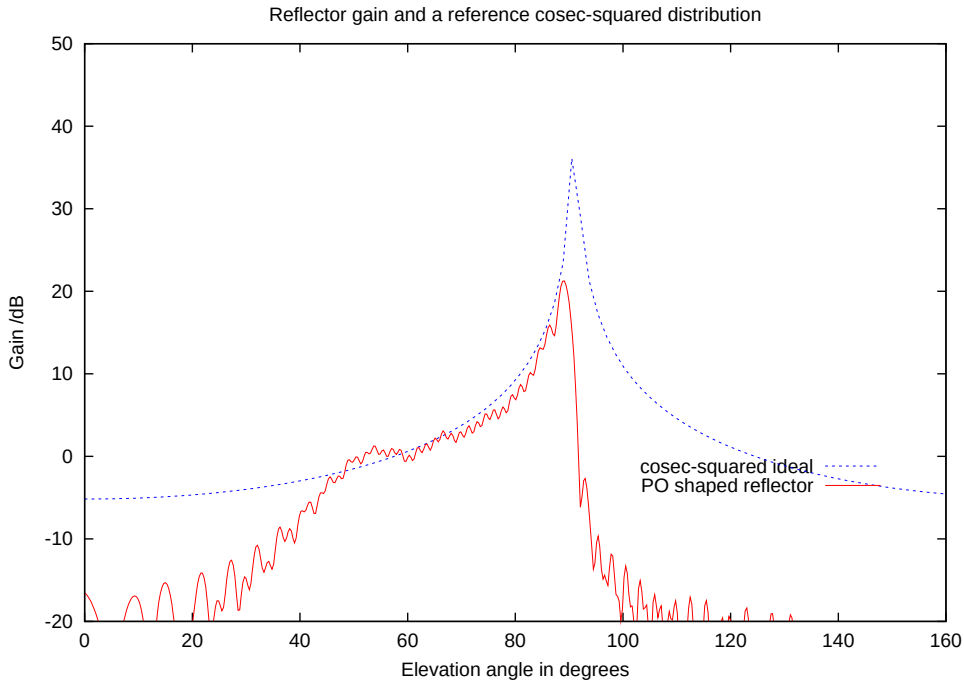

Figure 3-6: An ideal cosec-squared distribution and a far-field match using a shaping function.

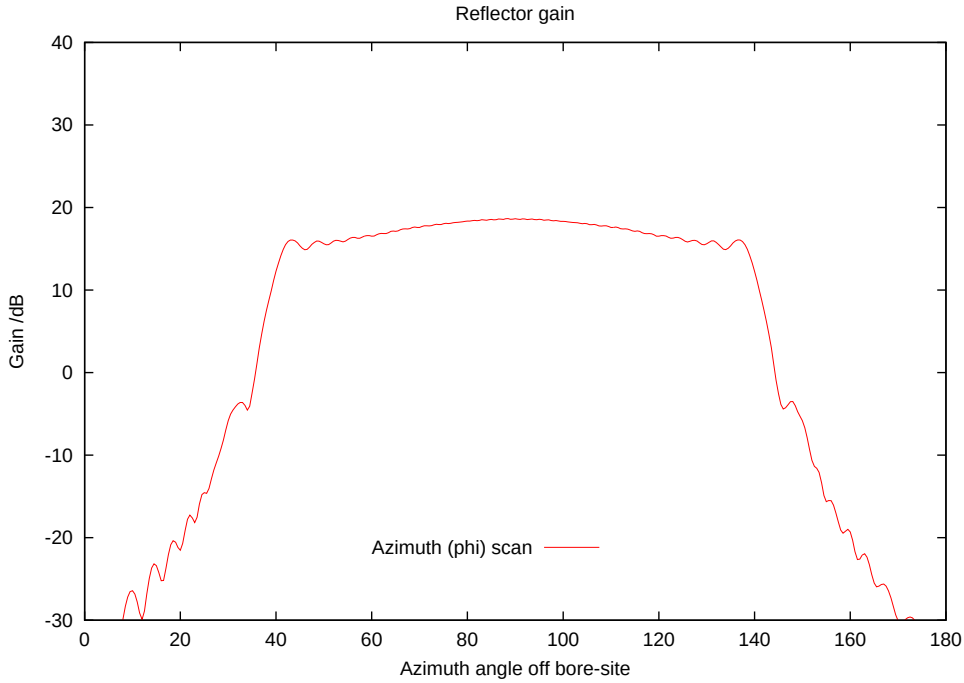

Figure 3-7: A near-rectangular azimuth distribution using a shaping function.

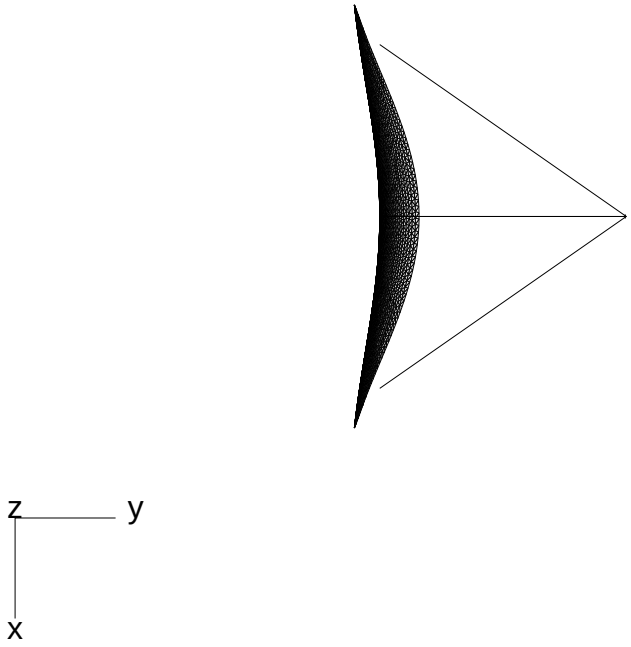

Figure 3-8: A profile plot of the reflector shape and Gaussian beam origin and angle lines

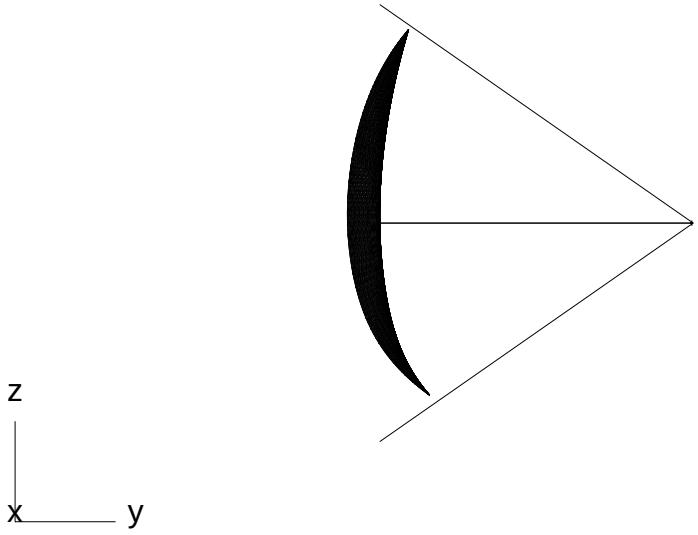

Figure 3-9: A second profile plot of the reflector shape and Gaussian beam origin and angle lines

The VARIPAR2D2 surface

This surface represents a more advanced distorted paraboloid option suitable for the design of shaped offset reflectors. It is found that the VARIPAR2D1 surface does not do a good job at decoupling the azimuth and elevation pattern shapes when significant distortion is required for azimuth control. This can lead to the introduction of elevation sidelobes through changes made in the function $B(s_x)$ which should significantly effect only the shape of the azimuth beam cut. It is believed that this strong non-seperability is a result of the fact that B is independent of s_z, whereas the pitch of the modulations should probably be a function of s_z. The nature of the unwanted sidelobes suggests that there is more than one stationary phase ray-path between the feed phase centre and infinity, in the $s_x = 0$ plane.

To avoid this problem we postulate a function $B(s_x, s_z)$ such that the set of curves defined by $(s_x(p), z_z(p))$ parameterised by an arbitrary parameter p for which $B(s_x(p), s_z(p)) = K$, for a constant K, are given by hyperbolas which intersect normal to the s_x axis ($s_z = 0$). This means that for non-offset reflectors, $B(s_x, s_z)$ can have similar form to the VARIPAR2D1 surface, in the neighbourhood of the s_x axis and for strongly offset reflectors, for large s_z, the curves approach lines through the origin. We also take the opportunity to taper the amplitude of the distortion $B(s_x, s_z)$ as a function of s_z and make $B(s_x, s_z)$ additive rather than multiplicative. and make some other changes.

We define,

$$Y(s_x, s_z) = \left(\frac{s_z^2}{4f_l} + \frac{g_0\, f_l}{4} \big(\Theta(s_x, s_z)\big)^p \right) A(s_z) + B(s_x, s_z) \tag{3-25}$$

where g_0 and p are dimensionless constants ($p > 0$), $A(s_z)$ is defined by the revised power series,

$$A(s_z) = 1 + \left(\frac{1 + \tanh\big(-\beta(s_z - z_c)\big)}{2} \right) \sum_{i=1}^{N} \alpha_i (s_z - z_c)^i \tag{3-26}$$

where $N \geq 0$ is a non negative integer (if $N = 0$ then the sum is defined as zero) and $B(s_x, s_z)$ is defined by,

$$B(s_x, s_z) = (1 + c_0\, s_z) \sum_{i=1}^{n} a_i \left(1 - \cos\left(\frac{i\pi\Theta(s_x, s_z)}{\theta_M} \right) \right) \tag{3-27}$$

where c_0, a_i and θ_M are constants, $n \geq 1$ is a positive integer and

$$\Theta(s_x, s_z) = \arctan\left(\frac{s_x}{\sqrt{s_z^2 + b_0^2}} \right) \tag{3-28}$$

and b_0 is a constant. An explanation of the function $\Theta(s_x, s_z)$ is illustrated figure 3-10, below. Here, we draw an elliptical profile representing an offset reflector with x and z coordinates, $X = s_x$ and $Z = s_z$ and y-coordinate as defined in (3-25).

Along any hyperbola defined by,

$$s_x^2 = K^2(s_z^2 + b_0^2) \tag{3-29}$$

for constant K we have the following properties for $\Theta(s_x, s_z)$; $\Theta = \arctan(K)$ along any of the hyperbolas, in the limit as $s_z \to \infty$ then $\Theta \to \theta = \arctan(s_x/s_z)$ and when $s_z = 0$ then $\Theta = \arctan(s_x/b_0)$ which for large b_0 can be made (with suitable scaling of the parameter x_M) to have the same functional form as employed for the VARIPAR2D1 surface. Note that there is no scope for an offset described by the parameter x_c in the VARIPAR2D1 surface.

This form has been successfully employed for shaping an offset reflector. Currently we have found it satisfactory to set the parameter $c_0 = 0$. A large value of p with non zero g_0 may be used to bend the reflector near its edges for spill-over reduction. An example structure is defined by the input file:

```
CALCOPTS  1
ANGLES  90.0  0.0  1  00.0  0.5  360
%%ANGLES  00.0  0.3  600  90.0  0.0  1
FREQS  24000.0  000.0  1
FILENAME  test_out_azim  junk_out
FEEDCEN  0.0  0.147  -0.00
FEEDROT  25.000  -90.000  90.0
SURFACE  VARIPAR2D2  0.15  0.12  -0.0  0.0  0.19  0  -5.0  2.00  1  0.4  -0.01
MGAUSSIAN  15.0
PLOTFILE  sideview1.ps  00.00  00.0  180.0  0.50  0.1  0.0  T  F  F  F
PLOTFILE  sideview2.ps  90.00  00.0  180.0  0.50  0.1  0.0  T  F  F  F
PLOTFILE  sideview3.ps  90.00 -90.0  180.0  0.50  0.1  0.0  T  F  F  F
BOUNDARY  ellipse  0.25  0.15  0.0  0.18  00.0  0.007
GEOMFILE  qqq_pj.dat  RW
```

Syntax for the SURFACE VARIPAR2D2 option is described below. The three sideview plots are shown in figure 3-11 (illustrated using a target mesh control factor of 0.012 rather than the figure of 0.007 defined above for the predictions), which should be compared to figure 3-10. This represents a useful design (though unoptimised) for a constant gain in azimuth. Figures 3-12 and 3-13 show the predicted gains together with a target envelope. Note that the input file defined above is configured for the azimuth sweep. The commented and uncommented ANGLES tags should be reversed for the elevation sweep. There are no unwanted sidelobes in the elevation plane.

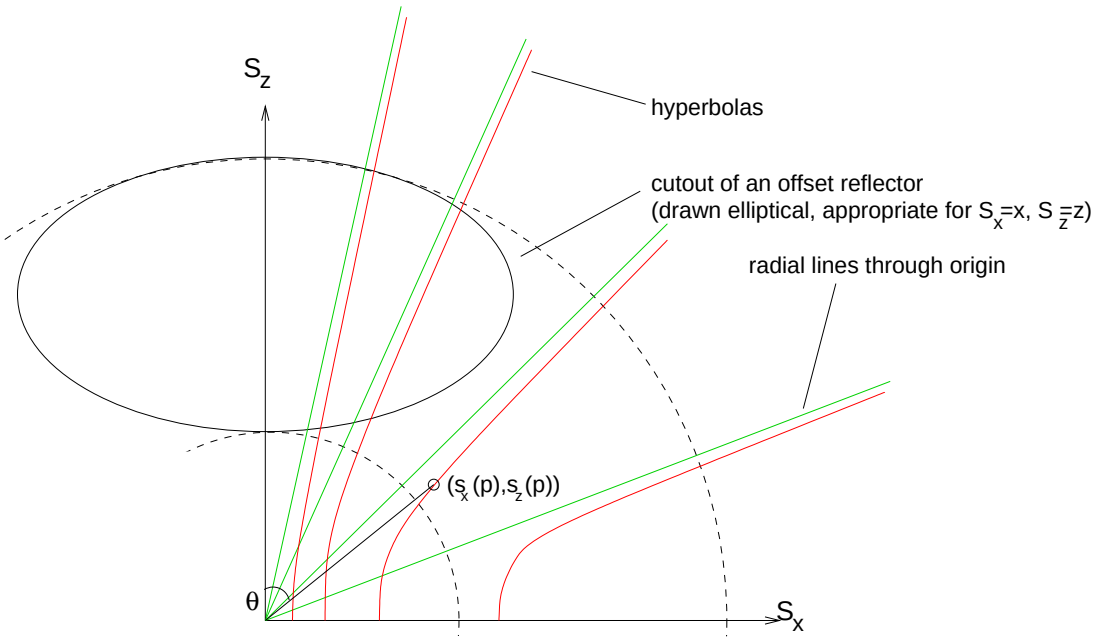

Figure 3-10: Hyperbolas of constant Θ

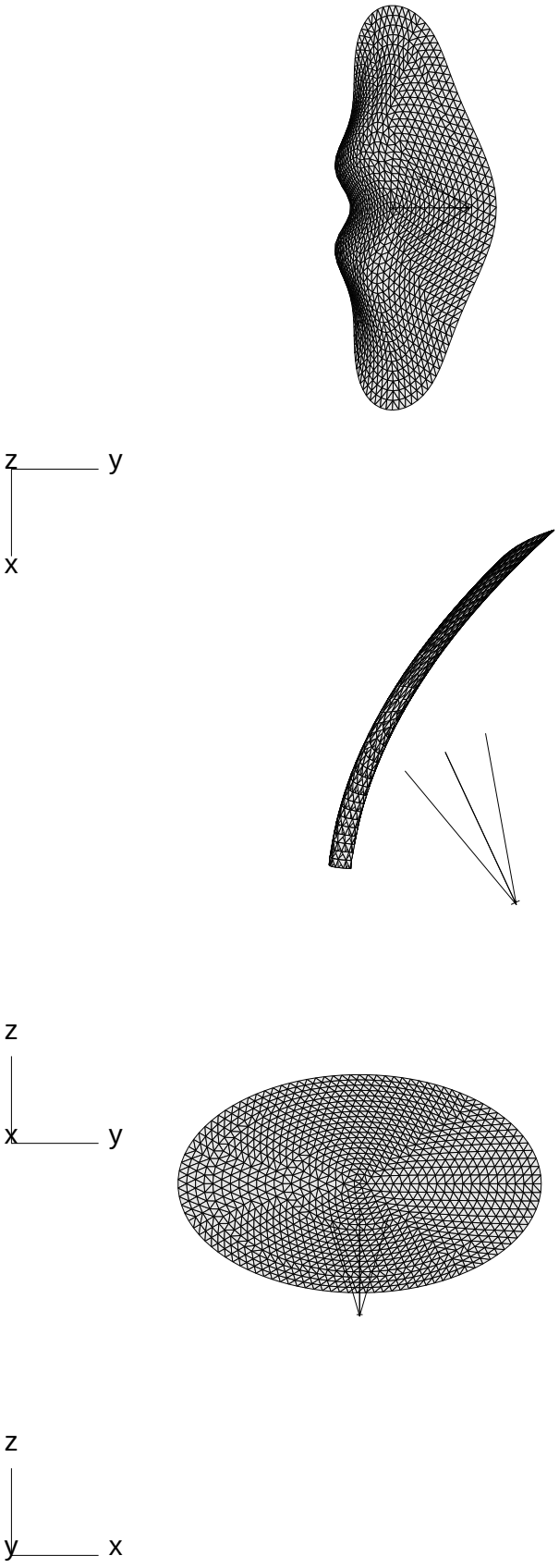

Figure 3-11: Profile plots of an offset shaped reflector as defined (top to bottom) by 'sideview1', 'sideview2' and 'sideview3'

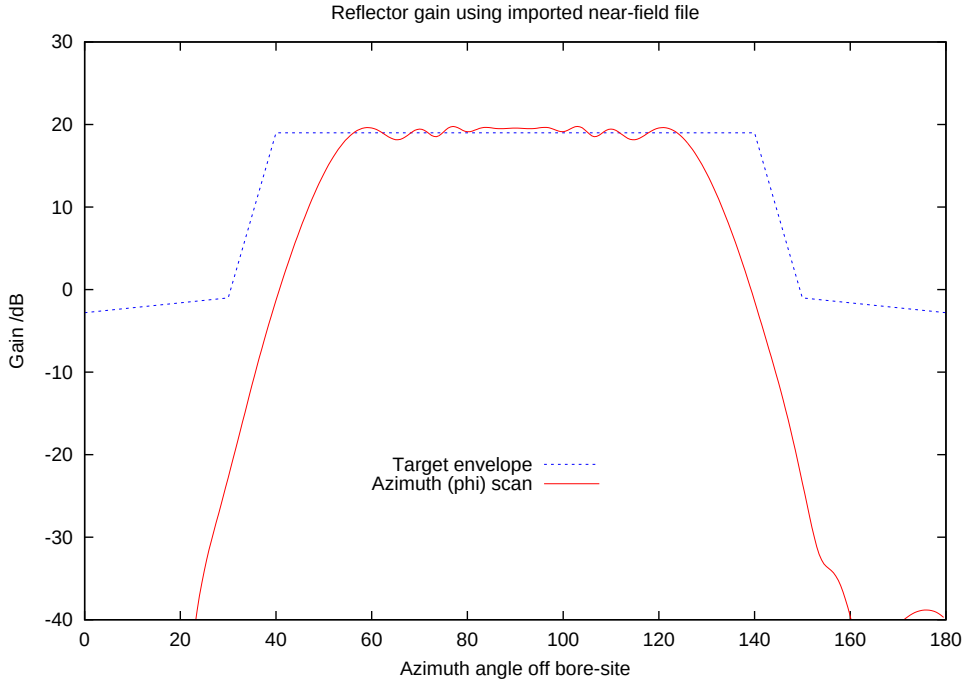

Figure 3-12: A target and predicted gain pattern in azimuth.

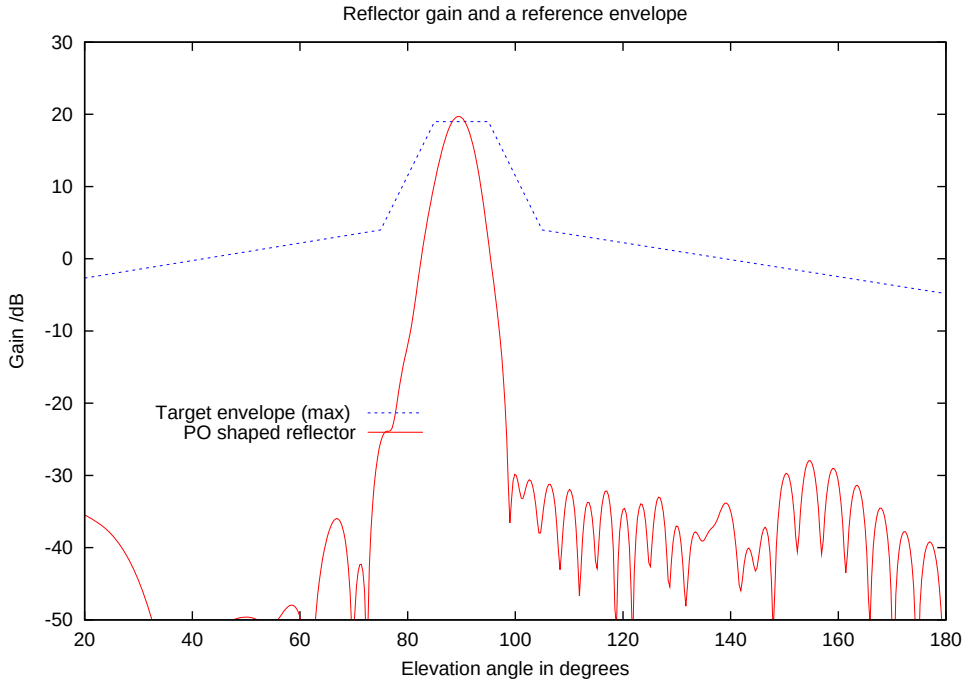

Figure 3-13: A target and predicted gain pattern in elevation.

The SPLINE1D surface

This surface assumes a general reflector surface given by,

$$Y(s_x, s_z) = C(s_x) \tag{3-30}$$

which is independent of s_z and where $C(s_x)$ is defined by a one-dimensional cubic spline function whose control points are provided by an auxiliary input file. We assume a 'natural' cubic spline with control points that must span the range of values required by the BOUNDARY defined boundary. The required input file name is a parameter, and takes the format (coordinates specified in metres) :

$itot$	
$s_x(1)$	$Y(1)$
$s_x(2)$	$Y(2)$
.	.
.	.
$s_x(itot)$	$Y(itot)$

where $s_x(1) < s_x(2) < \ldots < s_x(itot)$. The algorithm for defining $C(s_x)$ is given in pp119 of [3]. Such a surface may be regarded as an extrusion of a cubic curve along the s_z-axis, truncated by the BOUNDARY.

3.17.3 Syntax

Number of parameters	At least 2.
Key word requirement	Special. Mandatory if BOUNDARY command present.
Number of occurrences	1 or 0.
File location	Anywhere within input file.
Parameter #1	Character string. This defines the surface type and must be either PLANE, PARABOLOID, PARABDISTORT, VARIPARAB1, VARIPARAB2, VARIPARAB3, VARIPAR2D1 or SPLINE1D.

Further parameters depend on the surface type.

If Parameter #1 = PLANE then,

Parameter #2	Real number. Defines the s_x-component, n_x, of the surface normal in metres.
Parameter #3	Real number. Defines the s_y-component, n_y, of the surface normal in metres.
Parameter #4	Real number. Defines the s_z-component, n_z, of the surface normal in metres.
Parameter #5	Real number. Defines the s_x-component, p_x, of a surface point in metres.
Parameter #6	Real number. Defines the s_y-component, p_y, of a surface point in metres.
Parameter #7	Real number. Defines the s_z-component, p_z, of a surface point in metres.

If Parameter #1 = PARABOLOID then,

Parameter #2	Real number. Defines f_l, the focal length of paraboloid in metres.
Parameter #3	Real number. Defines x_f, the s_x-offset of the paraboloid in metres.
Parameter #4	Real number. Defines Y, the $Y-$offset of the paraboloid in metres.
Parameter #5	Real number. Defines z_f, the s_z-offset of the paraboloid in metres.

If Parameter #1 = PARABDISTORT then,

Parameter #2	Real number. Defines f_l, the focal length of parabola in metres.
Parameter #3	Real number. Defines x_f, the s_x-offset of the paraboloid in metres.
Parameter #4	Real number. Defines y_f, the $Y-$offset of the paraboloid in metres.
Parameter #5	Real number. Defines z_f, the s_z-offset of the paraboloid in metres.
Parameter #6	Integer number. Specifies the distortion type, *ndistort*.
Parameter #7	Character string. Specifies the name of the file containing the distortion parameters, *distortfilename*.

Currently the only options are $ndistort = 0$, where there is no distortion applied, functionally equivalent to the use of Parameter #1 = PARABOLOID, or $ndistort = 1$. When $ndistort = 1$, the coordinates of the reflector surface are defined by,

$$Y(s_x, s_z) = y_p(s_x, s_z) + d(r, \theta) \tag{3-31}$$

where r and θ are related by,

$$s_x = r\cos\theta \quad , \quad s_z = r\sin\theta \tag{3-32}$$

and $d(r,\theta)$ is a deviation function defined by,

$$d(r,\theta) = R(r)\Theta(\theta) \tag{3-33}$$

with

$$R(r) = \rho_1 r^2 + \rho_2 r^4 + \ \ldots + \rho_N r^{2N} \tag{3-34}$$

$$\Theta(\theta) = a_0 + \sum_{i=1}^{M} \big(a_i \cos(i\theta) + b_i \sin(i\theta)\big) \tag{3-35}$$

for real coefficients ρ_i, a_i and b_i.

The named file *distortfilename* must be present or an error is flagged. When present, it must take the form (comments optional):

M	comment.	Integer $M \geq 0$
a_0	comment.	
a_1		
....	..	
....	..	
a_M	comment.	Last of the a_i coefficients.
b_1	comment.	
....	..	
....	..	
b_M	comment.	Last of the b_i coefficients.
#	comment.	This must be a hash designator to separate file parts.
N	comment.	Integer $N \geq 1$
ρ_1	comment.	
....	..	
....	..	
ρ_N	comment.	Last of the ρ_i coefficients.

For example, for a circular dish of radius r_0 we may employ $N = 2$, specify $\rho_1 = 1/r_0^2$ and $\rho_2 = -1/r_0^4$ so that there is no deviation at the rim and maximum deviation when $r = r_0/\sqrt{2}$. With this special form, $R(r)$ is dimensionless and the coefficients a_i and b_i take units of distance. All units of distance are specified in metres.

If Parameter #1 = VARIPARAB1 then,

Parameter #2	Real number. Defines f_l, the focal length of unmodified parabola $y_p(x, z)$ in metres.
Parameter #3	Real number. Defines x_f, the s_x-shift for the parabolic part in metres.
Parameter #4	Real number. Defines y_f, the $Y-$shift for the parabolic part in metres.
Parameter #5	Real number. Defines z_f, the s_z-shift for the parabolic part in metres.
Parameter #6	Real number. Defines z_c, the s_z-shift for the distortion factor in metres.
Parameter #7	Integer ≥ 0. The number, N, of additional α_i parameters (for $1 \leq i \leq N$).
Parameter #8	Real number. The exponential decay parameter, β, in m^{-1}.
Parameter #9	Real number. The first mandatory dimensionless parameter, α_0.
Parameter #10 to #(N+9)	Real numbers. These are the values α_i in units of m^{-1}. If $N = 0$ these parameters are not specified.

If Parameter #1 = VARIPARAB2 then,

Parameter #2	Real number. Defines f_l, the focal length of unmodified parabola $y_p(x, z)$ in metres.
Parameter #3	Real number. Defines x_f, the s_x-shift for the parabolic part in metres.
Parameter #4	Real number. Defines y_f, the $Y-$shift for the parabolic part in metres.
Parameter #5	Real number. Defines z_f, the s_z-shift for the parabolic part in metres.
Parameter #6	Real number. Defines z_c, the s_z-shift for the distortion factor in metres.
Parameter #7	Integer ≥ 0. The number, N, of additional α_i parameters (for $1 \leq i \leq N$).
Parameter #8	Real number. The hyperbolic tangent transition parameter, β, in m^{-1}.
Parameter #9	Real number. The first mandatory dimensionless parameter, α_0.
Parameter #10 to #(N+9)	Real numbers. These are the values α_i in units of m^{-1}. If $N = 0$ these parameters are not specified.

If Parameter #1 = VARIPARAB3 then,

Parameter #2	Real number. Defines f_l, the focal length of unmodified parabola $y_p(x, z)$ in metres.
Parameter #3	Real number. Defines x_f, the s_x-shift for the parabolic part in metres.
Parameter #4	Real number. Defines y_f, the $Y-$shift for the parabolic part in metres.
Parameter #5	Real number. Defines z_f, the s_z-shift for the parabolic part in metres.
Parameter #6	Real number. Defines z_c, the s_z-shift for the distortion factor in metres.
Parameter #7	Integer ≥ 0. The number, N, of additional α_i parameters (for $1 \leq i \leq N$).
Parameter #8	Real number. The hyperbolic tangent transition parameter, β, in m^{-1}.
Parameter #9	Real number. The first mandatory dimensionless parameter, α_0.
Parameter #10 to #(N+9)	Real numbers. These are the values α_i in units of m^{-i}. If $N = 0$ these parameters are not specified.

If Parameter #1 = VARIPAR2D1 then,

Parameter #2	Real number. Defines f_l, the focal length of unmodified parabola $y_p(x, z)$ in metres.
Parameter #3	Real number. Defines x_c, the s_x-shift for the distortion factor $\beta(x)$ in metres.
Parameter #4	Real number. Defines y_f, the $Y-$shift for the parabolic part in metres.
Parameter #5	Real number. Defines z_f, the s_z-shift for the parabolic part in metres.
Parameter #6	Real number. Defines z_c, the s_z-shift for the distortion factor $\alpha(z)$ in metres.
Parameter #7	Integer ≥ 0. The number, N, of additional α_i parameters (for $1 \leq i \leq N$).
Parameter #8	Real number. The hyperbolic tangent transition parameter, β, in m^{-1}.
Parameter #9	Real number. The first mandatory dimensionless parameter, α_0.
Parameter #10 to #(N+9)	Real numbers. These are the values α_i in units of m^{-1}. If $n = 0$ these parameters are not specified.
Parameter #(N+10)	Integer ≥ 1. The number, n, of a_i parameters (for $1 \leq i \leq n$).
Parameter #(N+11)	Real positive number x_M. This is a scaling distance in units of m.
Parameter #(N+12) to #(N+11+n)	Real dimensionless numbers a_i for the harmonic expansion in $x-$direction.

If Parameter #1 = VARIPAR2D2 then,

Parameter #2	Real number. Defines f_l, the focal length of unmodified parabola $y_p(x, z)$, in metres.
Parameter #3	Real number. Defines b_0, the control parameter for the Θ scaling function, in metres.
Parameter #4	Real number. Defines c_0, a parameter for amplitude control, in m^{-1}.
Parameter #5	Real number. Defines g_0, a parameter for curvature control, dimensionless.
Parameter #6	Real number. Defines z_c, the s_z-shift for the distortion factor $\alpha(z)$ in metres.
Parameter #7	Integer ≥ 0. The number, N, of additional α_i parameters (for $1 \leq i \leq N$).
Parameter #8	Real number. The hyperbolic tangent transition parameter, β, in m^{-1}.
Parameter #9	Real number. Defines p, a dimensionless exponent > 0.
Parameter #10 to #(N+9)	Real numbers. These are the values α_i in units of m^{-i}. If $N = 0$ these parameters are not specified.
Parameter #(N+10)	Integer ≥ 1. The number, n, of a_i parameters (for $1 \leq i \leq n$).
Parameter #(N+11)	Real number θ_M. This is a scaling angle, dimensionless.
Parameter #(N+12) to #(N+11+n)	Real dimensionless numbers a_i for the harmonic expansion in $x-$direction.

If Parameter #1 = SPLINE1D then,

Parameter #2	Character string containing no separators. This defines the name of the input file specifying the cubic spline control points as defined previously.

3.18 The *BOUNDARY* key word

Description

This is mandatory if the reflector geometry is not defined by the GEOMFILE command. If present, it must also be accompanied by the SURFACE command. The first parameter defines the type of boundary. Subsequent parameters depend on the type of boundary.

Currently there are two types of permitted surface; a rectangular boundary RECTANGLE and an elliptical boundary ELLIPSE. The boundary command defines a uniformly meshed flat bounded rectangle or ellipse, defined in the (s_x, s_z) parameter plane. The triangles are constructed as groups of three nodes. The nodes are points that lie on the surface defined by $S(s_x, s_z)$, as defined by the SURFACE and optional DISTORTSURF command.

The list below summarises the command use.

Number of parameters	At least 2.
Key word requirement	Special. Mandatory if SURFACE command present.
Number of occurrences	1 or 0.
File location	Anywhere within input file.
Parameter #1	Character string. This defines the boundary type and must be either RECTANGLE or ELLIPSE.

Further parameters depend on the surface type.

If Parameter #1 = RECTANGLE then,

Parameter #2	Real number. The rectangle width, w_x, in metres defined parallel to the s_x-axis prior to rotation.
Parameter #3	Real number. The rectangle height, w_z, in metres defined parallel to the s_z-axis prior to rotation.
Parameter #4	Real number. s_x-coordinate of rectangle centre, x_c.
Parameter #5	Real number. s_z-coordinate of rectangle centre, z_c.
Parameter #6	Real number. Rotation angle, θ_r, of rectangle about its centre, rotated about a vector parallel to the cross-product, $\underline{\hat{s}}_x \times \underline{\hat{s}}_z$ where $\underline{\hat{s}}_x$ is the direction of increasing s_x and $\underline{\hat{s}}_z$ is the direction of increasing s_z. Defined in degrees.
Parameter #7	Integer. Number of meshing divisions, n_x, along the s_x-axis.
Parameter #8	Integer. Number of meshing divisions, n_z, along the s_z-axis.

See figure 3-14 below.

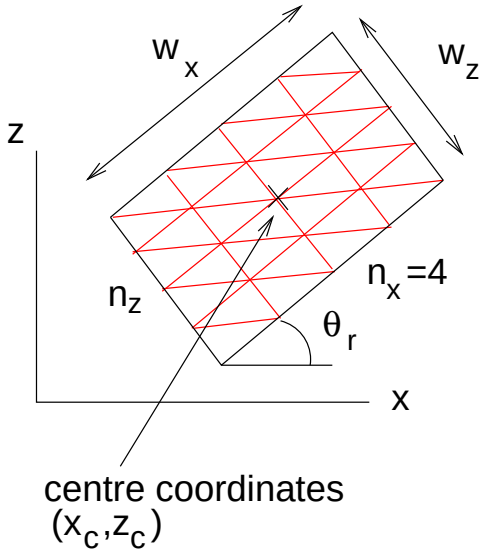

Figure 3-14: Boundary rectangle and its projection, shown for the case $s_x = X$ and $s_z = Z$.

If Parameter #1 = ELLIPSE then,

Parameter #2	Real number. Ellipse semi-major/minor axis along s_x-direction, before rotation. In metres.
Parameter #3	Real number. Ellipse semi-major/minor axis along s_z-direction, before rotation. In metres.
Parameter #4	Real number. s_x-coordinate of ellipse centre, x_c.
Parameter #5	Real number. s_z-coordinate of ellipse centre, z_c.
Parameter #6	Real number. Rotation angle, θ_r, of ellipse about its centre, rotated about a vector parallel to the cross product $\underline{\hat{s}}_x \times \underline{\hat{s}}_z$. (See RECTANGLE definition above). Defined in degrees.
Parameter #7	Real number. Target mesh size in metres.

Note that the target mesh size is used as a meshing criterion and is not precisely met. Meshing method is hard-wired into the software. This can be changed (and other hard-wired options used), but the default method divides the ellipse into concentric shells together with a central core. Each shell is uniformly divided into triangles to the target mesh size. The final remaining core assumes a single node at the centre, so that triangles here may be rather smaller.[5]

[5] Details of the method are not reported here.

3.19 The *DISTORTSURF* key word

3.19.1 Introduction

This command allows modification of the X and Z parts of the surface vector, as described in section 3.17.1. Introduced as of version 0.1.013 of the software, it may be used in addition to the SURFACE command. There are currently two optional forms implemented, but some background is required to explain them.

The requirement to explicitly control the z coordinate of a surface comes about when we wish to design an offset shaped reflector intended, for example, to provide a cosec-squared distribution with energy distributed in the downward direction towards the feed, as illustrated in figure 3-15 below. This requires a distortion at the edge of the reflector which

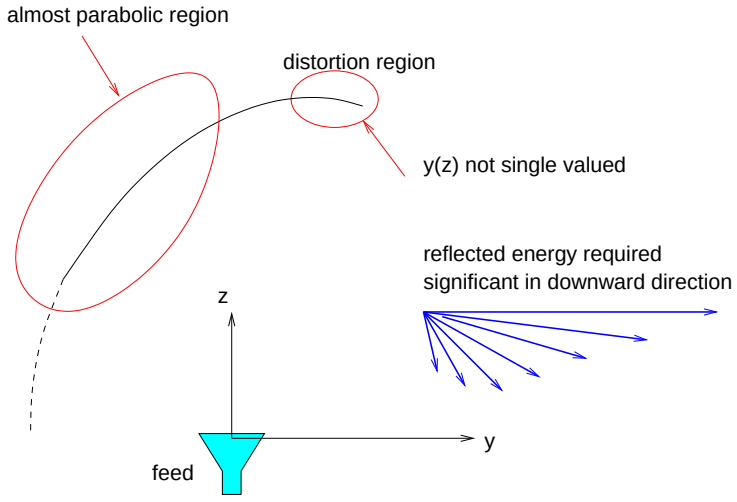

Figure 3-15: Shaping requirements for an offset shaped reflector with towards-feed directed power

focusses energy in a downwards direction. To do this it is usually necessary to permit the function $Y(Z)$ (e.g. in the cut $X = 0$) to be multi-valued and this cannot be achieved through parametrisation only of $Y(s_x, s_z)$.

We have experimented with suitable functional forms but we are constrained by a desire to employ a global parametric method; i.e. one for which the surface is defined by a single continuous function for the entire range of s_x and s_z. This allows us to avoid local spline functions which can be complex and non-portable between different CAD platforms. In particular, it allows a description of the geometry in a straight forward manner within CST [4].

3.19.2 Function options

Two functional forms are currently provides at software revision 0.1.013. These are decribed below.

Option 1

In this option we define

$$X(s_x, s_z) = s_x \tag{3-36}$$

and

$$Z(s_x, s_z) = Z(s_z) = s_z + \frac{c'}{2}\Big(1 + \tanh\big(-\beta'(s_z - s'_0)\big)\Big).\Big(\alpha'_0 + \sum_{i=1}^{N'} \alpha'_i\,(s_z - s'_0)^i\Big) \tag{3-37}$$

where the primed constants are set by the user independently of those employed in the SURFACE command. The motivation for this form is to ensure that for $s_z < s'_0$ we may choose a β such that there is negligible introduced distortion, i.e. $Z(s_z) \to s_z$ and for $s_z > s'_0$ the distortion is controlled by the power series.

Improvements

Option 1 has found some use, but it suffers from a lack of control over the shape of the distortion in the neighbourhood $s_z \sim s'_0$. Suppose that for $s_z < s'_0$ we have a limit function $Z(s_z) = f_1(s_z)$ and for $s_z > s'_0$ we have a second limit function $Z(s_z) = f_2(s_z)$ such that $f_2(s'_0) = f_1(s'_0)$. In the limit of some control parameter $\beta \to \infty$ we require that,

$$Z(s_z, \beta) = \begin{cases} f_1(s_z) & \text{for } s_z < s'_0 \text{ as } \beta \to \infty \\ f_2(s_z) & \text{for } s_z > s'_0 \text{ as } \beta \to \infty \end{cases} \tag{3-38}$$

and for β not infinite we require $Z(s_z, \beta)$ to be a smooth function of s_z such that $Z(s_z, \beta) \to f_1(s_z)$ for $s_z \ll s'_0$ and $Z(s_z, \beta) \to f_2(s_z)$ for $s_z \gg s'_0$. Moreover we also require $Z(s_z, \beta) < \min(f_1(s_z), f_2(s_z))$. For our application we have the special form $f_1(s_z) = s_z$. The 'minimum of the functions' requirement is necessary to ensure there is no 'dimpling' of $Z(s_z)$ in the neighbourhood of s'_0 which leads to the reflection of power in unexpected directions.

We considered an approach where we let $Z(s_z, \beta)$ take the form,

$$Z(s_z, \beta) = A(s_z, \beta) f_1(s_z) + \big(1 - A(s_z, \beta)\big) f_2(s_z)$$

where $0 \le A(s, z, \beta) \le 1$ is a function like $\big(1 + \tanh(-\beta(s_z - s'_0))\big)/2$. This is an approach from standard function theory but $Z(s_z, \beta)$ does not have the required property $Z(s_z, \beta) < \min(f_1(s_z), f_2(s_z))$ and leads to the same kind of problems noted for *option 1*. In the second option below, we provide an alternative functional form which does meet all the requirements.

Option 2

Here we assume a model of the form,

$$X(s_x, s_z) = s_x \tag{3-39}$$

$$Z(s_z) = (1+w)\left(\left(\frac{1+w}{|f_1(s_z)+a|}\right)^Q + \left(\frac{1}{|f_2(s_z)+a|}\right)^Q\right)^{-1/Q} - a \tag{3-40}$$

where Q is a parameter $|Q| > 1$ which assumes the role of β and there are two weight constants, $w > -1$ and a chosen so that both $f_1(s_z) + a > 0$ and $f_2(s_z) + a > 0$ for all s_z in the domain of the surface. As $Q \to +\infty$ the larger of the two terms to the power Q dominates, corresponding to the smaller of either $f_1(s_z)$ or $f_2(s_z)$. Similarly as $Q \to -\infty$ the smaller of the two terms dominates.

If we choose $Q > 1$,

$$f_1(s_z) = s_z \tag{3-41}$$

$$f_2(s_z) = s_0' - \left(h' + \sum_{i=1}^{N'+1} \alpha_i' \left(s_z - s_0'\right)^i\right)\left(1 + \tanh\left(d'(s_z - s_0')\right)\right) \tag{3-42}$$

we have a suitable function. Note that the hyperbolic tangent function is intended to provide an additional level of control of the smallest value that $f_2(s_z)$ can attain for $s_z < s_0'$. The constant h' can now be included with the sum over powers of $(s_z - s_0')$ and the constant d' controls the rate of fall-off of the hyperbolic tangent.

3.19.3 Syntax

The key word DISTORTSURF is optional. If present it may be called only once. To summarise,

Number of parameters	At least 9.
Key word requirement	Optional.
Number of occurrences	1 or 0.
File location	Anywhere within input file.
Parameter #1	Integer. This defines the function option; Option 1 (= 1) or Option 2 (= 2). Currently only values of 1 and 2 are recognised.

Further parameters are option specific.

If Parameter #1 = 1 then

Parameter #2	Real number. Offset value s'_0 in metres.
Parameter #3	Real number. Scale value c' in metres.
Parameter #4	Real number. Unused in this option.
Parameter #5	Real number. Unused in this option.
Parameter #6	Real number. Unused in this option.
Parameter #7	Integer. Number of terms $N' \geq 0$
Parameter #8	Real number. Transition factor, β', dimensionless.
Parameter #9	Real number. Value α'_0, dimensionless.
Parameters #10 to #(9+N')	The N' Real numbers, α'_i with units of m^{-1}.

If Parameter #1 = 2 then

Parameter #2	Real number. Offset value s'_0 in metres.
Parameter #3	Real number. Weight value a in metres.
Parameter #4	Real number. Weight value w, dimensionless.
Parameter #5	Real number. Rate control constant d' in m^{-1}.
Parameter #6	Real number. Offset value h' in metres.
Parameter #7	Integer. Number of terms $N' \geq 0$
Parameter #8	Real number. Transition value Q, dimensionless.
Parameters #9 to #(9+N')	The $N'+1$ Real numbers, α'_i with units of m^{-i}.

3.20 The *BLOCKAGE* key word

Description

This command is optional. If present it must occur only once and describes the nature of a blockage to the main reflector. Currently it takes two forms. The first form is where the feed is responsible for the blockage (*FEED* designation), assuming that it is only the feed aperture that is responsible for the blockage. The second form (*NEW* designation) requires the user to specify a file name containing a points-and-joins file with the same format as specified in defining the reflector. This allows an arbitrary blockage to be defined.

The list below summarises the command use.

Number of parameters	At least 1.
Key word requirement	Optional.
Number of occurrences	1 or 0.
File location	Anywhere within input file.
Parameter #1	Character string. This defines the feed type and must be either *FEED* or *NEW*.

If Parameter #1 = FEED then there are no further parameters.

If Parameter #1 = NEW then

Parameter #2	Character string containing no separators. This is the file name of the points-and-joins file specifying the blockage.

3.21 The *NEARFIELD* key word

Description

This command is optional. If present it defines a rectangular region in space where the near fields are to be evaluated. Various options are also provided to provide an alternative method to calculate the effect of the blockage to the far field and/or to determine the far field from the calculated near field.

Firstly, it is assumed that the near field is to be specified on the plane $y = y_0$ on a rectangle defined by $x_{min} \leq x \leq x_{max}$ and $z_{min} \leq z \leq z_{max}$.

Secondly, if the *shadow mask* option is flagged as 'True', the geometric rays from the reflector are traced through the blockage on to the $y = y_0$ plane. These are the rays previously discussed when defining the effect of the blockage. Now, however, *all* the currents are evaluated on the reflector including the ones which (in the basic method) are taken as non-contributory if the respective facets were taken as being in shadow. In this option, it is the plane $y = y_0$ plane which defines the existence of shadow. Thus, if any shadowing rays intersect the plane $y = y_0$ for which $x_{min} \leq x \leq x_{max}$ and $z_{min} \leq z \leq z_{max}$ then the fields there are assumed zero. Given a, possibly masked, near field region the far field may then be computed. This is performed if the *far-field* flag is set 'True'. Figure 3-16 shows the near field region and the effect of shadowing on the far field using this alternative method.

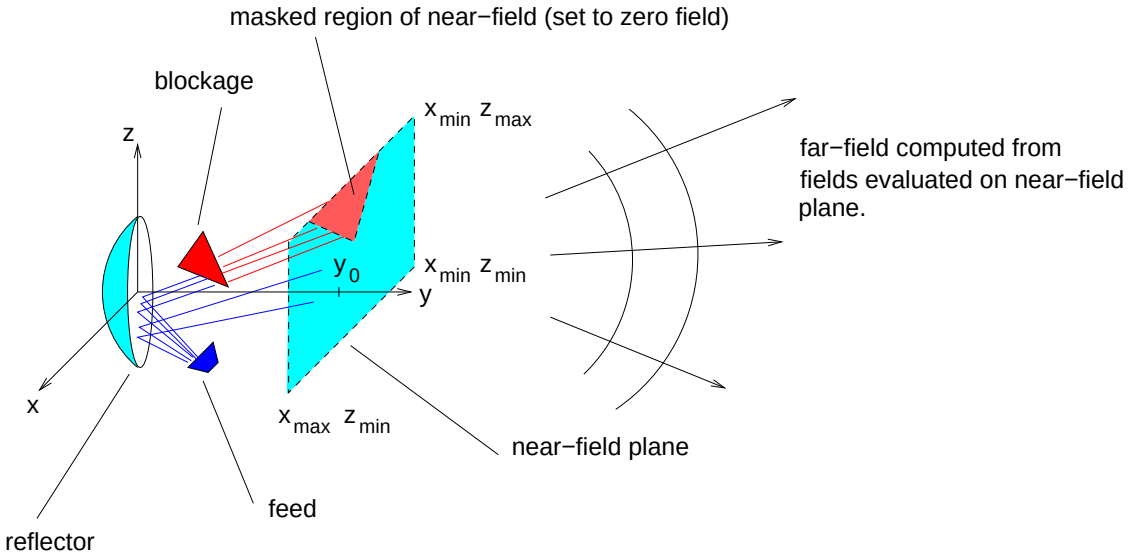

Figure 3-16: Near field region, shadow masking and far field

The method of near-to-far field transformation is similar to the one employed for evaluating the far-field from the aperture of a feed. The method uses (2-2) where,

$$\begin{aligned} \underline{J}_e &= \hat{n} \times \underline{H}(\underline{r}') \\ \underline{J}_m &= -\hat{n} \times \underline{E}(\underline{r}') \end{aligned} \qquad (3\text{-}43)$$

where $\underline{H}(\underline{r}')$ and $\underline{E}(\underline{r}')$ are the near fields evaluated on the plane $y = y_0$. Note that the sampling must be sufficiently fine over the rectangular region (typically with a sampling interval $\lesssim 0.2$ wavelengths) and the rectangle must be large enough such that the fields outside the rectangle are small enough to ignore. When no blockages are present, this method of computing the far-field is inferior to the main method.

The list below summarises the command use.

Number of parameters	11.
Key word requirement	Optional.
Number of occurrences	1 or 0.
File location	Anywhere within input file.
Parameter #1	Real number. This defines the position of evaluation plane, y_0, in metres.
Parameter #2	Real number. Minimum x-coordinate of near-field region, x_{min} in metres.
Parameter #3	Real number. Maximum x-coordinate of near-field region, x_{max} in metres.
Parameter #4	Integer. Number of evaluation points, n_x, between x_{min} and x_{max}.
Parameter #5	Real number. Minimum z-coordinate of near-field region, z_{min} in metres.
Parameter #6	Real number. Maximum z-coordinate of near-field region, z_{max} in metres.
Parameter #7	Integer. Number of evaluation points, n_z, between z_{min} and z_{max}.
Parameter #8	Character. This represents the shadow mask flag. If this is 'T', then a blockage is assumed, defined by the *BLOCKAGE* option. If this is 'F', no account is take of blockage effects.
Parameter #9	Character. This represents the far-field flag. If this is 'T', then the far-field is calculated from the near-field region, with or without masked fields. If this is 'F' then no far-fields are calculated using this method.
Parameter #10	Character string with no separators. This is the name of the near-field file to be created.
Parameter #11	Character string with no separators. This is the name of the far-field file to be created.

3.22 The *GRASPOUT* key word

Description

This command is optional and constructs a geometry .sfc file suitable for input into GRASP [1]. The file only contains the reflector nodes with coordinates offset by the specified x, y and z offsets , x_0, y_0, z_0 defined here. GRASP uses this "scatter profile" to generate its own curved surface through these points. This file may only be used if a full version of GRASP is available for analysis.

Number of parameters	4
Key word requirement	Optional.
Number of occurrences	1 or 0.
File location	Anywhere within input file.
Parameter #1	Character string containing no separators. This defines the name of the GRASP .sfc file.
Parameter #2	Offset coordinate x_0 in metres.
Parameter #3	Offset coordinate y_0 in metres.
Parameter #4	Offset coordinate z_0 in metres.

3.23 The *FILEREFL* key word

Description

This command is optional and is intended mostly for diagnostics. It provides the incident electric and magnetic field components evaluated at the nodal points of the reflector surface. There are three coordinate options. In the first (designation *CART_REFL*), the fields are defined in the global Cartesian coordinate system. In the second (designation *SPH_FEED*), the fields are defined in spherical coordinates in the coordinate system of the feed. In the third (designation *CART_FEED*), the fields are defined in Cartesian coordinates in the coordinate system of the feed.

Number of parameters	4
Key word requirement	Optional.
Number of occurrences	1 or 0.
File location	Anywhere within input file.
Parameter #1	Character string containing no separators. Name of the electric field output file.
Parameter #2	Character string containing no separators. Name of the magnetic field output file.
Parameter #3	Character string. Coordinate options *CART_REFL*, *SPH_FEED* or *CART_FEED*.

3.24 The *FEEDFILE* key word

Description

One of several methods to illuminate the reflector and is logically considered as a feed type. This defines a user-specified feed aperture distribution, whose format is that of a CST [4] output file. It is included as of version 0.0.013 of the software.

An aperture distribution is specified by a regular rectangular grid of points where the fields are specified. There are two options: either the electric field only is specified or both the electric and magnetic fields. In both cases, the z'-components of the fields perpendicular to the aperture are not required although they are specified in the input files for consistency with CST [4] format.

Let us assume the aperture lies in the x', y' plane of the feed coordinate system. When only the (transverse) electric field is specified The transverse magnetic field is approximated from the tangential electric fields (E_x, E_y) via,

$$\underline{H} \approx H_x \underline{\hat{x}}' + H_y \underline{\hat{y}}' \tag{3-44}$$

where

$$H_x \approx -f_0 Z_0 E_y \quad H_y \approx f_0 Z_0 E_x \tag{3-45}$$

where f_0 is a power scaling factor taken to ensure that the power radiated by the aperture is unity. This approximation is valid if the fields are locally planar and is usually considered good in the apertures of horns, provided the flair angle is not too large.

If both the (transverse) electric and magnetic fields are available this approximation is not required, though we still calculate the component of the (real part of the) Poynting vector normal to the aperture to normalise total power radiated by the aperture to unity.

The component of the reactive part of the Poynting vector is also computed as a diagnostic. If only the (transverse) electric field is available this component is estimated as exactly zero. If both electric and magnetic fields are available it will in general be non-zero. If so, the reactive component should be small compared to the radiated part or else the program will issue a warning and the supplied aperture fields should be considered as poor.

The specified aperture files take the same format. This consists of a list of rows where the first three entries in a row specify the x', y' and z' coordinates in millimetres (the z' coordinates are not used except for diagnostics; i.e. the field is assumed to be given on the plane $z' = 0$).[6]. The next three specify the x', y' and z' components of the real part of the electric field (electric field file) or magnetic field (magnetic field file). The next three specify the x', y' and z' components of the imaginary parts of the fields. This is consistent with CST output files.

[6] Note that the units of mm are for compatibility with CST export files where, for microwave applications, units are specified in mm. This is in contrast with the units of distance elsewhere in the software which are specified in metres

When using an exported CST output file, employ a text (.txt) export format and ensure distance units are in millimetres (this is the CST default condition). Also ensure that the plane on which the fields are specified is transverse to the CST z-axis, for consistency with the above feed coordinate system. Finally, remove (delete) the header lines in the text file containing the ASCII description of the numbers. There are (CST versions 6 to 10) two such header lines.

When only the electric field file is supplied, scaling of the electric field components is unimportant since the program will re-normalise to assume 1 Watt is radiated. When both field files are supplied there can be an arbitrary common scaling factor, but this must be common to both files. A regular grid is assumed. If the program detects the specified coordinates do not form a regular grid, or there is a difference in the specified coordinates for the electric and magnetic field data, an error is flagged.

In theory, the use of both electric and magnetic fields should result in greater accuracy which is important for describing cross-polar performance and side-lobes. However, as of version 1.003 (May 2010), we have not performed exacting validation to test this assertion.

The list below summarises the use of this command.

Number of parameters	5 or 6
Key word requirement	Special. Represents one of several types of 'feed'. One and only one 'feed' type must be present.
Number of occurrences	1 or 0.
File location	Anywhere within input file.
Parameter #1	Integer. This must be 1 or 2. If 1, then only the electric field is supplied. If 2, then both electric and magnetic field files are required.
Parameter #2	Name of the required electric field data file (CST format)

If parameter #1=1, then

Parameter #3	Frequency in MHz
Parameter #4	Display scaling distance in metres (x' direction)
Parameter #5	Display scaling distance in metres (y' direction)

If parameter #1=2, then

Parameter #3	Name of the required magnetic field data file (CST format)
Parameter #4	Frequency in MHz
Parameter #5	Display scaling distance in metres (x' direction)
Parameter #6	Display scaling distance in metres (y' direction)

Parameter #3(4) must be the same as the frequency (only a single frequency is permitted here) specified by the *FREQS* command. This is required explicitly as a check since the

CST file does not specify the frequency at which the data is generated. If the frequency is not correct the program terminates. Parameters #4(5) and #5(6) are only used for illustration purposes for use by *PLOTFILE*. This is because the aperture file is read in after the information is defined for generating the POSTSCRIPT files and so the aperture coordinates available here cannot easily be used for this purpose. It is logically preferred to use different illustration parameters until or unless the software is significantly changed.

3.25 The *FARPOL* key word

Introduced in version 0.1.003 of the software, this permits a rotation of the polarisation base of the far-field and defines a value for the polarisation rotation angle ζ_s. The key word is optional but if present must occur only once. If not present is is assumed that $\zeta_s = 0$.

The list below summarises the use of this command.

Number of parameters	1
Number of occurrences	0 or 1
File location	Anywhere within input file.
Parameter # 1	Real number. This is the angle ζ_s is degrees.

4 Examples and CST comparison

4.1 Introduction

It is difficult to validate PO software because (1) by definition the method only provides an approximate solution to Maxwell's equations and (2) there are a large number of methods of implementation and subsidiary approximations that can give rise to slightly different results. We are also hampered by the lack of an alternative PO software tool, since general purpose PO tools are commercial (e.g. GRASP [1]) and we have no license for such.

We have employed this PO software for construction of parabolic antennas with rectangular feeds and with feed files imported from CST for more realistic feeds. We have also tested the software using simple flat plates. For example, a large flat plate illuminated by a Gaussian beam should reflect the Gaussian beam unaltered into the far-field.[7] Other tests, including mesh convergence comparisons, have also been performed.

In our first documented example we will provide a comparison of the predicted far-fields by our software and by CST of a shaped reflector illuminated by a modified Gaussian beam. This is a demanding comparison of our implementation of PO vs an "exact" solution using CST. In performing this test it has been necessary to write two CST macros. We have made these globally available. One macro constructs a shaped reflector according to our *VARIPAR2D1* definition. The second macro defines an incident far-field file for a modified Gaussian beam according to our *MGAUSSIAN* definition in the far field of the beam. These both require some further description.

4.2 CST Macro "Shaped_Reflector_A"

This macro has been written specially for the shaped reflector application for use with CST [4]. It is globally available[8] and is accessed under top menu item "macros" under "macros - solver - construct - miscellaneous - shaped reflector A" within CST. The macro is written in CST Visual Basic and implements the construction of a shell structure of perfect conductor as defined under *SURFACE VARIPAR2D1* and *BOUNDARY ELLIPSE*. Currently we assume that there is no rotation of the structure under the *BOUNDARY* definition; i.e. $\theta_r = 0$. (First version of the macro). The target mesh size in the *BOUNDARY* definition is not required for CST application.

The macro is based around the CST intrinsic command "object - basic shapes - analytical face". This requires a character string definition of a formula for parameterising a surface, which the macro provides according to our *VARIPAR2D1* definition. We assume both N and n are fixed with $N = 5$ and $n = 6$ (which we have found is a large enough number for

[7] This is only strictly true provided the Gaussian beam approximates a solution of Maxwell's equations sufficiently well over the range of angles under consideration. For wide angles and relatively small beam wastes this requires the use of the "modified" non-paraxial form.

[8] On both CST platforms within the company

most purposes). The macro is simple to use; a pop-up window will appear requiring the user to enter all the required parameters. These include both the parameters of the surface and the parameters of the boundary to the surface. The latter is elliptical and defined by the third column of numbers in the window. Note that all distances are assumed in the pop-up window to be in metres and the CST default units are assumed to be in mm with frequencies in GHz. These parameters will appear in the parameter file should they be required for other purposes. An independent check of the defined structure was made by generation of a *SOLIDWORKS* [7] model and comparison of the Solidworks model with a CST exported .SAT file.

4.3 CST Macro "Far field Gaussian beam source"

This macro has been written specially for use with the moment method frequency domain solver in CST and in particular the integral equation solver. Currently, as of CST version 12, there is no intrinsic macro for creation of a far field Gaussian beam source. If the integral equation solver is employed an incident Gaussian beam must be specified as an incident far-field (a near-field incident field has not been implemented).

The standard paraxial Gaussian beam is inconsistent with a representation in the far-field because, by definition, it assumes the direction of propagation is along the axis of the beam whereas radial propagation from the source is required. This is not an issue with our modified Gaussian beam definition where no further non-paraxial approximations are required.

Under the modified Gaussian beam, defined according to our *MGAUSSIAN* definition section 2.4.3, we require a far-field consistent with the CST definition of far-field. In CST, the far-field is defined by the *electric field pattern* as a function of polar angles θ and ϕ defined, as usual, in a local (source) cartesian x', y', z' coordinate system. The far-field electric field $\underline{E}(\underline{r})$ and the CST electric field pattern vector $\underline{E}_{pattern}(\theta, \phi)$ are related by,

$$\underline{E}(\underline{r}) = \left(\frac{\exp(-jk_0 r)}{r} \right) \underline{E}_{pattern}(\theta, \phi) \qquad \text{as } r \to \infty \tag{4-1}$$

where r is the radial distance from the source. The pattern vector is written in the form,

$$\underline{E}_{pattern}(\theta, \phi) = E_\theta \, \underline{\hat{\theta}} + E_\phi \, \underline{\hat{\phi}} \tag{4-2}$$

where E_θ and E_ϕ are tabulated in a pattern file in terms of their real and imaginary parts.

Using the fact that $\Theta \to \theta$, $w_0/w(z) \to z_R/(r\cos\theta)$ and $\rho\cos\theta/w(z) \to (z_R \sin\theta)/w_0$ as z and hence $r \to \infty$ under the definitions of section 2.4.3, it is straightforward to show that,

$$\underline{E}_{pattern}(\theta, \phi) = A_0 z_R e^{j\pi/2} \exp\left(- \left[\frac{z_R \sin\theta}{w_0} \right]^2 \right) \left(\sin\phi \, \underline{\hat{\theta}} + \cos\phi \cos\theta \, \underline{\hat{\phi}} \right) \tag{4-3}$$

for the modified Gaussian beam, assuming $z > 0$. For $z < 0$ the phase factor $e^{j\pi/2} \rightarrow e^{-j\pi/2}$ and for $z = 0$ the phase factor $e^{j\pi/2}$ is replaced by unity. However, in our context it makes sense only to define a beam in the forward direction so we will redefine,

$$\underline{E}_{pattern}(\theta,\phi) = \begin{cases} A_0 z_R e^{j\pi/2} \exp\left(-\left[\frac{z_R \sin\theta}{w_0}\right]^2\right)\left(\sin\phi\,\underline{\hat{\theta}} + \cos\phi\cos\theta\,\underline{\hat{\phi}}\right) & \text{for } 0 \le \theta \le \pi/2 \\ 0 & \text{for } \pi/2 < \theta \le \pi \end{cases} \tag{4-4}$$

where θ is assumed defined only over the range $0 \le \theta \le \pi$. Note that we take,

$$A_0 \approx \frac{2}{w_0}\sqrt{\frac{Z_0 P_0}{\pi}} \tag{4-5}$$

assuming $P_0 = 1$ Watt, and

$$z_R = k_0 w_0^2/2 \tag{4-6}$$

as defined in section 2.4.3. This is a beam polarised with electric field in the local coordinate y (i.e. in our context the y') direction.

If $k_0 w_0/2 < 1$ the far-field is significant when $\theta = \pi/2$ resulting in an un-physical discontinuity as θ is varied over this value. This is a limitation of the modified Gaussian beam formulation when the beam waste radius w_0 is too small compared to a wavelength (i.e. if $w_0 < \lambda/\pi$). We therefore issue a warning if parameters are entered into the macro resulting in a value of $k_0 w_0/2 < 1$. The limit arises when the -8.69 dB power level corresponds to a value of $\theta = \pi/2$ for which no Gaussian beam model is appropriate.

In use, the CST macro is set up to generate a pattern file which must subsequently be referenced using the "farfield source - sourcefile" option under the "alignment" tag within the farfield source pull-up menu. The named data file is stored within the CST project file name folder under the following directories:

$$\langle\text{project file name}\rangle - \text{Model} - \text{3D} - \langle\text{named farfield source data file}\rangle$$

The farfield source file assumes the source lies at the global origin $x = y = z = 0$ with local y' axis aligned by default with the $y-$axis. The source will therefore need to be moved. Usually this will be a translation to the focal point of the reflector and rotated by 90^o so that the beam points towards the reflector. Also note that the source file is defined at a particular frequency which must match the CST analysis frequency.

4.4 Use of an optimiser

The PO software can be used with an optimiser. This is a general purpose simplex optimiser which can be used as a shell to any program, treated as a 'black box' that requires an input file with a specification of parameters to be controlled and an output file that generates data to be optimised. Since objectives can be quite complicated, it is necessary to modify one of the modules that reads the output data and generates a cost function on a case-by-case basis. Use of the optimiser is described in [8] and has been found useful for the design and refinement of shaped reflectors.

4.5 Example 1. Reflection of Gaussian beams from a flat reflector

In this example we use both a standard paraxial and a modified Gaussian beam as the source to a disk shaped flat reflector of finite size. We consider the source beam waste to be at a distance f_l from the reflector, with beam normal to the reflector. The beam is $x-$polarised in the reflector coordinate system so the scattered waves are $\underline{\hat{\theta}}$ polarised in the principal planes. We plot the gain, in dB, for a set of different values of f_l using either the standard paraxial (*PGAUSSIAN*) or the modified form (*MGAUSSIAN*).

In both cases the beam width is defined by the semi-angle $\theta_0 = 25^o$. For the modified Gaussian beam this corresponds to a beam waste radius $w_0 \approx 9.032$ mm at 25 GHz, whereas for the paraxial Gaussian beam $w_0 \approx 8.748$ mm. The input file is defined by,

```
CALCOPTS 1
ANGLES  90.0, 0.0, 1, 00.0, 0.5, 900
FREQS 25000.0 000.0 1
FILENAME test_out junk_out
FEEDCEN  0.0  <fl>  0.0
FEEDROT  90.000  -90.000  90.0
MGAUSSIAN 25.0
%%PGAUSSIAN 25.0
PLOTFILE sideview1.ps 00.00  00.0 180.0  0.50 0.1 0.0 T F F F
PLOTFILE sideview2.ps 90.00  00.0 180.0  0.50 0.1 0.0 T F F F
SURFACE PLANE 0.00 1.00 0.00 0.00 0.00 0.00
BOUNDARY ellipse 0.500, 0.500, 0.0 0.00 00.0 0.003
GEOMFILE qqq_pj.dat  RW
```

where the entry $< fl >$ is the distance f_l in metres. In these predictions $f_l = 0.15$, 0.30, 0.45, 0.60 and 0.75. The input file, as given verbatim, is setup for the MGAUSSIAN modified Gaussian case.

If the beam represents an exact solution to Maxwell's equations and the reflector is large enough such that the edges of the disk are not illuminated, then the PO solution should represent an accurate reflection of the beam and the scattered wave (gain) should have magnitude independent of f_l and represent a continuation of the beam.

Figure 4-1 shows predictions for the paraxial Gaussian case and 4-2 for the modified Gaussian case. For the paraxial form the -8.7 dB points occur at 90 ± 25.0^o, for the modified form the -8.7 dB points occur at 90 ± 24.0^o. This slight difference is to be expected since the modified Gaussian beam does not have perfect axial symmetry. Both forms show a non-Gaussian distortion as f_l is increased. This is probably due to edge illumination becoming more significant (and is more so for the modified Gaussian beam[9]). For example when $f_l = 0.75$ m, the angle between beam axis and reflector edge is $\arctan(0.5/0.75) \approx 33.5^o$ which corresponds to a power level about 16 dB below the value on axis. Overall, both representations show a gain which is approximately independent of f_l to about 15dB from peak.

[9]The beam width for the modified Gaussian represents a kind of average, since the beam is not perfectly axially symmetric. Also, more power is radiated at larger angles for a given beam width because the paraxial form incorrectly assumes radiated power is directed only parallel to the axis.

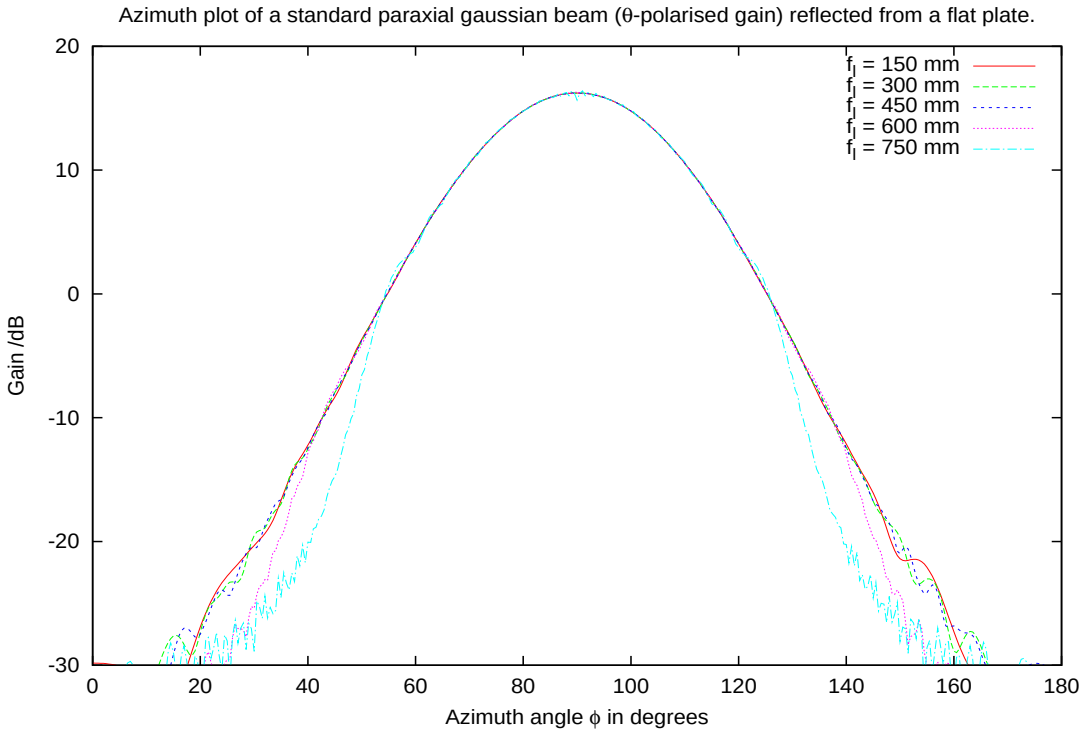

Figure 4-1: Standard paraxial Gaussian beam reflection (absolute gain in dB), x-polarised, azimuth cut.

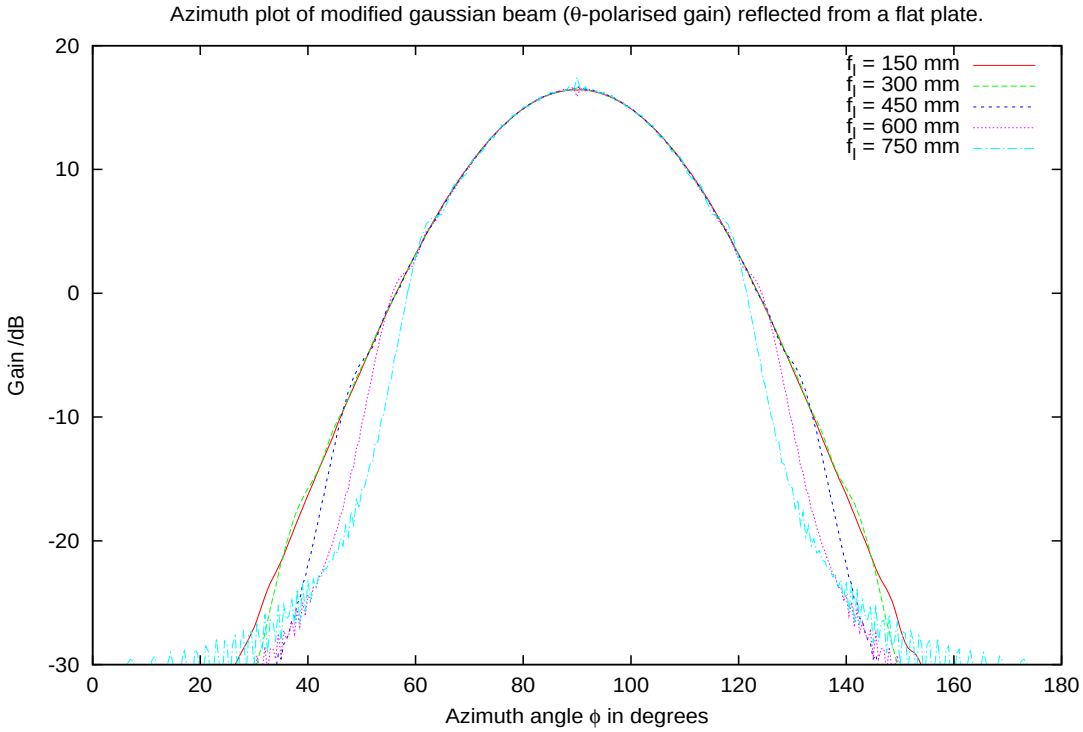

Figure 4-2: Modified non-paraxial Gaussian beam reflection (absolute gain in dB), x-polarised, azimuth cut.

4.6 Example 2. A large shaped reflector

We consider a shaped reflector test problem with a modified Gaussian beam with source at the nominal focal point of the reflector and beam directed along the global $y-$axis of the reflector. The reflector is "pringle" shaped with an elliptical footprint of size 1000 mm by 680 mm operating at 25 GHz, specified by the parameters set out in the PO input file below:

```
CALCOPTS 1
ANGLES  90.0, 0.0, 1, 00.0, 0.5, 720
%%ANGLES 00.0, 0.3, 600, 90.0, 0.0, 1
FREQS 25000.0 000.0 1
FILENAME test_out junk_out
FEEDCEN  0.0 0.60 0.0
FEEDROT  90.000  -90.000  90.0
MGAUSSIAN 28.5
PLOTFILE sideview1.ps 00.00  00.0 180.0  0.50 0.1 0.0 T F F F
PLOTFILE sideview2.ps 90.00  00.0 180.0  0.50 0.1 0.0 T F F F
SURFACE VARIPAR2D1 0.60 0.0 0.60 0.0 0.0 5 12.000 0.00 0.02 -0.00 0.5 -0.00  1.50  \
                   6 1.40 0.00 0.08 0.0 0.0 0.002  0.003
BOUNDARY ellipse 0.500, 0.34, 0.0 0.02 00.0 0.010
GEOMFILE qqq_pj.dat  RW
```

The ANGLE tab is set for an azimuth scan. The elevation scan is obtained by uncommenting the %%ANGLE tab and commenting out the current one. The SURFACE line is split into two parts (represented by the ' \' character) to fit in this document, but is taken as one line in the input file. Note that the Gaussian beam semi-width specified by 28.5^o is equivalent to a beam waste radius $w_0 = 8.000$mm at 25 GHz. The source field is polarised in the $x-$direction of the global coordinate system, which results in a $\underline{\hat{\theta}}$ directed scattered field along the principal $y-z$ (elevation) and $x-y$ (azimuth) planes. Since the reflector is mirror symmetric about the $y-z$ plane there should, for an exact prediction, be no $\underline{\hat{\phi}}$ directed scattered field component in the principal planes either in CST or in physical optics.

The far-field predictions generated using this input file are compared with CST, using the specially written 'far field Gaussian beam source' macro and the 'shaped reflector A' macro under the integral equation solver as described earlier. It is important, when using the former, to use a sufficient angular sampling of the beam. In this example, the number of θ samples, $N_\theta = 46$ and the number of ϕ samples, $N_\phi = 61$. CST will generally flag a warning if insufficient samples are employed. The beam waste radius w_0 is set at $w_0 = 8.0$ mm and the frequency defined at 25 GHz. We employ CST version 12 with default meshing set at 5 cells per wavelength. Figure 4-3 shows a screen dump of the CST geometry, where the red radiation symbol represents the position of the source phase centre. Figure 4-4 shows a screen dump of the predicted three dimensional field magnitude. The geometry is set exactly as defined for the physical optics predictions.

Figures 4-5 and 4-6 compare the CST and PO $\underline{\hat{\theta}}$ components of the scattered field along the principal planes. This is shown as the absolute gain in dB. Comparison shows excellent agreement in the forward scattering hemisphere, $y > 0$, where PO is considered a valid approximation.

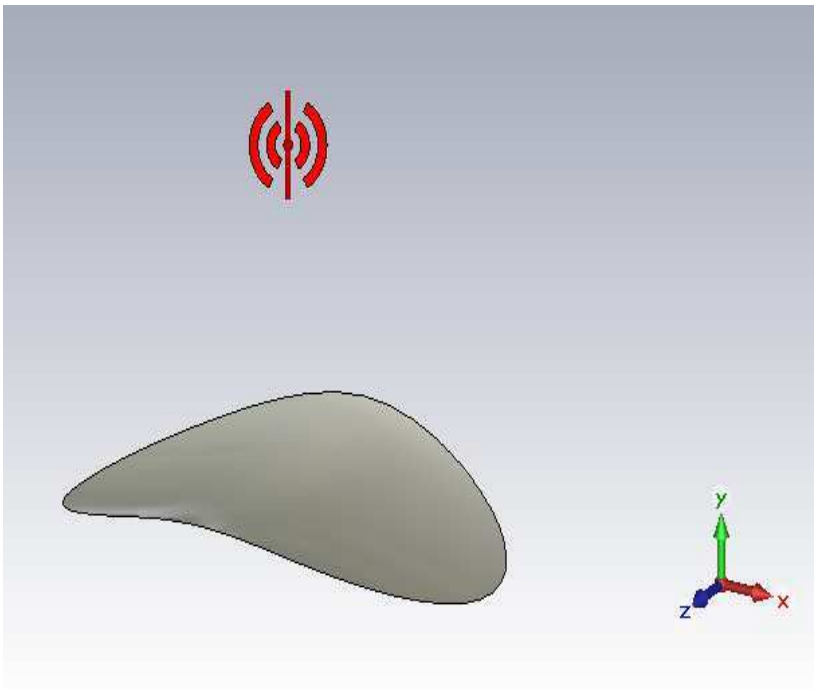

Figure 4-3: Screen dump of CST geometry, example 2.

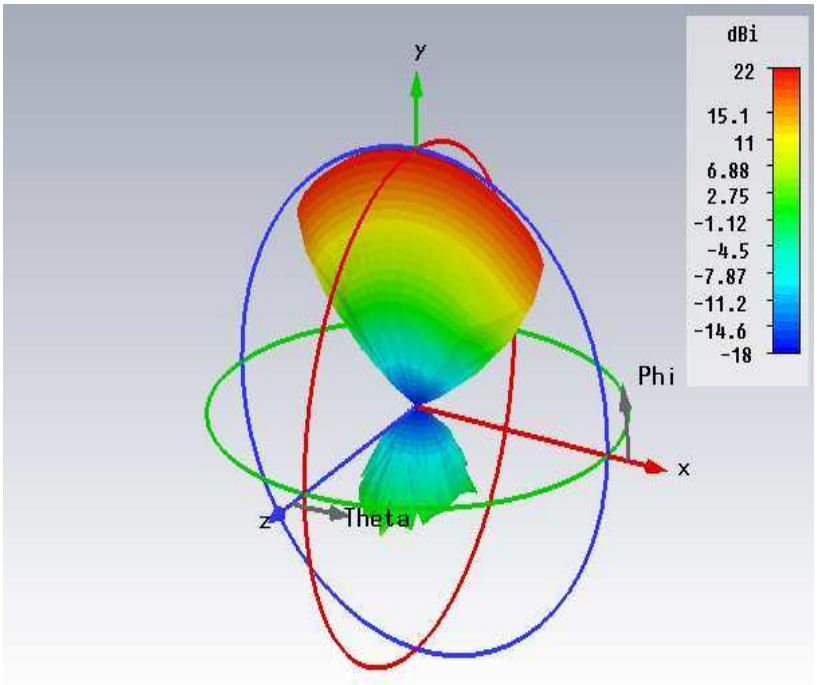

Figure 4-4: Screen dump of CST predicted 3D field magnitude

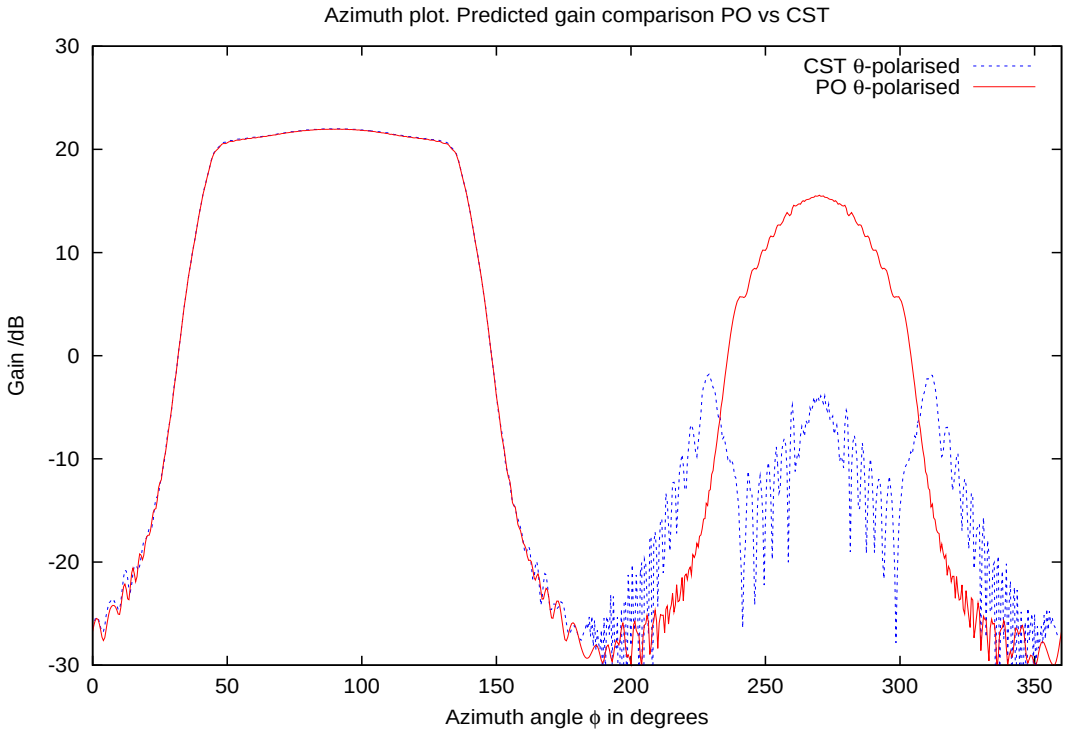

Figure 4-5: Scattered field comparison (absolute gain in dB), x-polarised, principal azimuth ($x-y$) plane

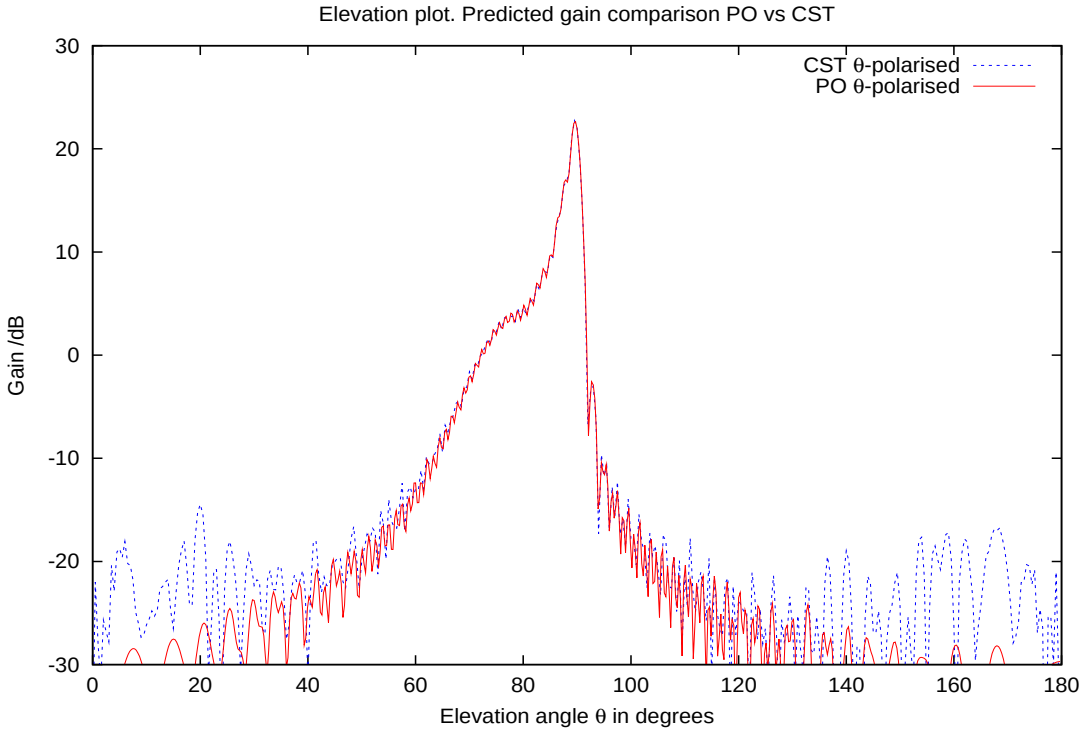

Figure 4-6: Scattered field comparison (absolute gain in dB), x-polarised, principal elevation ($y-z$) plane

4.7 Example 3. A smaller shaped reflector with a real feed

In this example a smaller shaped reflector is considered for use with a real feed. The feed was designed with CST [4] using the t-solver, to generate a beam which is approximately Gaussian (but not to high fidelity) over the frequency band 23 GHz to 29 GHz. The feed is physically small, polarisation insensitive and simple to produce with a minimum of machining operations. The beam width is not very sensitive to frequency. It is possible to reduce the frequency sensitivity by introducing a circular choke outside the radiating aperture, but the extra complexity is probably not warranted for this application (see final reflector results). Figure 4-7 and 4-8 show views of the feed. Figures 4-9 and 4-10 show the 3D patterns at the extremes of frequency range. Figures 4-11, 4-12, 4-13 and 4-14 show the principal cut patterns at 23 GHz, 25 GHz, 27 GHz and 29 GHz respectively when the feed is excited by an electric field source in the $\underline{\hat{x}}$ direction.

With this feed we need to (1) export the far-fields for use with a reflector using the integral equation solver in CST and (2) export the near fields on a plane $z = z_0$ for some constant z_0 for use with our physical optics software. This is because the different software packages have different requirements; the current version (version 12) of CST cannot employ near fields as an illumination source and the current version of our PO software (version 1.009) cannot employ tabulated far-fields as an illumination source. There are special requirements for both, as outlined below.

When exporting far fields it is essential that the phase centre of the equivalent far field source be correctly defined. A way to do this is to ensure that the global coordinate system of the feed is defined such that the aperture plane is centred on the origin of the global coordinate system. Next, ensure that the "far-field plot properties" in CST is set to use "local coordinate origin" and *not* the default "bounding box centre" under the "origin" tab. The "far-field plot properties" must also be set to 3D mode with sufficiently fine sampling. Theta and phi intervals of 2 degrees are used here, though this is not too critical. Once these are set, use the "save as source" button to save the exported file in the correct format. Note that we do not use the "phase centre" tab under "far-field plot properties" so the true (or at least best fit) phase centre of the feed may not coincide with our assumption that the phase centre is at the exit aperture. For our purposes this does not matter since we make the same assumption in both the PO software and in the CST comparison.

When generating the near fields, the field evaluation plane is set to $z = z_0$ a little way out from the aperture plane $z = 0$ so that the fields are not sharply varying near the aperture boundaries. $z_0 = 3.0$ mm in this example. CST is set to calculate both the electric and the magnetic fields on this plane which can be exported using the main export menu option when either the electric or magnetic field is illustrated on the main panel. Spatial sampling is sufficient at $\Delta x = \Delta y \leq 0.1\lambda$ (a tenth of a wavelength) but accuracy is almost as good at a fifth of a wavelength. In this example $\Delta x = \Delta y = 1.0$ mm which meets the sampling requirements over the frequency band. When $z \neq 0$ (as is the case here) it is important to include z_0 as an offset distance to the position of the feed specified in the physical optics

software, since the feed lies a distance z_0 behind the near field plane. Both the electric and magnetic field files are used to provide greatest accuracy.

The input file for the physical optics software is given by;

```
CALCOPTS 1
%%ANGLES  90.0, 0.0, 1, 00.0, 0.5, 360
ANGLES 00.0, 0.3, 600, 90.0, 0.0, 1
FREQS 23000.0 000.0 1
FILENAME test_out_elev junk_out
FEEDCEN 0.0 0.147 0.0
%%FEEDROT  90.000  -90.000  90.0
FEEDROT  90.000  -90.000  00.0
FEEDFILE 2 expo1_3.0mm_offset_23GHz_e-field.txt expo1_3.0mm_offset_23GHz_h-field.txt \
          23000.0 0.012 0.0120
PLOTFILE sideview1.ps 00.00  00.0 180.0  0.50 0.1 0.0 T F F F
PLOTFILE sideview2.ps 90.00  00.0 180.0  0.50 0.1 0.0 T F F F
SURFACE VARIPAR2D1 0.15 0.0 0.15 0.0 0.0 5 12.000 0.00 0.05 -0.00 2.0 -0.00  4.50 \
          6 0.65 0.00  0.07 0.0 0.00 0.005  0.015
BOUNDARY ellipse 0.20, 0.15, 0.0 0.00 00.0 0.007
GEOMFILE qqq_pj.dat  RW
```

This is defined for a feed position given by $y_{fl} = 150$ mm and a frequency of 23 GHz. Note that the FEEDCEN defines the position of the centre of the near field data array, hence the y-coordinate is set as 147.0 mm, since the near field is specified 3.0 mm from the aperture plane of the feed. Also note that the FEEDFILE frequencies must match the FREQS evaluation frequency.

As specified by the feed, the near fields are generated by a source in the local (feed) x-coordinate direction, so the dominant component of the electric field on the near field plane is in this same direction. Under the FEEDCEN rotation, this translates to a dominant electric field in the global (reflector) z-coordinate direction. (See later figures showing the geometry.) This means that the principal polarisation of the far field, in the global $y-z$ plane (for which ANGLES line is set) is $\underline{\hat{\theta}}-$polarised. Change of polarisation is obtained by rotating the feed by 90 degrees (uncommenting the %%FEEDROT line and commenting out the existing FEEDROT line). Change of scan angle to the global $x-y$ plane is obtained by commenting out the ANGLES line and uncommenting the %%ANGLES line. In summary (tabulated below) we have two principal planes that should be examined for each of two polarisations.:

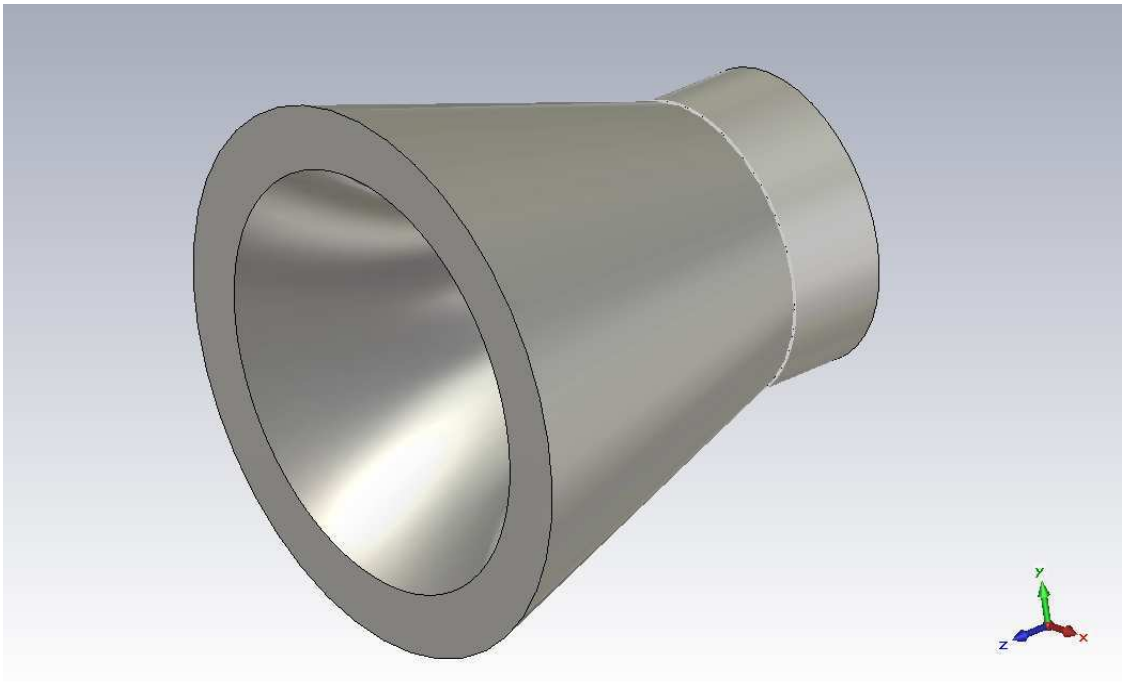

Figure 4-7: CST screen dump of the feed in profile, local (feed) coordinate system.

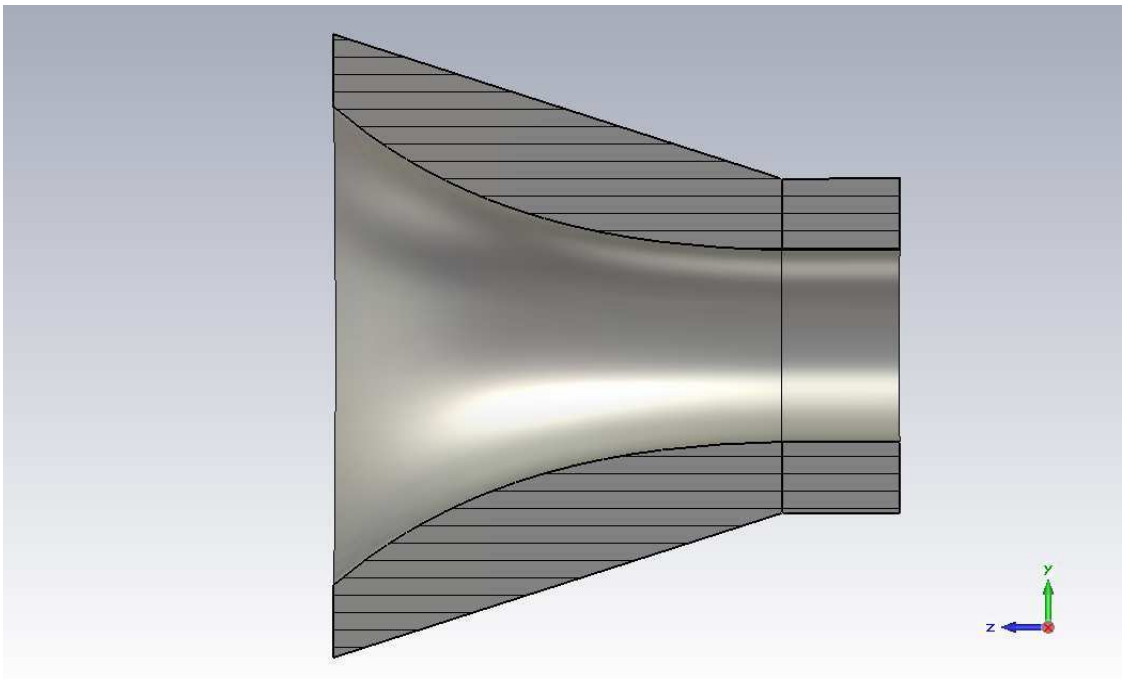

Figure 4-8: CST screen dump of the feed in cross section, local (feed) coordinate system.

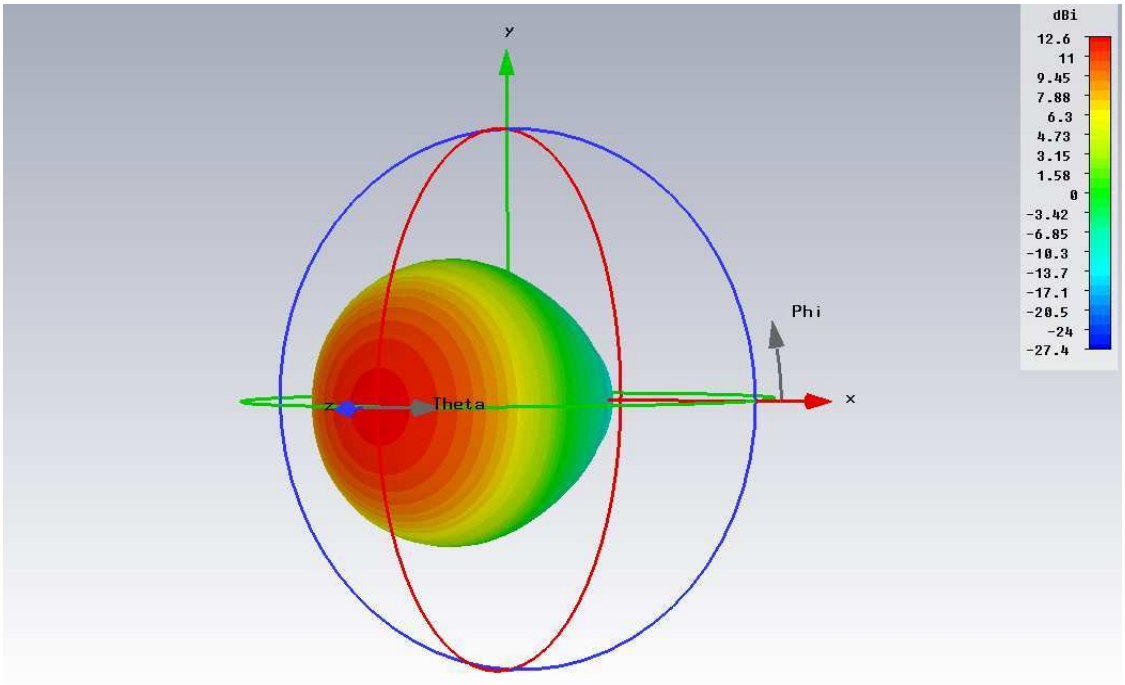

Figure 4-9: CST view of 3D far-field at 23 GHz in local (feed) coordinate system.

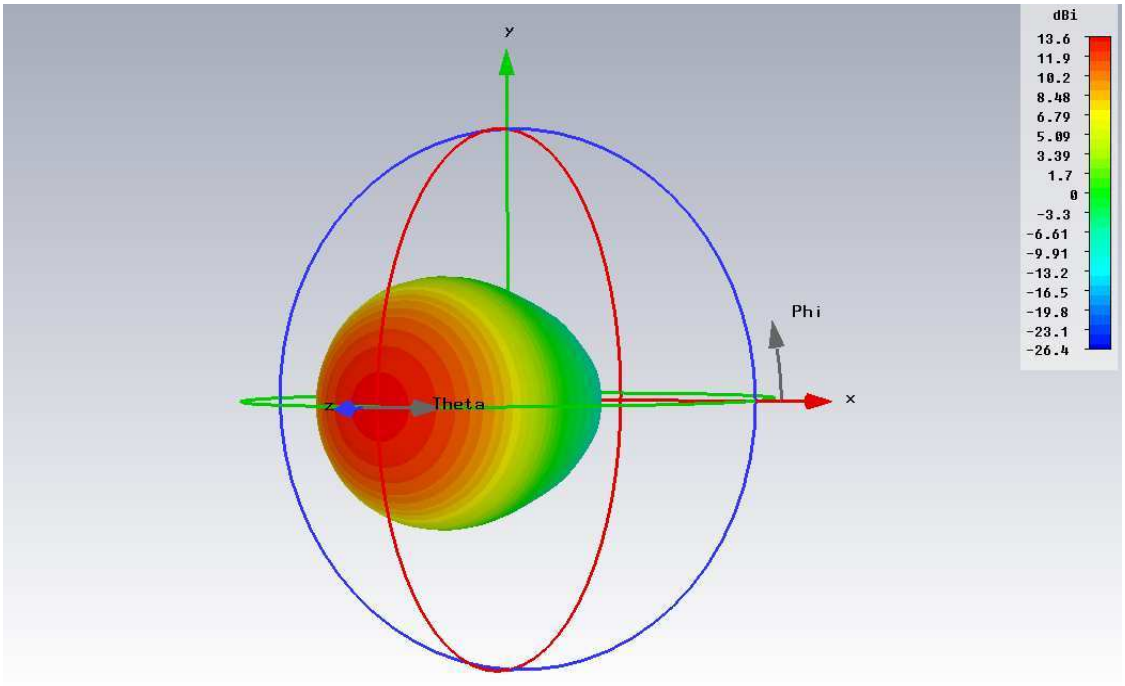

Figure 4-10: CST view of 3D far-field at 29 GHz in local (feed) coordinate system.

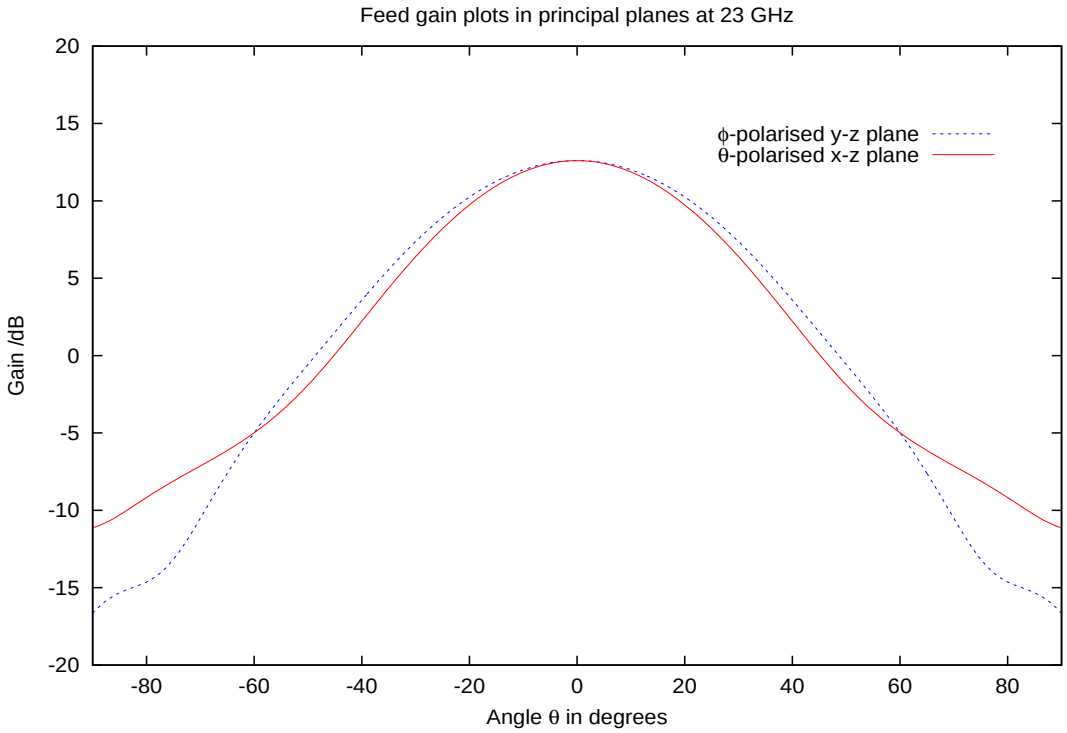

Figure 4-11: CST predicted feed gain in principal planes at 23 GHz, local (feed) coordinate system.

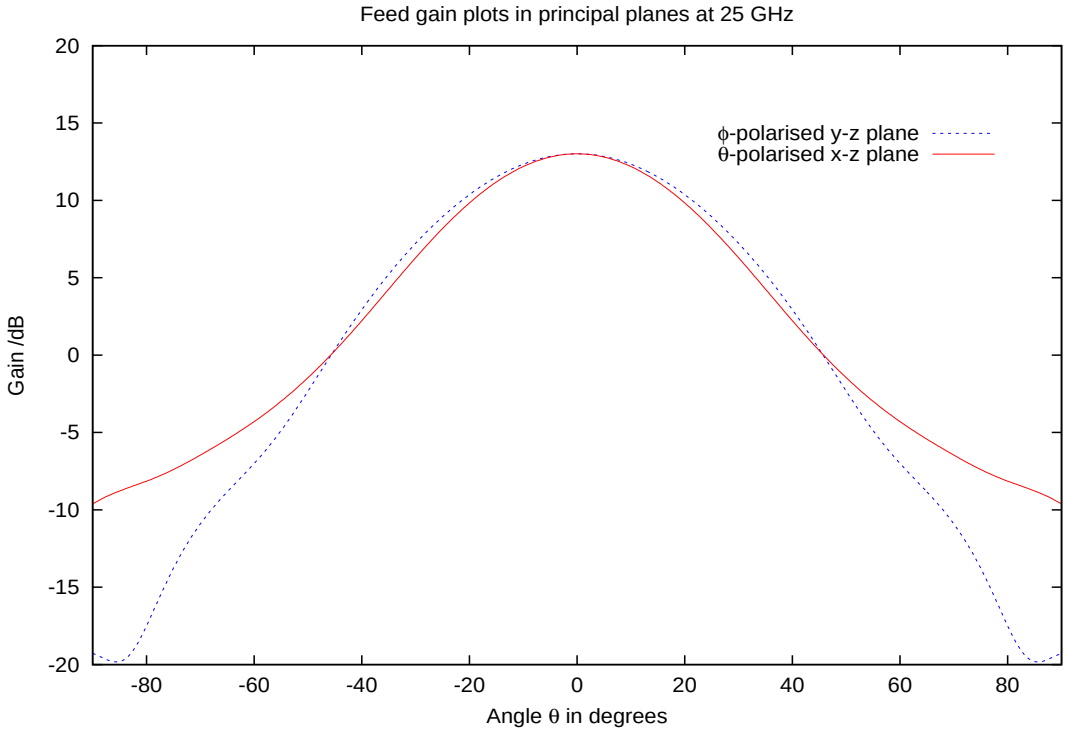

Figure 4-12: CST predicted feed gain in principal planes at 25 GHz, local (feed) coordinate system.

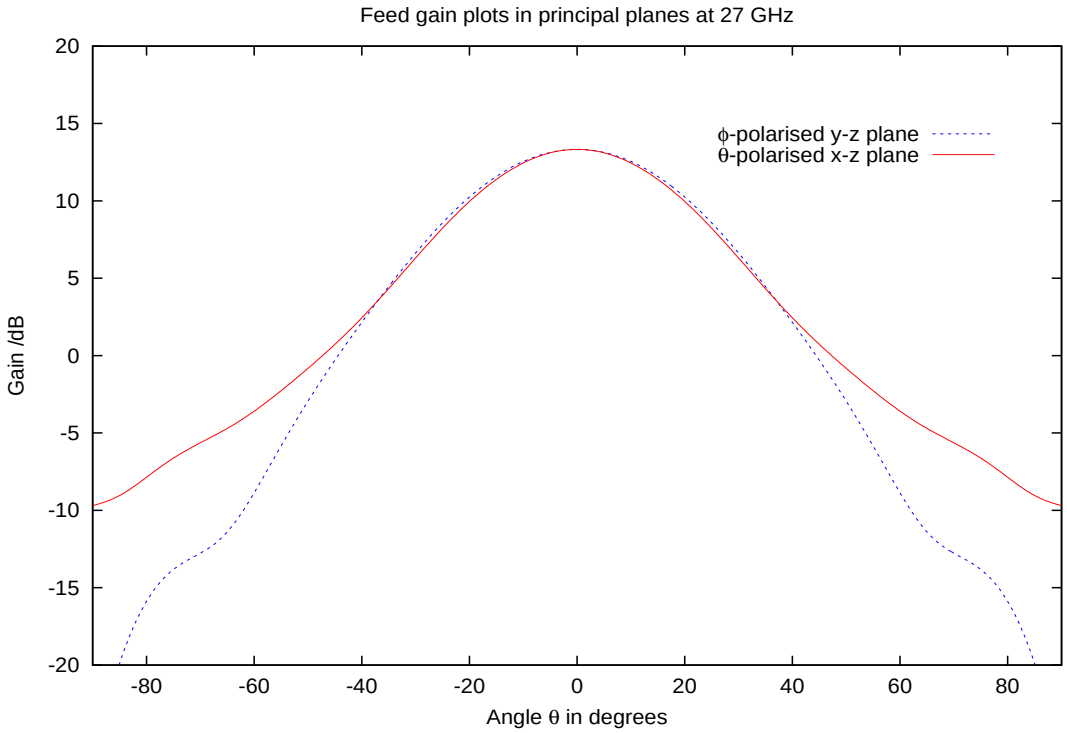

Figure 4-13: CST predicted feed gain in principal planes at 27 GHz, local (feed) coordinate system.

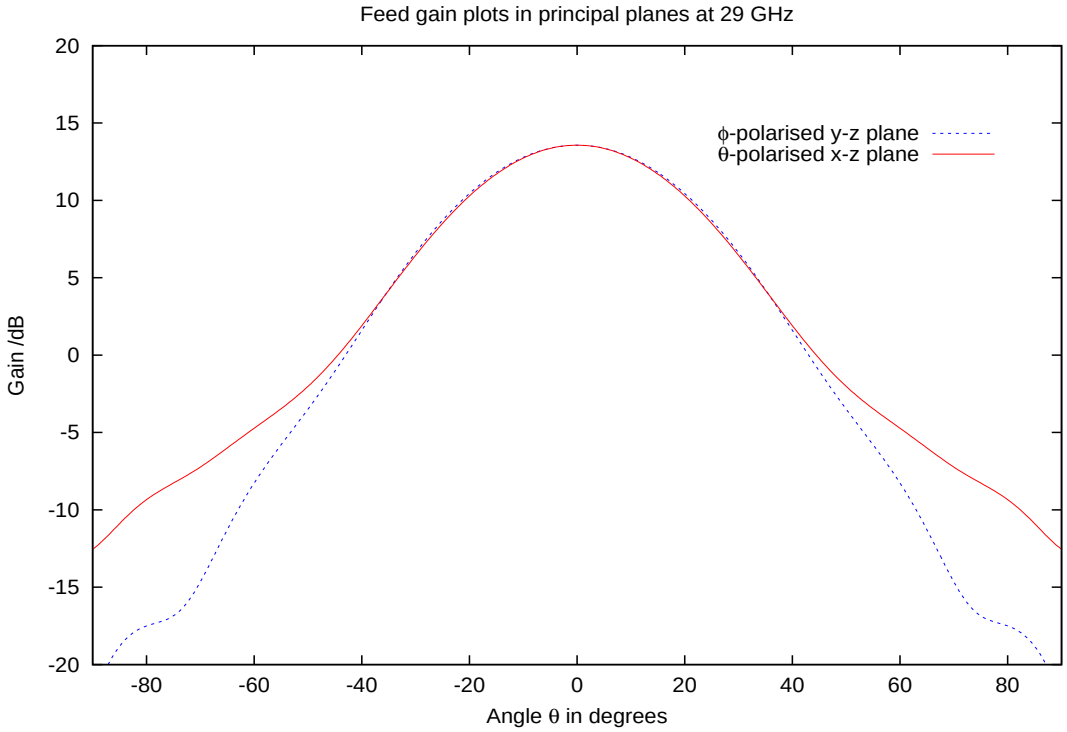

Figure 4-14: CST predicted feed gain in principal planes at 29 GHz, local (feed) coordinate system.

1. FEEDROT 90.000 -90.000 00.0
   ANGLES 00.0, 0.3, 600, 90.0, 0.0, 1
   'Horizontal plane' (x-y plane, global coordinate system), 'vertically polarised feed'
2. FEEDROT 90.000 -90.000 00.0
   ANGLES 90.0, 0.0, 1, 00.0, 0.5, 360
   'Vertical plane' (y-z plane, global coordinate system), 'vertically polarised feed'
3. FEEDROT 90.000 -90.000 90.0
   ANGLES 00.0, 0.3, 600, 90.0, 0.0, 1
   'Horizontal plane' (x-y plane, global coordinate system), 'horizontally polarised feed'
4. FEEDROT 90.000 -90.000 90.0
   ANGLES 90.0, 0.0, 1, 00.0, 0.5, 360
   'Vertical plane' (y-z plane, global coordinate system), 'horizontally polarised feed'

Computation time is a few seconds per simulation on a DELL Optiplex 390 PC. The integral equation solver in CST may also be used to predict the antenna gain patterns. In this case the far-field feed patterns are exported from the t-solver, imported into the integral equation solver and translated and rotated into the correct position and orientation. As described above, care must be taken to ensure the phase centre in in the expected position. The 'Shaped Reflector A' macro is used to generate the reflector to the same geometry as specified in the physical optics input file. This macro is described in section 4.2 and the resulting reflector is illustrated in figure 4-15. Default CST meshing is employed (version 12) of 5 cells per wavelength and use of the iterative solver. Computation time is about 5 minutes on a DELL PowerEdge T310 (Intel Xeon processor)[10].

The reflector shape has been designed so that the far field meets certain constraints. These are based on gain relative to maximum, but for illustrative purposes the envelopes are all shown with the same maximum value of 20 dBi. Figures 4-16 to 4-23 show comparisons at the extremes of the range at 23 GHz and 29 GHz for both principal planes and polarisations. Predictions at 25 GHz and 27 GHz are similar. Note that the PO predictions show the gain based on the field scattered by the reflector and do not show a direct contribution from the feed. Thus these plots show scattered rather than total field. The differences are significant only in the shadowed region of the reflector (azimuth angle $180^o < \phi < 360^o$, not plotted here) where PO is not valid. The CST predictions, on the other hand, show the gain based on the total field, i.e. including the direct contribution from the feed.[11]

[10] Q-Par CST machine PC13

[11] According to recent communications with CST, 17th Jaunary 2013. Note, however, that there are some troubling inaccuracies in the forward scatter region, if this is correct. We are still awaiting developments on this.

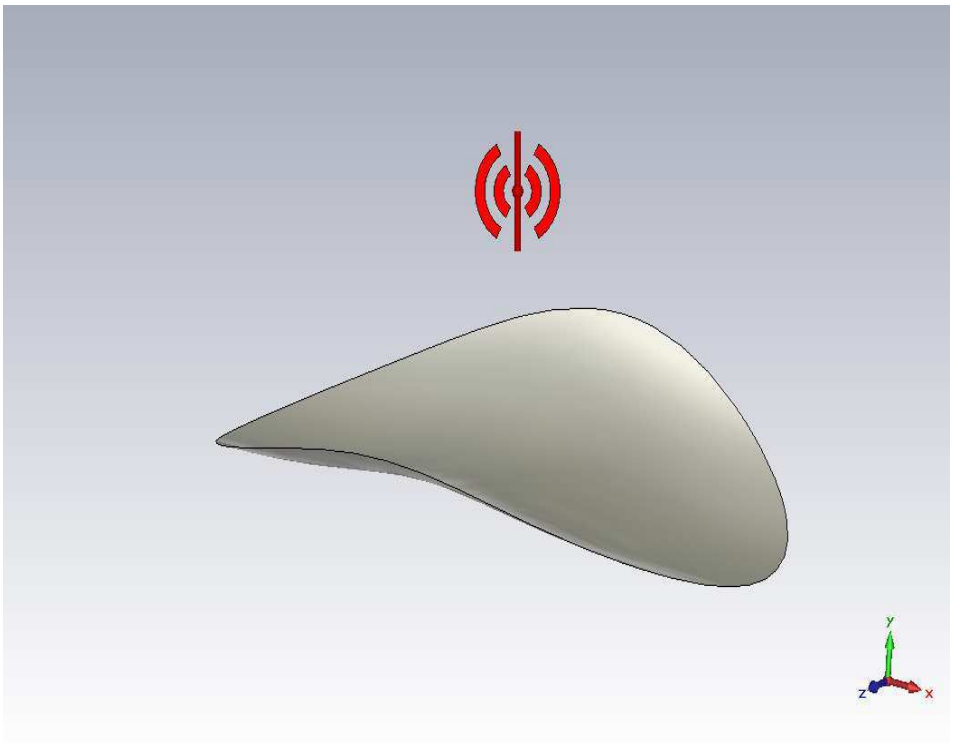

Figure 4-15: Screen dump of CST geometry, example 3.

Finally we show a PO comparison at 23 GHz using predictions using the same structure fed by a Gaussian beam (using the MGAUSSIAN feed option) with equivalent beam width at the -8.7 dB points. At 23 GHz this corresponds to a value of $\theta_0 \approx 37.7^o$. Results are shown in figures 4-24 and 4-25 for a horizontally polarised beam, comparing the real feed predictions (illustrated previously) with the Gaussian beam feed.

The predictions using the Gaussian beam and the real feed are similar and neither show the strong ripple at the low gain levels characteristic of the CST predictions. Agreement between the PO predictions and the CST predictions is acceptable though not as good as in the previous example. We attribute these differences to the smaller size of the reflector and its higher degree of curvature.

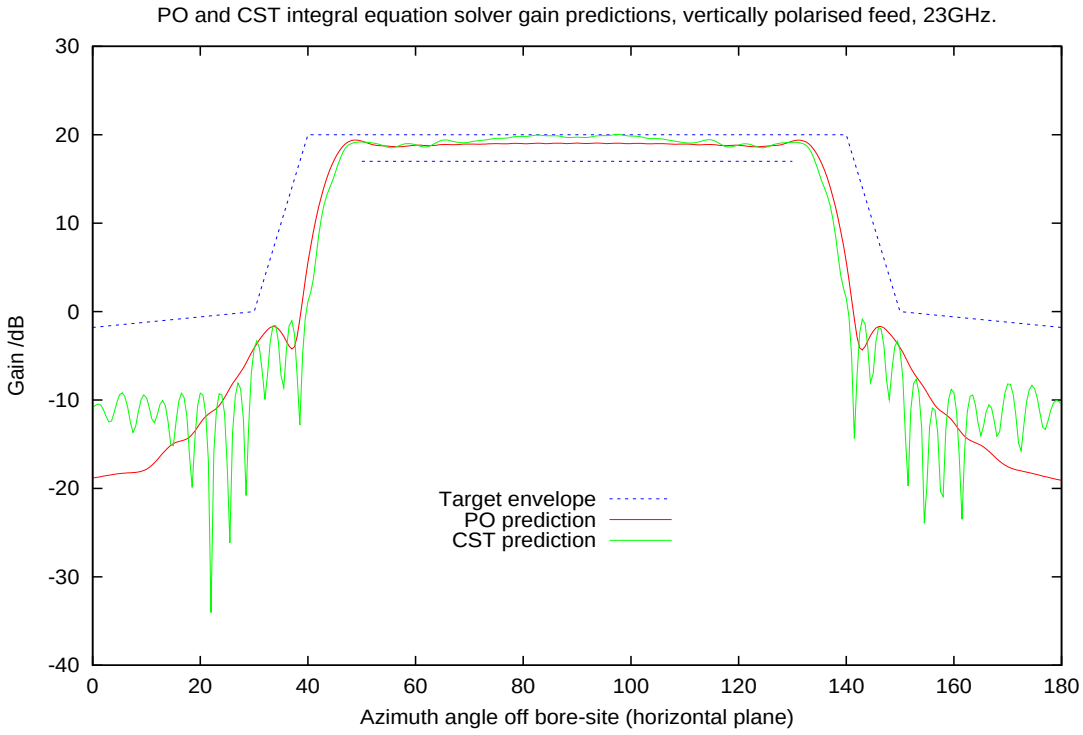

Figure 4-16: Absolute gain comparison, 23 GHz, vertically polarised feed, principal horizontal plane

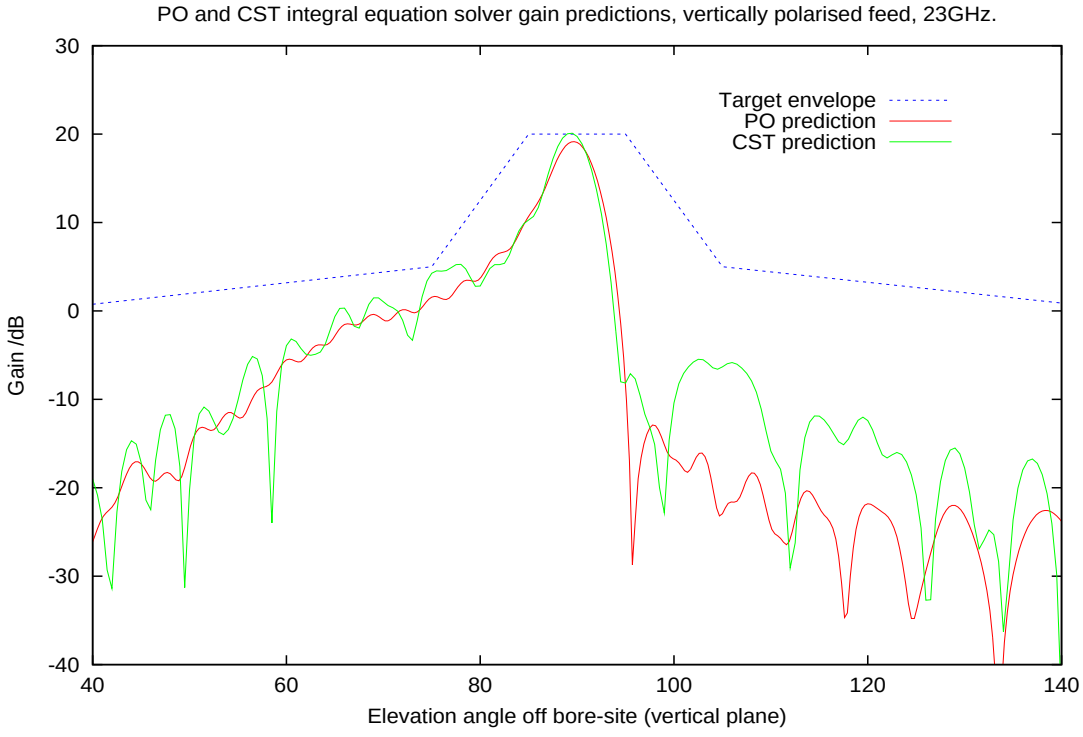

Figure 4-17: Absolute gain comparison, 23 GHz, vertically polarised feed, principal vertical plane

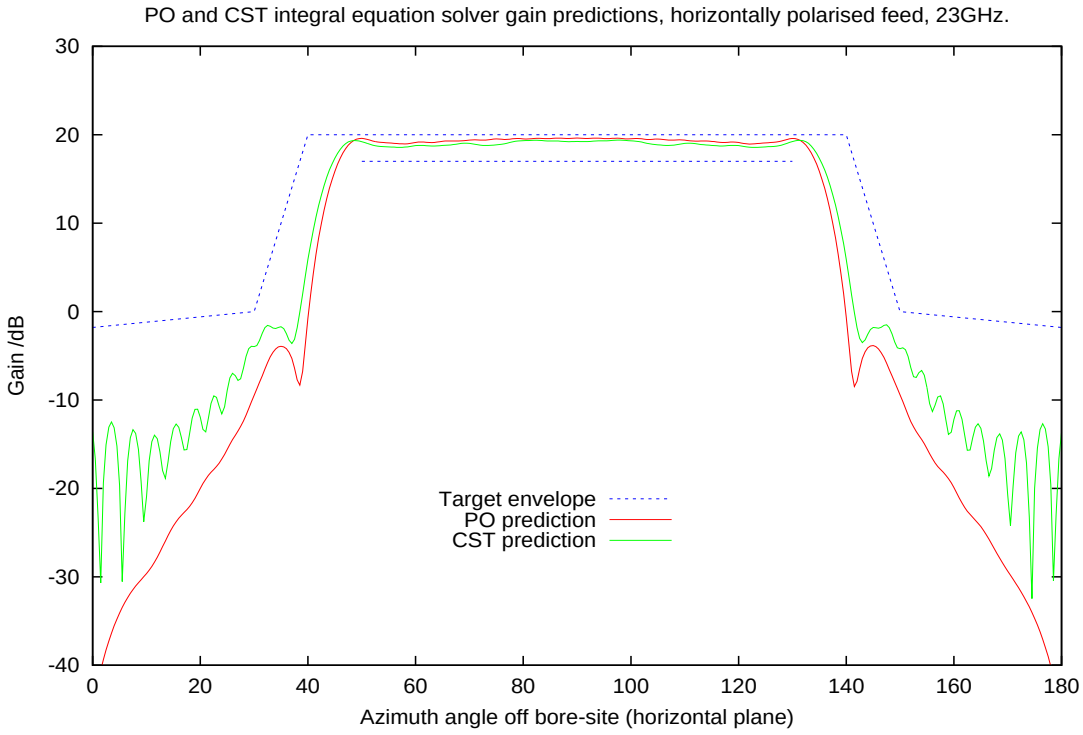

Figure 4-18: Absolute gain comparison, 23 GHz, horizontally polarised feed, principal horizontal plane

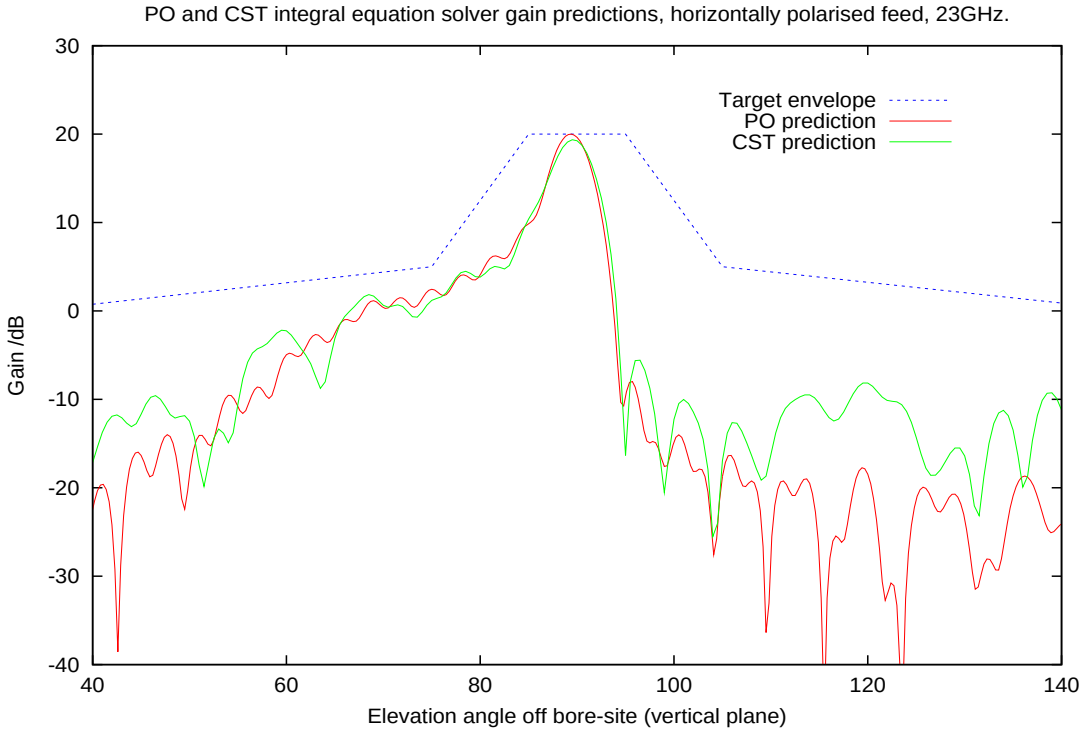

Figure 4-19: Absolute gain comparison, 23 GHz, horizontally polarised feed, principal vertical plane

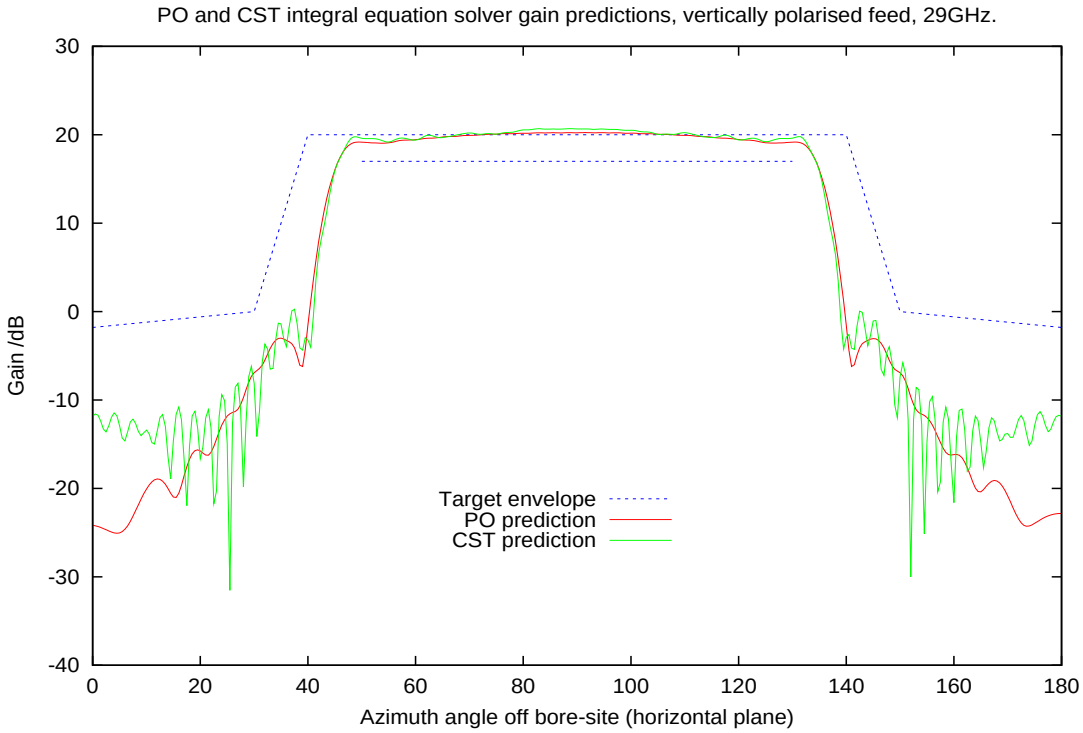

Figure 4-20: Absolute gain comparison, 29 GHz, vertically polarised feed, principal horizontal plane

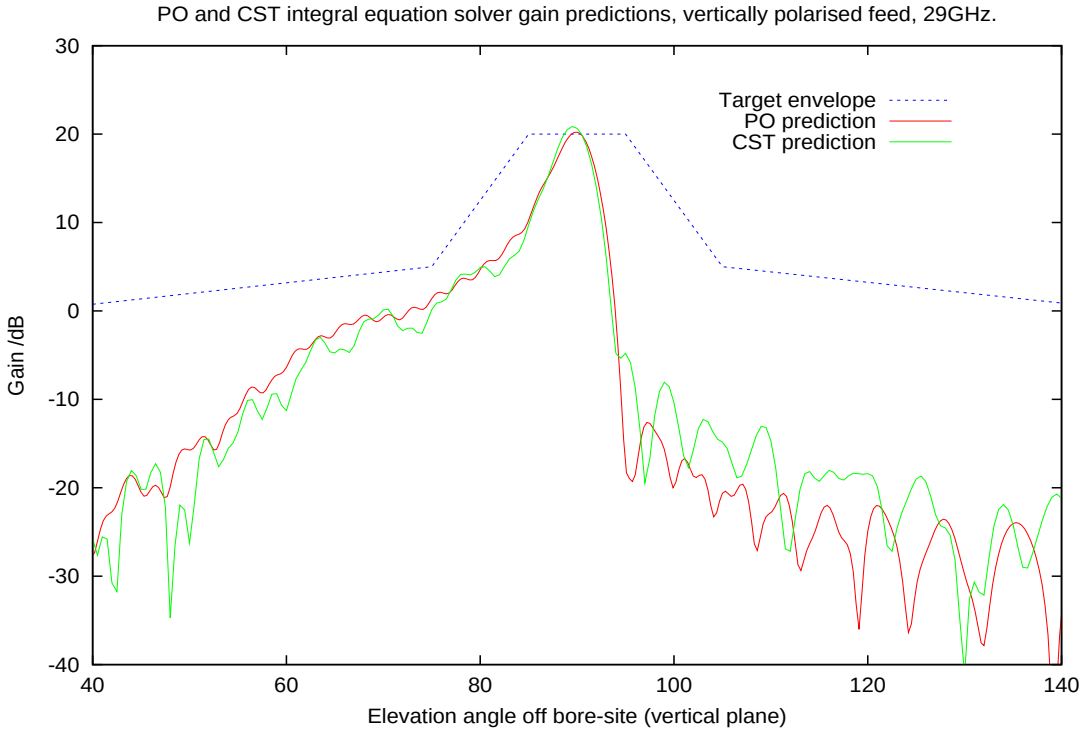

Figure 4-21: Absolute gain comparison, 29 GHz, vertically polarised feed, principal vertical plane

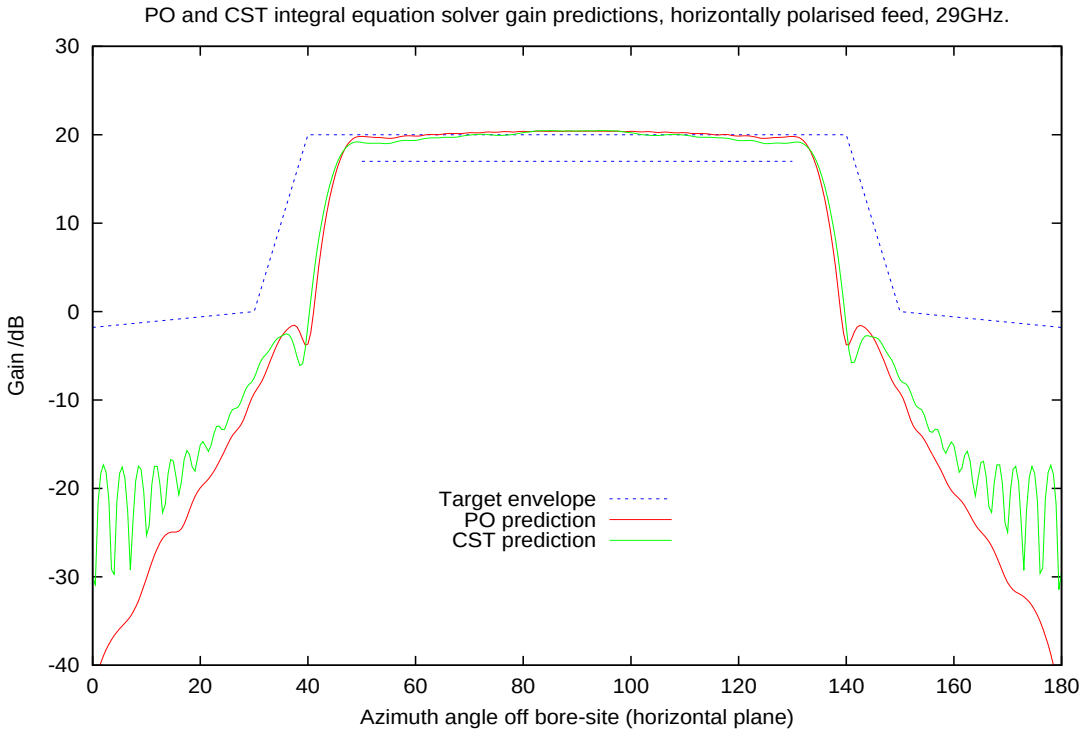

Figure 4-22: Absolute gain comparison, 29 GHz, horizontally polarised feed, principal horizontal plane

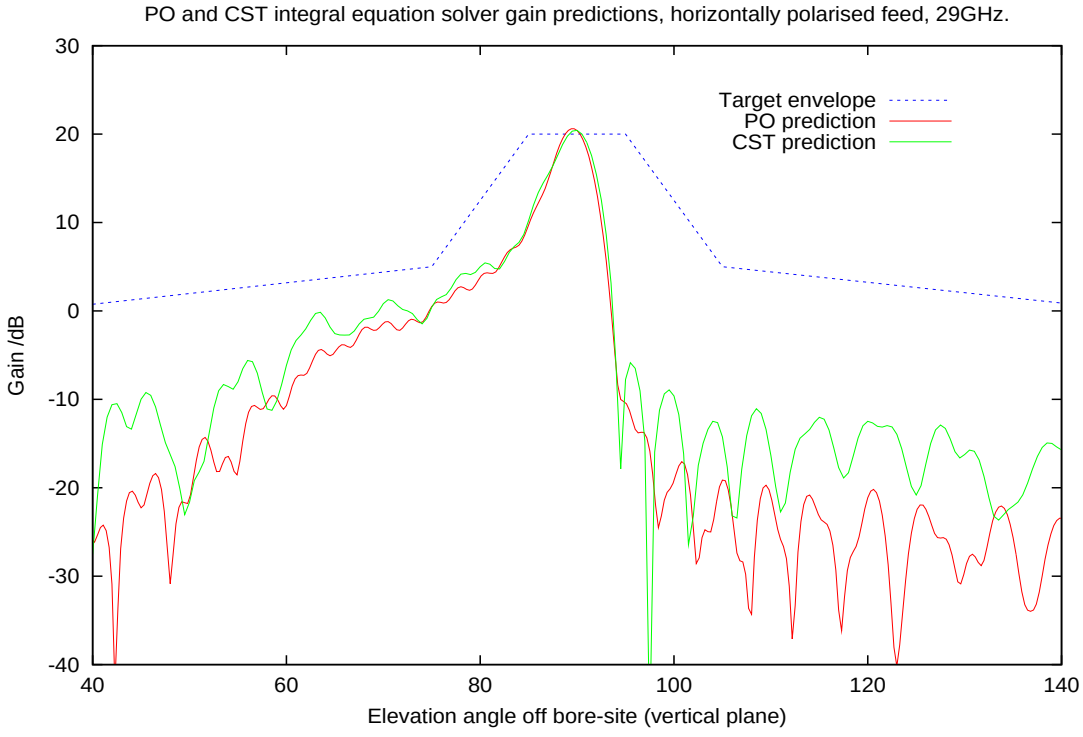

Figure 4-23: Absolute gain comparison, 29 GHz, horizontally polarised feed, principal vertical plane

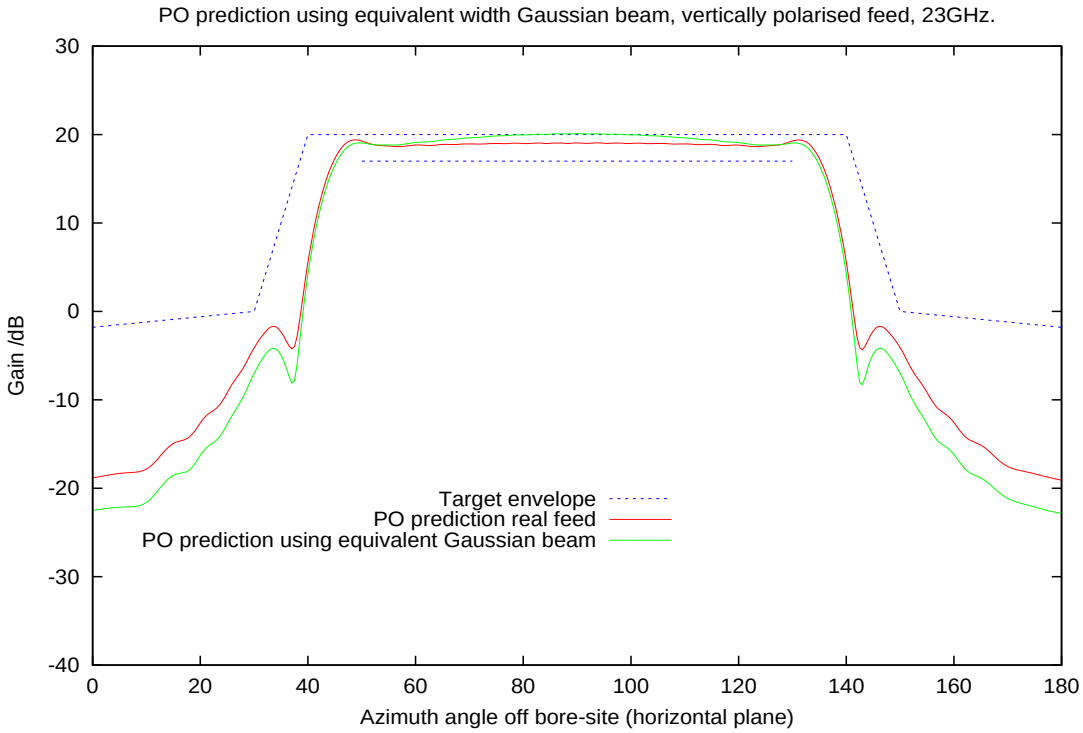

Figure 4-24: Absolute gain comparison PO using real and Gaussian beam feeds, 23 GHz, vertically polarised feed, principal horizontal plane

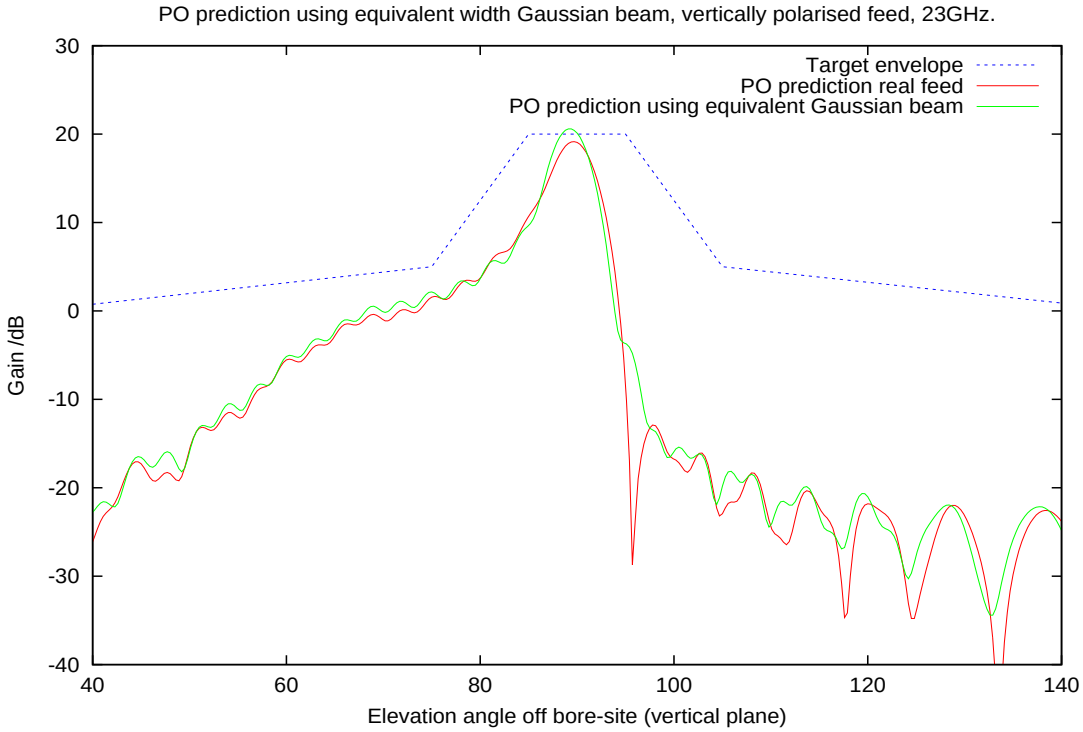

Figure 4-25: Absolute gain comparison PO using real and Gaussian beam feeds, 23 GHz, vertically polarised feed, principal vertical plane

4.8 Example 4. Use of an optimiser

In this section we employ the same feed as described in example 3 and employ an optimiser [8] to aid in the control of the shape of the reflector. For this purpose we hand modify the shape parameters that principally control the vertical cross section of the reflector, i.e. the parameters α_i and β in the *VARIPAR2D1* surface description and use the optimiser to control the four most important a_i parameters for azimuth shaping. Since the feed is designed to have a beam shape that is relatively insensitive to frequency it is legitimate to choose the optimisation based on a single frequency. For our purposes we will optimise at 25 GHz on the beam shape in the horizontal (azimuth) plane, assuming the feed is vertically polarised (according to the example 3 definitions). Once the optimised a_i parameters are obtained we use them to make additional predictions at 23 GHz, 27 GHz and 29 GHz.

The optimiser requires the user to define a job-specific module written in Fortran, 'costfunction.f', which is printed verbatim below.

```
C     ******************************************************************
C     ******************************************************************
C     THIS ROUTINE IS APPLICATION SPECIFIC AND SHOULD BE
C     WRITTEN BY THE USER FOR A GIVEN OPTIMISATION PROBLEM
C     ................................................
C     INPUTS:
C     .........
C     iolog   = integer channel number for output warnings and logs
C               (recommend leave this alone - do not change)
C     icoltot = number of columns defined in input file from
C               which cost function can be constructed. These are
C               taken as the ordered set taken from the total number
C               of output columns in the output file defined in input file
C     irowtot = total number of legal rows in output file
C     vdata   = output data array vdata(irowtot,icoltot)
C     iu      = integer record of number of calls to routine
C               (recommend leave iu alone)
C
C     OUTPUTS:
C     .........
C     costfn  = the cost function. The optimiser will try to minimise this
C               quantity so it should be positive with perfection achieved
C               when it takes the value zero.
C     stag    = user defined tag to be output in log file on first use
C     iu      = integer record of number of calls to routine
C               (recommend leave iu alone)
C     ******************************************************************
C     ******************************************************************

      SUBROUTINE costf_user(iolog,icoltot,irowtot,vdata,costfn,iu,stag)
      IMPLICIT none
      INCLUDE 'parameters.inc'
      INTEGER iolog,icoltot,irowtot,iu
```

```
      DOUBLE PRECISION vdata(MAXOUTLINES,MAXCOLUMNS)
      DOUBLE PRECISION costfn
      CHARACTER*(MAXLETS) stag

C     ...local declarations (user defined)
      DOUBLE PRECISION ab0, ab1, ab2
      DOUBLE PRECISION costlarge, weight
      DOUBLE PRECISION gtot, angle_deg, gain, gain_db
      DOUBLE PRECISION gain_av, gain_av_db
      DOUBLE PRECISION cost, marginfactor, costp, ang, gainf
      INTEGER igtot, i
      iu=iu+1      !...record of number of calls to this routine

C     ...define name of this cost function
      stag='PO optimiser for flat gain central region response, v1.2'

C     ...check on number of columns
      if (icoltot.ne.2) then
        write(iolog,*) '...ERROR in user cost function. Two',
     &                 ' columns of data expected.'
        stop
      endif



C     ...angle breaks (must be increasing)
      ab0=45.0     !...region end where gain should be as level as possible
      ab1=47.5     !...region start when gain should start to fall
      ab2=60.0     !...region end where gain should have fallen by 20 dB

C     ...value for extreme penalty on cost (large)
      costlarge=1.0d+10
C     ...weight on reducing gain in side regions
      weight=0.1

C     ...form average gain over central region
      gtot=0.0
      igtot=0
      do i=1,irowtot
        angle_deg=vdata(i,1)-90.0   !...angle offset assuming broadside at 90 degrees
        if (angle_deg.ge.-ab1 .and. angle_deg.le.ab1) then
          gain=10.0**(0.1*vdata(i,2)) !...magnitude scale
          !
          gtot=gtot+gain
          igtot=igtot+1
        endif
      enddo
      if (igtot.lt.1) then
        write(iolog,*) '...APPLICATION ERROR. ',
     &                 'No data in central region.'
        stop
      endif
```

```
      gain_av=gtot/real(igtot)
      gain_av_db=10.0*log10(gain_av)
C     write(udatlog,*) 'average gain central region = ',gain_av_db,
C    &          '  over ',igtot,' entries'

C     ...form difference from average and add to cost
      cost=0.0
      marginfactor=2.0*(ab2-ab1)*0.1 !...envelope control for outligh region
      do i=1,irowtot
        angle_deg=vdata(i,1)-90.0   !...angle offset assuming broadside at 90 degrees
        gain_db=vdata(i,2)          !...gain in dB
        gain=10.0**(0.1*gain_db) !...magnitude scale
        !
        if ((angle_deg.ge.-ab0 .and. angle_deg.le.ab0) .and.
     &       (angle_deg.ge.-90.0 .and. angle_deg.le. 90.0) ) then
          costp=(gain_av-gain)**2
          cost=cost+costp          !...cost function uses magnitude (not in dB)
        elseif (angle_deg.lt.-ab1 .and. angle_deg.gt.-ab2) then
          ang=angle_deg+ab1          !<0
          gainf=gain_av_db+ang*marginfactor      !...assumes gain in dB and angle in degrees
          if (gain_db.gt.gainf) then
            cost=cost+weight*(gain_db-gainf)**2  !...drive down
          else
            cost=cost+weight*gain    !...the smaller the gain over this region the better
          endif
        elseif (angle_deg.gt.ab1 .and. angle_deg.lt.ab2) then
          ang=-angle_deg+ab1          !<0
          gainf=gain_av_db+ang*marginfactor      !...assumes gain in dB and angle in degrees
          if (gain_db.gt.gainf) then
            cost=cost+weight*(gain_db-gainf)**4  !...drive down
          else
            cost=cost+weight*gain    !...the smaller the gain over this region the better
          endif
        endif

      enddo
      costfn=cost

      RETURN
      END
```

The essential requirement is that there is a significant penalty for exceeding the edge region around the central region and a significant penalty for moving away from a flat response over the central region. There are any number of ways in which this may be done; this is just one.

The optimiser is launched via the command: *optimiz <control_file>*, where the control file takes the form,

```
testpo
test_po7_azim.dat
""
"y"
test_out_azim
2  5  6

10 17 0.4 0.9
10 19 0.0 0.5
10 21 0.0 0.2
10 23 0.0 0.05
```

which expresses the fact that four variables from line 10 at string positions 16, 18, 20 and 22 (from the line beginning SURFACE VARIPAR2D1) in the input file shown below, will be under optimiser control. Details of the syntax and optimiser use are given in [8].

Once the optimiser has completed its task, the optimised PO input file takes the form,

```
CALCOPTS  1
ANGLES  90.0  0.0  1  00.0  0.5  361
FREQS  25000.0  000.0  1
FILENAME  test_out_azim  junk_out
FEEDCEN  0.0  0.147  0.0
FEEDROT  90.000  -90.000  00.0
FEEDFILE  2  expo1_3.0mm_offset_25GHz_e-field.txt  expo1_3.0mm_offset_25GHz_h-field.txt \
            25000.0  0.012  0.0120
PLOTFILE  sideview1.ps  00.00  00.0  180.0  0.50  0.1  0.0  T  F  F  F
PLOTFILE  sideview2.ps  90.00  00.0  180.0  0.50  0.1  0.0  T  F  F  F
SURFACE  VARIPAR2D1  0.15  0.0  0.15  0.0  0.0  5  15.000  0.00  0.25  -0.00  -3.0  -0.00  5.5 \
            6  0.575E+00  0.00  0.662E-01  0.0  0.177E-01  0.0  0.839E-02
BOUNDARY  ellipse  0.20  0.15  0.0  0.00  00.0  0.007
GEOMFILE  qqq_pj.dat  RW
```

where the controlled parameters have been rounded to 3 significant digits. The same SURFACE VARIPAR2D1 parameters are then used to make predictions using the relevant FEEDFILE files at other frequencies.

Side views of the reflector, as defined by the PLOTFILE lines are illustrated in figures 4-26 and 4-27. The azimuth and elevation absolute gain predictions are shown in figures 4-28 and 4-29. Note that the target envelope is shown with a maximum value of 19 dBi (it was shown as 20 dBi in Example 3) and the elevation plots now cover the range 30^o to 150^o.

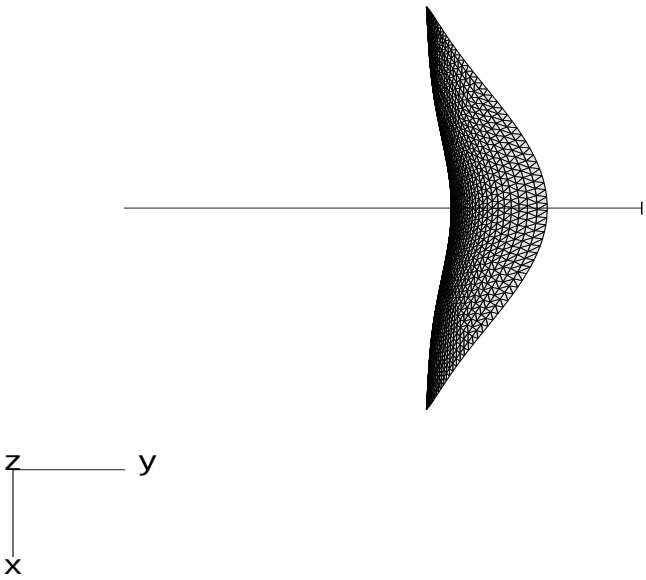

Figure 4-26: Cross section of the shaped reflector, horizontal cut.

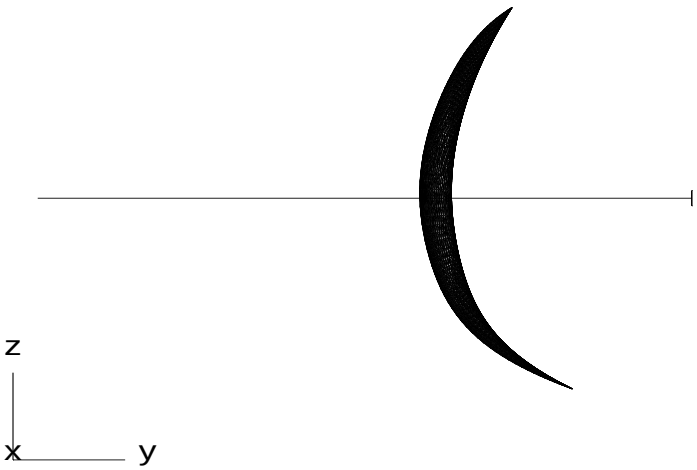

Figure 4-27: Cross section of the shaped reflector, vertical cut.

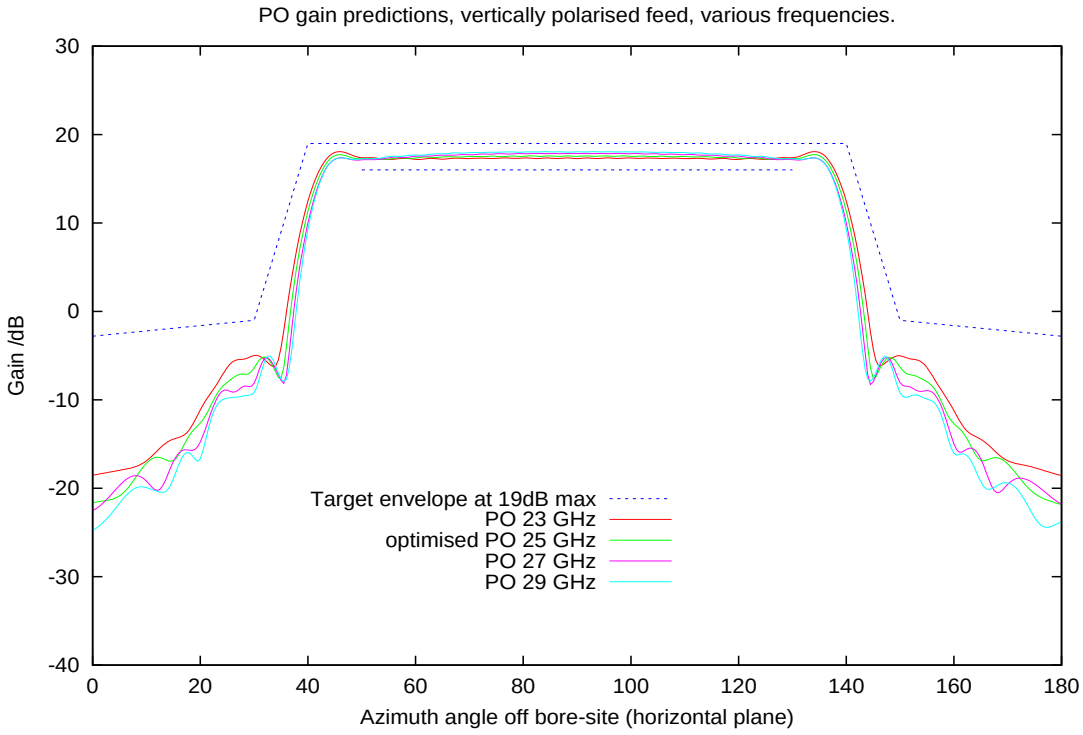

Figure 4-28: Predicted PO gain in dBi based on horizontal plane optimisation at 25 GHz. Vertically polarised feed, horizontal plane

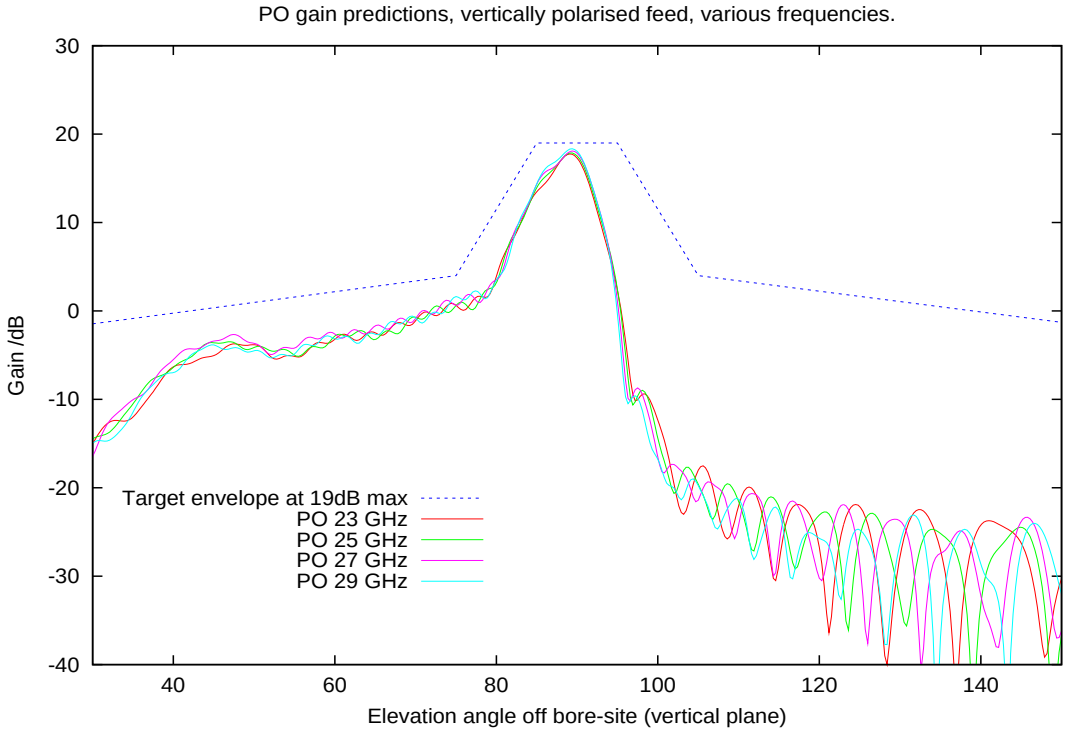

Figure 4-29: Predicted PO gain in dBi based on horizontal plane optimisation at 25 GHz. Vertically polarised feed, vertical plane

5 References

1 *TICRA report ed. K. Pontoppidan* 'Technical description of GRASP 8', version 8.2.5, TICRA, March 2002.

2 *J. Synge, B. Griffith* 'Principles of Mechanics', McGraw-Hill, 1970.

3 *W.H.Press, W.T.Vetterling, S.A.Teukolsky, B.P.Flannery* 'Numerical Recipes in Fortran', Cambridge University Press, 1986.

4 *CST EM simulation software*. See web site www.cst.com.

5 *S. Silver* 'Microwave antenna theory and design', McGraw-Hill, 1949.

6 *A.T. Friberg, T. Jaakkola, J. Tuovinen* 'Electromagnetic Gaussian beam beyond the paraxial regime', IEEE Trans Anten and Prop., Vol. 40, No. 8, pp 984-989, August 1992

7 *Solidworks 3D CAD design software*. See website www.solidworks.co.uk.

8 *A. J. Mackay* 'General use simplex optimizer - user guide', Q-par Technical report Q-par/AJM/optimiz/v1.3, January 2013.