A PROJECT REPORT ON

ALGORITHMS FOR PHASED ARRAY BEAM GENERATION

PUJA VACATION PROJECT CUM TRAINING
AT GMRT, KHODAD , NCRA ,TIFR

BY
SOMPROVA NANDY & SWAGATA CHAKRABARTI

B.TECH FINAL YEAR STUDENTS OF
INSTITUTE OF RADIOPHYSICS & ELECTRONICS
UNIVERSITY OF CALCUTTA
92,A.P.C Road . Kolkata - 700009

UNDER THE GUIDANCE OF
Dr. YASHWANT GUPTA
(Chief Scientist – GMRT)
&
Mr. JAYANTA ROY

From
23rd September 06 to 3rd November 2006

National Centre for Radio Astrophysics
TATA INSTITUTE OF FUNDAMENTAL RESEARCH
NCRA, PUNE UNIVERSITY CAMPUS, POST BAG 3, GANESHKHIND, PUNE 411 007, INDIA
Telephone + 91 020 2569 7107, 25691382, 2569 1384, 2569 1385;
Fax +91 020 2569 2149 Gram RASTRON; E-mail: res@ncra.tifr.res.in

CERTIFICATE

This is to certify that Miss Swagata Chakrabarti and Miss Somprova Nandy students of Institute of Radio Physics & Electronics, C.U. have carried out the project entitled "ALGORITHM FOR PHASED ARRAY BEAM GENERATION" satisfactorily under my guidance during the puja vacation 2006 and submitted the project report as per the requirement of GMRT.

Project Guide.
Dr. Yashwant Gupta
Chief Scientist
GMRT, Khodad
NCRA, TIFR
PUNE – 410 504.

ACKNOWLEDGEMENT

We want to express our deep gratitude to our project guide Dr. (Prof) Yashwant Gupta (Chief Scientist – GMRT) for his valuable guidance in helping us to accomplish this project assignment. Discussions with him on the physical and practical aspects of our work has helped us delve more into the physics of the subject and understanding of the experimentally obtained results. We are also thankful to our co-guide Mr Jayanta Roy (Research Associate) who guided us through the finer aspects of C coding and operation on LINUX.

We are also thankful to Mr S. Sabhapathy (FPS, Servo System) who has been a well-wisher throughout . Our successful completion of project is also the result of constant help from Mr. Sandeep C Choudhari (IF/LO/FE System) and Sumit Mirajkar (System Admin).

Our cordial thanks to all who have contributed directly or indirectly towards the completion of our assignment.

Somprova Nandy
Swagata Chakrabarti

ABSTRACT

We are to develop an algorithm that will generate phased array beams for any symmetric or unsymmetric antenna array configuration. The array factor derived theoretically is used to find the position of nulls and peaks of the power pattern.

The linear array is to be phased for broadside and end fire case and the Beam Width at First Null obtained experimentally from graph and derived from theory are compared. The match of the two will exhibit the proper working of the algorithm.

Next,the aspect of multiple beam generation over a single dimension or over a 2-D sky plane will be developed. The study is broadly done over the GMRT central square and the arm antenna.

CONTENTS

Introduction to GMRT

The GMRT consists of 30 45 m diameter parabolic dishes spread over 25 km near the village of Khodad, about 80 km north of Pune, India. The site coordinates are:

```
latitude  = 19 deg 05 min 48 sec North,
longitude = 74 deg 03 min 00 sec East,
altitude  = 588 m.
```

There are fourteen telescopes randomly arranged in the central square, with a further sixteen arranged in three arms of a "Y"-shaped array (similar to the VLA) giving an interferometric baseline of about 25 km. The GMRT is an interferometer which uses a technique known as aperture synthesis to make image.

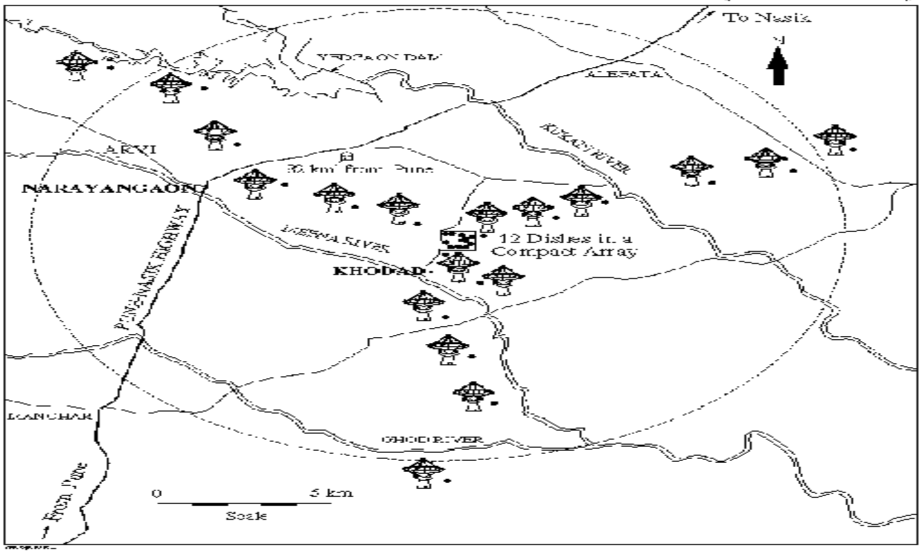
Each antenna is 45 metres in diameter and, instead of a solid surface like many radio telescopes, the reflector is made of wire rope stretched between metal struts in a parabolic configuration. This works because of the long wavelengths (21 cm and longer) at which the telescope operates. Each antenna has four different receivers mounted at the focus. Each individual receiver assembly can rotate so that the user can select the frequency at which to observe.

The maximum baseline in the array gives the telescope an angular resolution (the smallest angular scale that can be distinguished) of about 1 arcsecond at the frequency of neutral hydrogen (1420 MHz).

Astronomers from all over the world regularly use this telescope to observe many different astronomical objects such as galaxies, pulsars and supernovae.

Why metre wavelengths : The metre wavelength part of the radio spectrum has been particularly chosen for study with GMRT because man-made radio interference is considerably lower in this part of the spectrum in India. Although there are many outstanding astrophysics problems which are best studied at metre wavelengths, there has, so far, been no large facility anywhere in the world to exploit this part of the spectrum for astrophysical research

LOCATIONS OF GMRT ANTENNAS (30 dishes)



GMRT Antenna Specifications

The large size of the parabolic dishes implies that GMRT will have over three times the collecting area of the Very Large Array (VLA) in New Mexico, USA which consists of 27 antennas of 25 m diameter and is presently the world's largest aperture synthesis telescope operating at centimetre wavelengths. At 327 MHz, GMRT will be about 8 times more sensitive than VLA because of the larger collecting area, higher efficiency of the antennas and a substantially wider usable bandwidth because of the low level of man-made radio interference in India.

Electronic Frontends and Backends : Apart from the novel low-cost design of the parabolic dishes, the instrument has state-of-the-art electronics systems developed indigenously and consisting of the following main sub units.

- Antenna feeds at six different frequency bands between 50 MHz and 1500 MHz, having good polarization characteristics as well as simultaneous multiband operation.
- Low-noise amplifiers, local oscillator synthesizers, mixers, IF amplifiers.
- Optical fibres linking the entire array with the CEB. These are used both for the telemetry signals and local oscillator phase reference communication between the CEB and each antenna base.
- A digital 2,30,000-channel FX-type correlator providing upto 128 spectral channels and covering a maximum bandwidth of 32 MHz

CHAPTER 1

INTRODUCTION TO LINEAR ARRAY

This project involves the testing and modification of an algorithm for phased array beams. The platform OS is used is LINUX and program coded in C.

Phased Array: Group of antennas in which relative phases of respective signals feeding the antennas are varied in such a way that effective radiation pattern of array is reinforced in desired direction and suppressed in desired direction.

The program given to us gmrt_arraypattern.c was expected to calculate the array factor for any given array.

For testing the code, we started with the simple linear array case.
For that let us derive the array factor for linear array of n elements of equal amplitude and spacing.

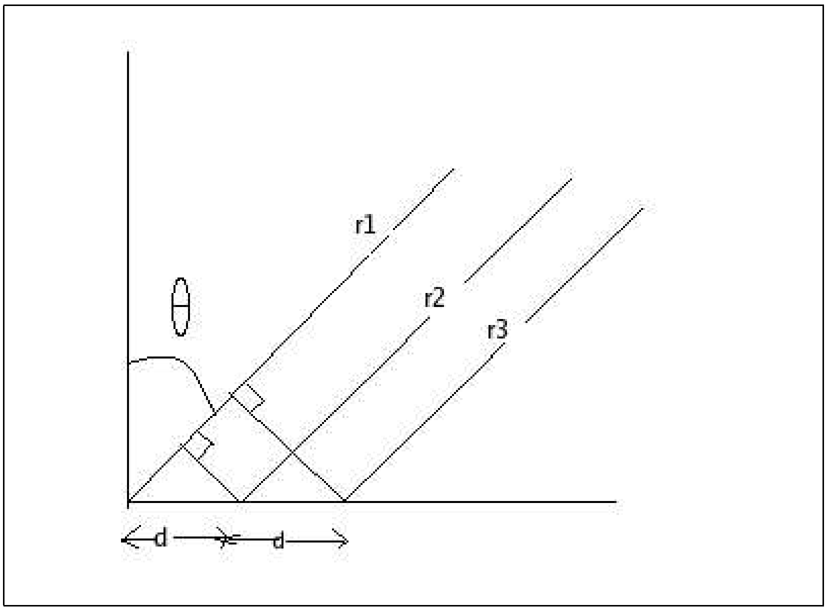


fig1.1

A uniform array is defined by uniformly spaced identical elements of equal magnitude with linearly progressive phase from element to element.

$\phi_1 = 0$, $\phi_2 = \alpha$, $\phi_3 = 2\alpha$, $\phi_N = (N-1)\alpha$

Inserting this linear phase progression into formula for general N element array

Array Factor= AF = 1 + exp j(α + kd sin θ) + exp 2j(α + kd sin θ)
+ ....+exp j(N-1)(α + kd sin θ)

Since $r_1 = r$
$r_2 = r - d \sin \theta$
$r_3 = r - 2 d \sin \theta$

$$\text{Therefore, AF} = \sum_{n=1}^{N} \exp j(N-1)\psi \quad \text{---(1)}$$

Now ψ = f (element spacing , phase shift , frequency , θ)
If position of array is shifted so that the centre of the array is located at origin,

$$AF = [\sin (N\psi / 2) / \sin (\psi / 2)]$$

fig1.2

For main lobe , $\Psi = 0$

$$\mathbf{AF\ (max) = N}$$

So the normalized array factor is

$$\mathbf{AF = [\sin (N\psi / 2) / \sin (\psi / 2)] / N} \quad \text{-------------------(2)}$$

Nulls are determined by 0 in the numerator of equation 3 while denominator $\neq 0$
Position of nulls are given by:

$\theta_n = \sin^{-1}[(-\alpha \pm 2n\Pi/N)/(kd)]$ ------------------------------------(3)

If $\lambda = d$; and for broadside array i.e $\theta = 0$, $\Phi = 0$ where the array is in x axis. In this case all elements are in phase and maxima occurs at $\theta = 0$ (when $\Psi = 0$)

$$0 = \alpha + kd\sin\theta$$
$$\Longrightarrow \alpha = 0$$

$\theta_1 = \sin^{-1}(1/N)$

$\theta_2 = \sin^{-1}(2/N)$

$\theta_3 = \sin^{-1}(3/N)$

$\theta_4 = \sin^{-1}(4/5)$

If N= no. of elements=5,

$\theta_5 = \sin^{-1}(5/N) = \Pi/2$

From experimental graph,the main lobe is repeated when $\theta = 90$.
Therefore, $\Psi = \alpha + k*d\sin\theta = kd$ --------------------------------------(4)

Therefore,when n=0;

$\theta_0 = 0$.

Therefore $\Psi = 0$.
But $\Psi = 0$. stands for main lobe . So when deriving for nulls,n =0,N,2N...are not used.

Lets test an array along X axis for B**roadside case**. To check whether the nulls occur at the value of θ calculated.
Here n=5 , d= $\lambda = 100$ m , $\theta = 0$ $\Phi = 0$

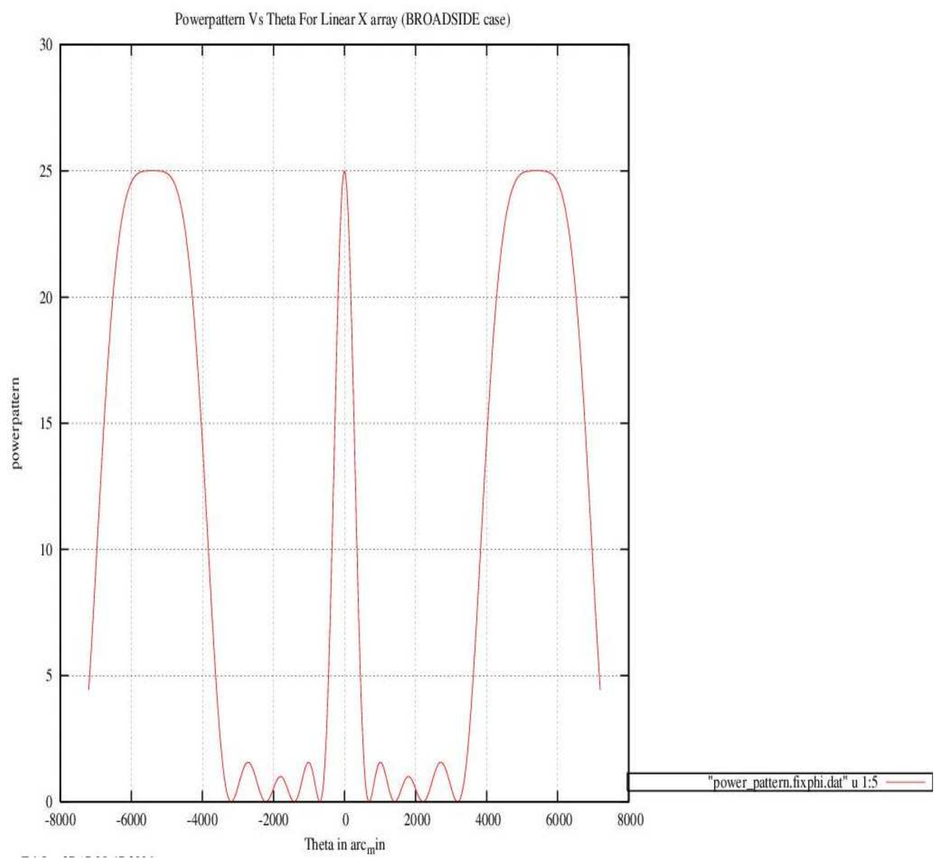


fig1.3

No of elements = N = 5

BWFN	***CALCULATED THETA(deg) MEASURED***	
FIRST	23.1	23.27
SECOND	47.16	47.56
THIRD	73.74	73.42
FOURTH	106.26	106.46

For N = 5 ,

since in between two principal maxima there are 4 nulls (corresponding to n =

1,2,3,4) therefore we have three sidelobes.

So in general for N element array , there are (N-1) nulls and (N-2) sidelobes.

Peaks of array function are found by determining the zeros of the numerator term where the denominator term is simultaneously zero.

i.e , $\sin(N \psi / 2) = 0$ and $\sin(\psi / 2) = 0$.

When $\psi = (+ -) 2m \Pi = \alpha + kd \sin \theta_m$

$\theta_m = \sin^{-1}[\{ \lambda (- \alpha +- 2 m \Pi)\} / 2 \Pi d]$

For Broadside $\alpha = 0$.

Therefore $\theta_m = \sin^{-1}[\lambda m / d]$ m= 0,1,2.......

When m=0 $\theta_m = \sin^{-1}[0] = 0$

For broadside,

$\psi = 0$ so m = 0 corresponds to main lobe.

For m=1 , $\lambda = d$

$\theta_m = \sin^{-1}[1] = \Pi / 2$

For broadside,

$\psi = \alpha + kd \sin \theta = 0 + kd * 1 = kd$ -------------------------(4)

From (3) and (4) we get

When n= 5 it corresponds to same θ and ψ as m = 1

==> Next main lobe occuring at m=1 is also reflected for nulls when we take n= multiples of no of elements of array.

Therefore n $\neq$ 0 , N, 2N,...........

Lets test an array along X axis for **Endfire case**. To check whether the nulls occur at the value of θ calculated.

Here N=5 , d= λ = 100 m , θ = 90deg Φ = 0 deg.

Therefore α = -kd.

Nulls:

$$\theta_n = \sin^{-1} [(-\alpha \pm 2 n \Pi /N) /(kd)]$$

Putting the values we get

$\theta_1 = \sin^{-1} (-4/5)$

$\theta_2 = \sin^{-1} (3/5)$

$\theta_3 = \sin^{-1} (2/5)$

$\theta_4 = \sin^{-1} (1/5)$

$\theta_5 = \sin^{-1} (0/N)$

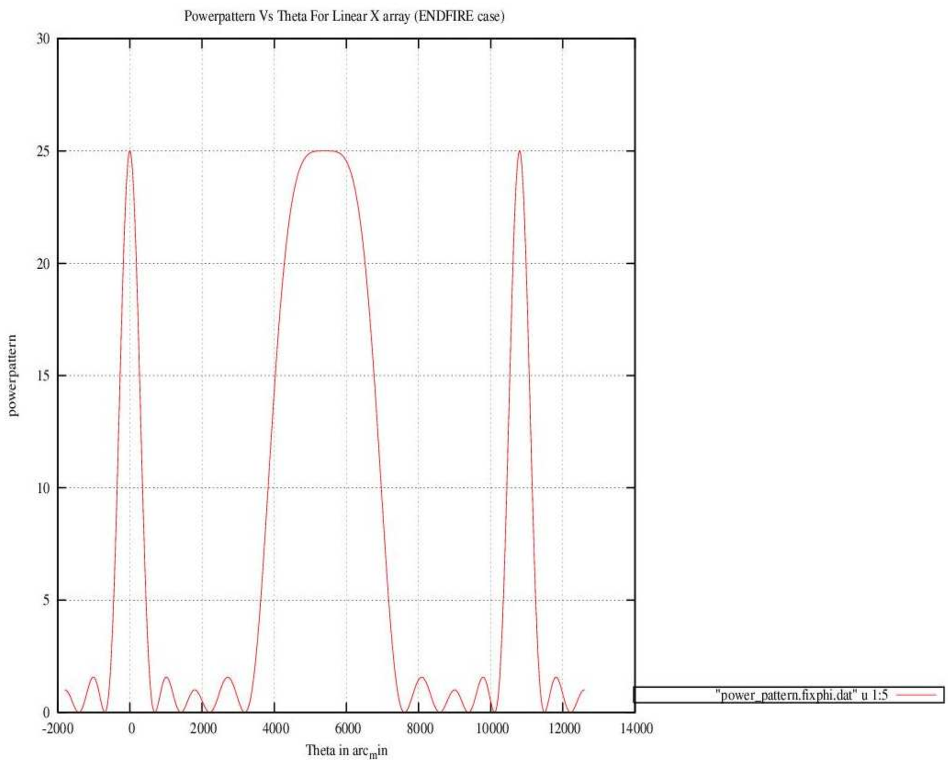


fig 1.4

Comparing the measured and the calculated value of BWFN we tabulate then as below:

No of elements = N = 5

BWFN	***CALCULATED*** ***THETA(deg)***	***MEASURED***
FIRST	74	74
SECOND	106.28	106.39
THIRD	132.86	133.1
FOURTH	156.94	156.75

For Nulls n $\neq$ 0,N,2N,....

n= 1,2,3,4 If N=5

For n=0 ;

$\theta_0 = \mathbf{sin^{-1}\ (-1)} = -90$ deg

Therefore θ_0=(180-90)= 90deg.

Now $\psi = \alpha + kd \sin\theta = -kd + kd \sin 90 = 0$

==> Main lobe.

Between the main lobe and the grating lobe there are four nulls corresponding to n= 1,2,3,4 Therefore there are three side lobes as obtained experimentally.

Peaks:

$$\theta_m = \sin^{-1}[\lambda(-\alpha \pm 2m\Pi)/(2\Pi d)]$$

For m=0

θ_0 **= 90 deg**

Therefore $\psi = 0$

Hence it is the main lobe.

Therefore n = 0 (from null equation) and m = 0 (from peak equation) represent the same main lobe.

Lets test an array along X=Y line for broadside case.
Here θ =0, Φ = 0.

The formula for BWFN = 2 $\sin^{-1}$ [(- α $\pm$ 2 n Π /N) /(kd)]
Here α =0.
gives, = 16.26 deg (calculated)

But when it is measured from graph we get
BWFN = 23.27 deg (measured)
As depicted in graph()

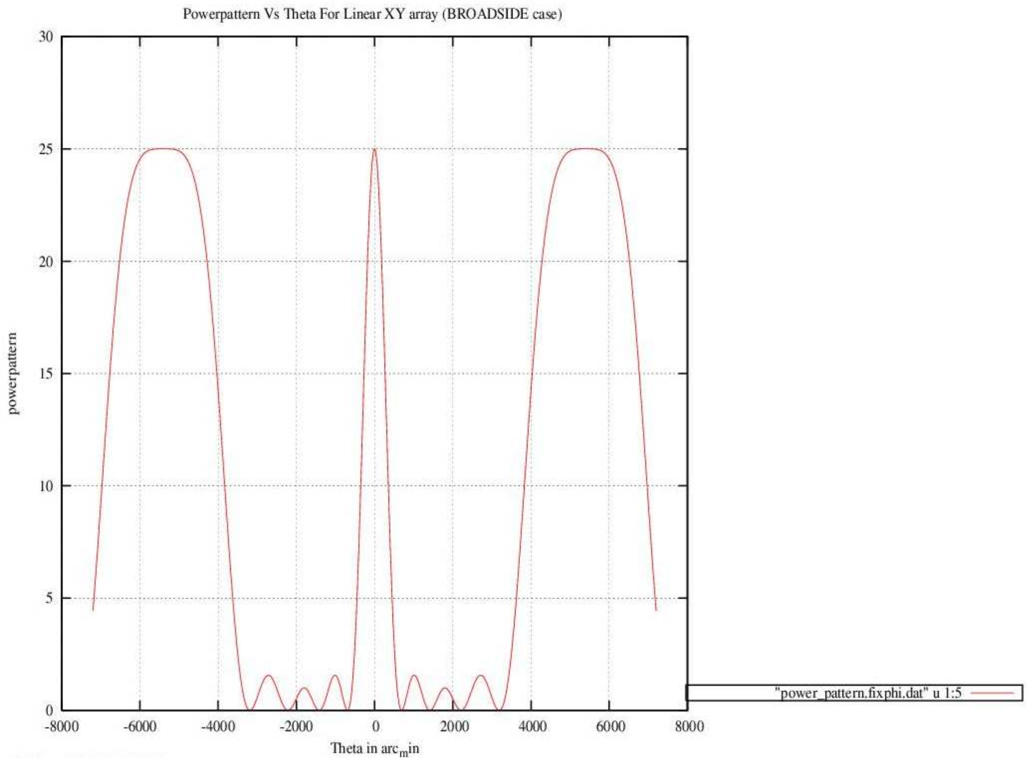


fig1.5

Hence there is a discrepancy of ratio
= (23.27 deg (measured) / 16.26 deg (calculated))
=1.4
This discrepancy of factor (sqrt2) is due to the fact that in deriving the formula for NULLS we assumed that there is no component due to azimuthal angle. In deriving the array factor we had inherently assumed that the plane where the field contribution due to the array points is to be found, must pass through the line containing the array points.

So to align the array points in the proper broadside for X-Y array,we must take

$$\theta = 0, \quad \Phi = 45\text{deg}.$$

Here BWFN(calculated)= 16.26 deg and (measured) = 16.37 deg.
As depicted in graph()

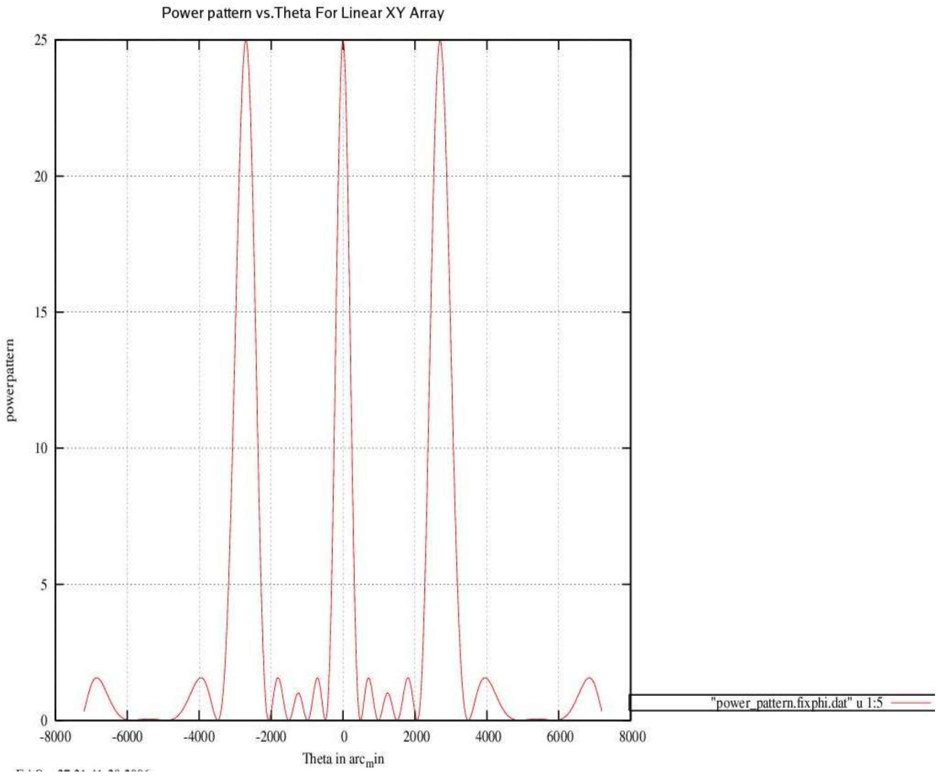


fig 1.6

If the number of array points contributing to the array factor is increased, the number of side lobes increases . Also if the array factor is not normalised then we note that the peak amplitude of the main lobe increases . It is proportional to (number of elements) ^{2} . This is as shown in the graph( )

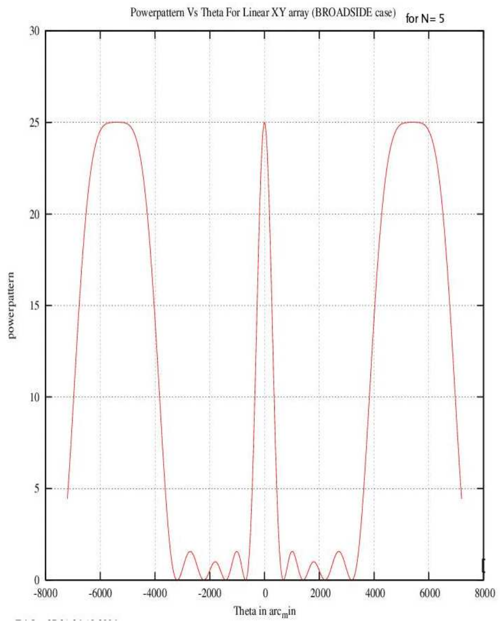


fig1.7

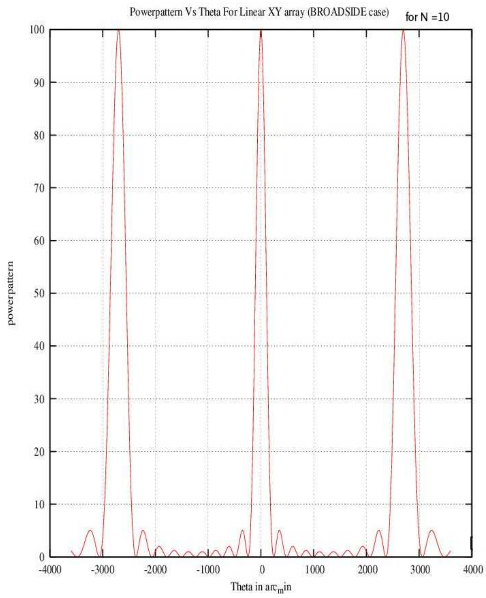


fig1.8

Observation:

For an array along X=Y line,

When Φ = -45deg i.e 135deg , Whatever be the value of θ ,the power level remains constant. This is as depicted in the graph ( )below

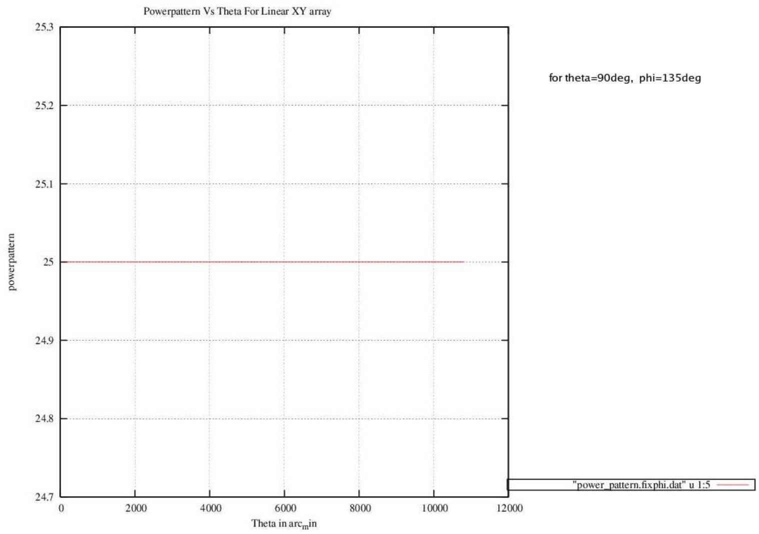


fig1.9

Broadside array produces disc shaped beams covering full 360deg in plane normal to axis of array- i.e cylindrical symmetry.

Endfire array produces cigar shaped beam which has same shape in all planes containing axis of array i.e planar symmetry.

CHAPTER 2

PHASING OF BEAMS OF A LINEAR ARRAY:

The program is then modified to incorporate two attributes.

1) Direction for phasing the array beam. (θ_p, ϕ_p)
2) Direction for calculation of the power pattern . (θ_0, ϕ_0)

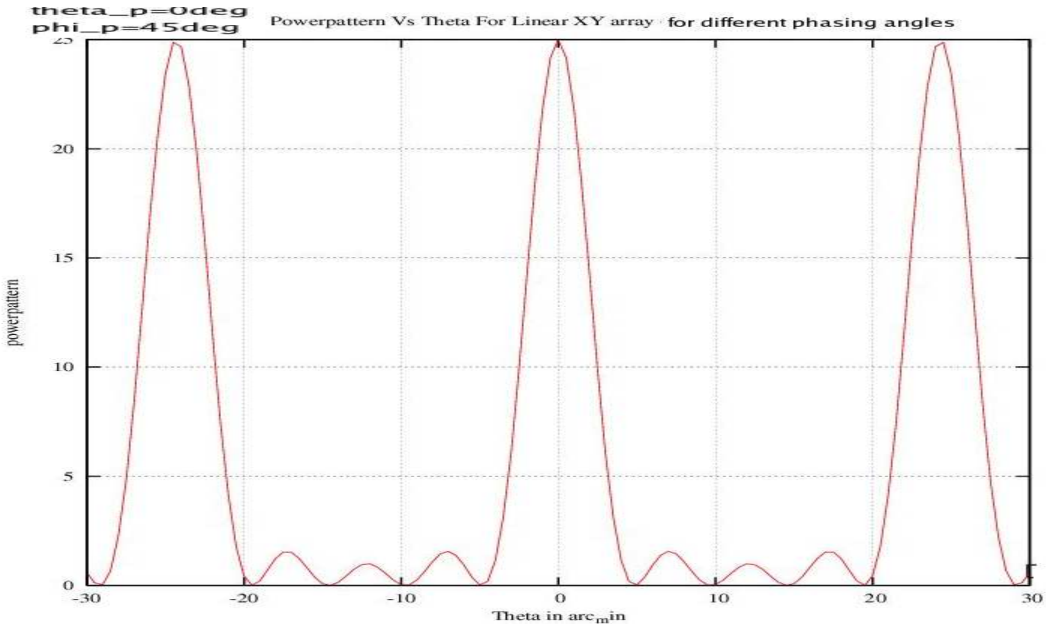
   This change is incorporated for practical application of the array pattern. It is because , it might be necessary to phase the beam in a required direction ,while the beam pattern may be necessary to be calculated in different direction for an antenna.

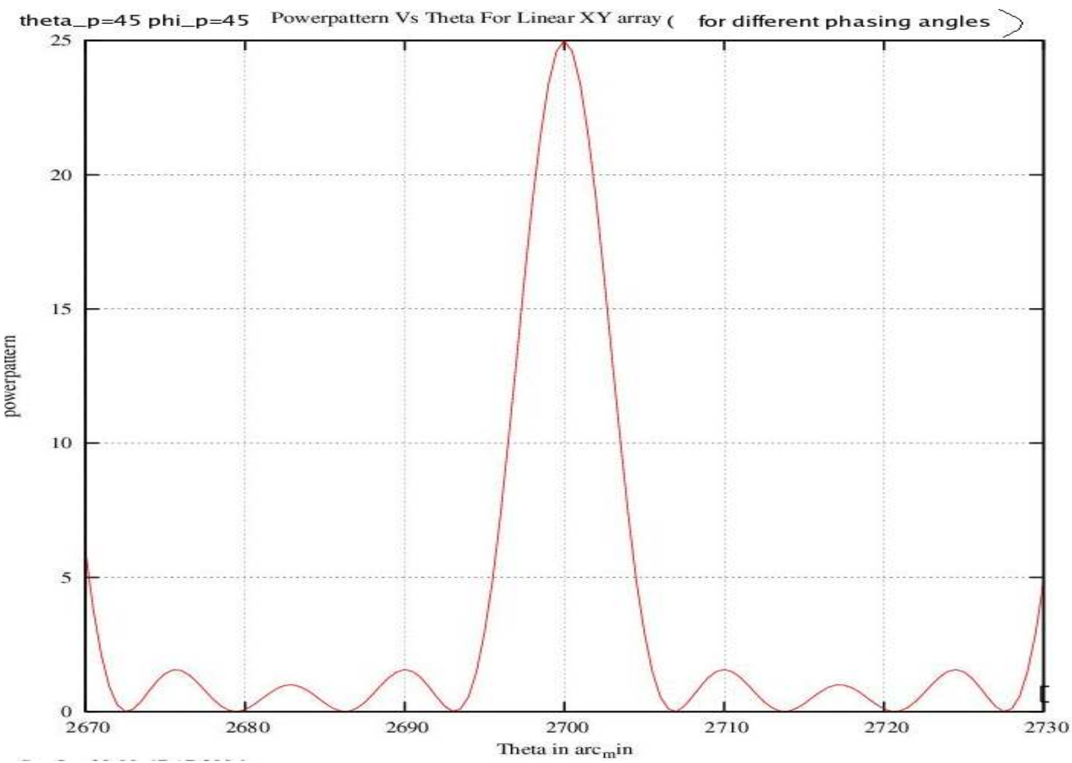
This is experimented for X-Y array with 5 elements with different θ_p. values,and the power pattern being calculated about that angle.

Here λ =1 m , N = 5,

From below data we note that with λ =1m,BWFN of the order of arcmin is obtained

θ_p (in deg)	ϕ_p (in deg)	θ_0 (in deg)	ϕ_0 (in deg)	Range of θ (in deg)	Range of ϕ (in deg)	Accuracy of θ (in arcmin)	Accuracy of ϕ (in arcmin)	***BWFN (measured)(in arcmin)***	***BWFN (calculated)(in arcmin)***
0	45	0	45	0.5	1	0.5	30	9.98	9.72
45	45	45	45	0.5	1	0.5	30	13	14.4
90	45	90	45	7	1	0.5	30	366	366


theta_p=0deg
phi_p=45deg
Powerpattern Vs Theta For Linear XY array for different phasing angles
powerpattern
Theta in arc_min
fig2.1


theta_p=45 phi_p=45
Powerpattern Vs Theta For Linear XY array (for different phasing angles)
powerpattern
Theta in arc_min
fig2.2

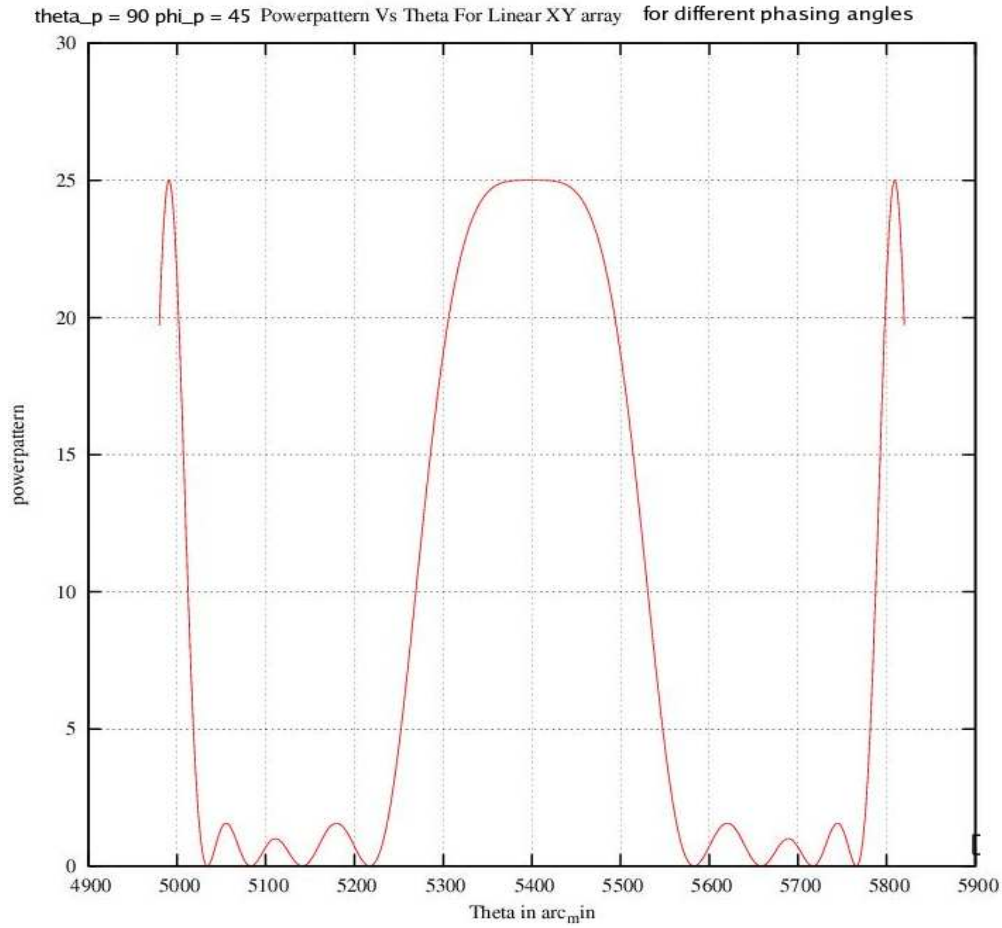
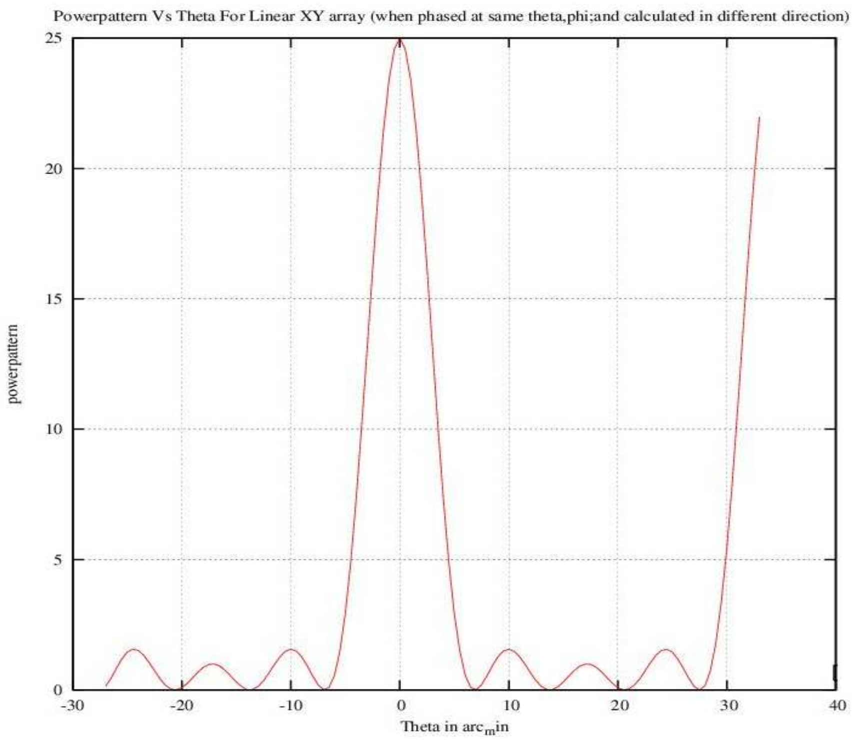


Fig 2.3

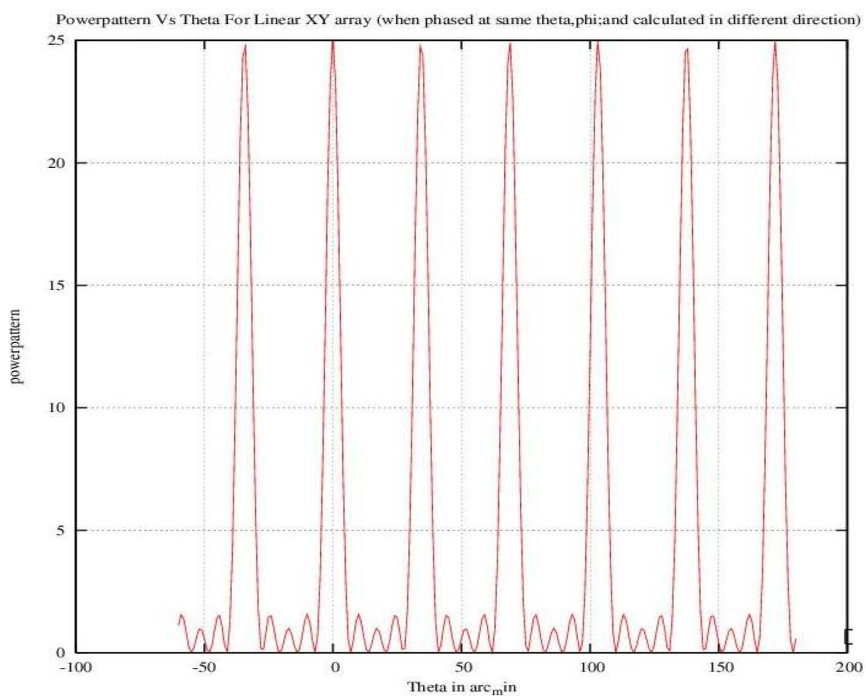
Next we keep $\theta_p = 0$, $\phi_p = 0$ fixed,and observe the nature of the power pattern by varying θ_0, i.e the point of calculation of the power pattern is changed keeping the phasing of the beam at same theta_p



θ_{0}, =3arc min

ϕ_{0} = 0deg

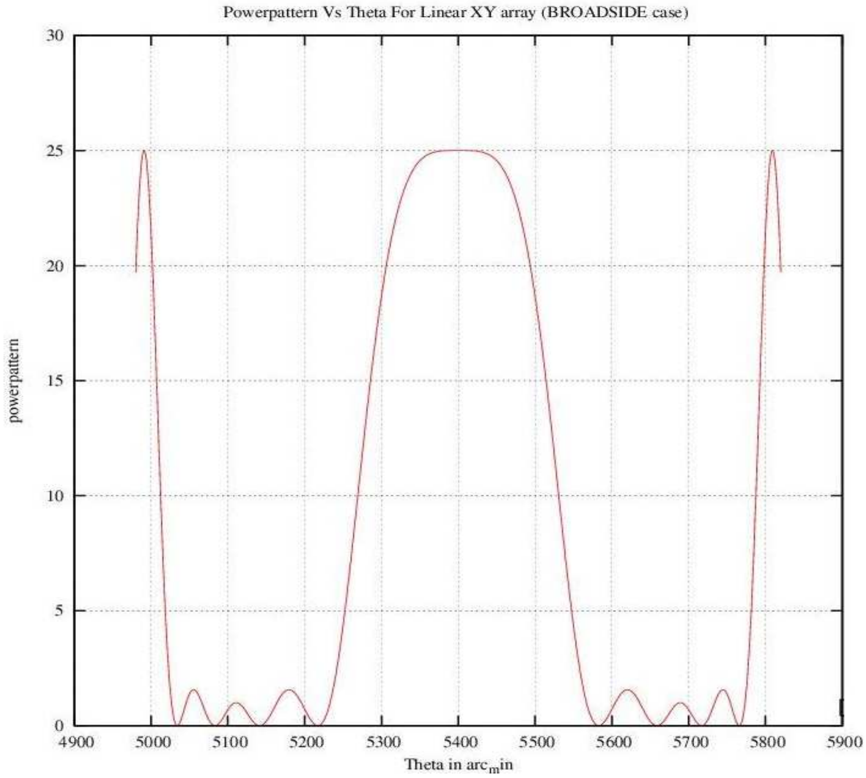
fig2.3



θ_{0} = 1 deg

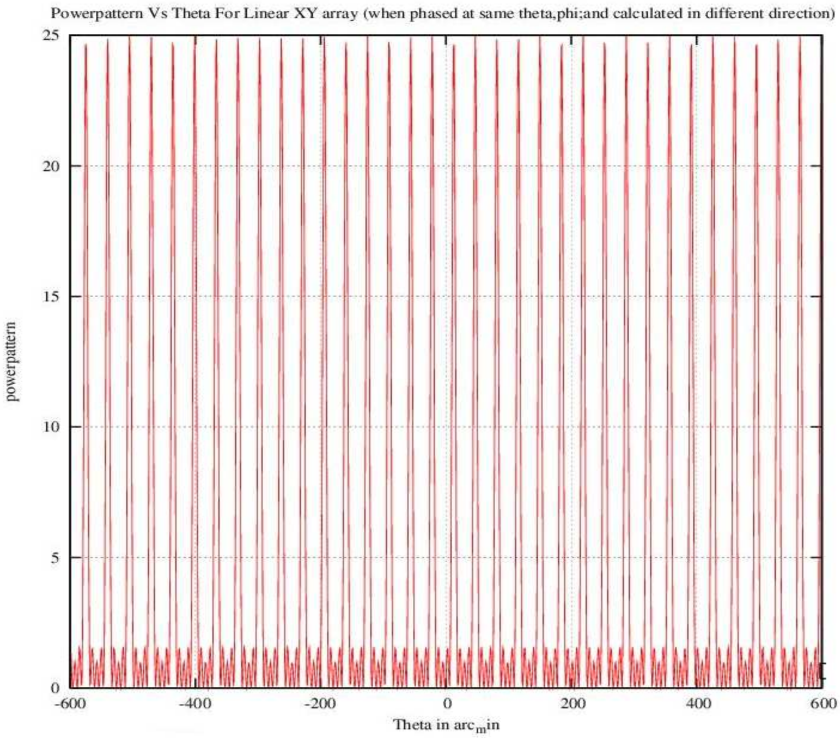
ϕ_{0} = 0deg

fig2.4



$\theta_0 = 5\text{deg}$
$\phi_0 = 0\text{deg}$

fig2.5



θ_0 = 5 deg

$_0$ =0 deg

fig2.6

Now we make the following observation:

We keep θ_0 = 0deg , ϕ_0 = 0deg , N=5 , λ =70m

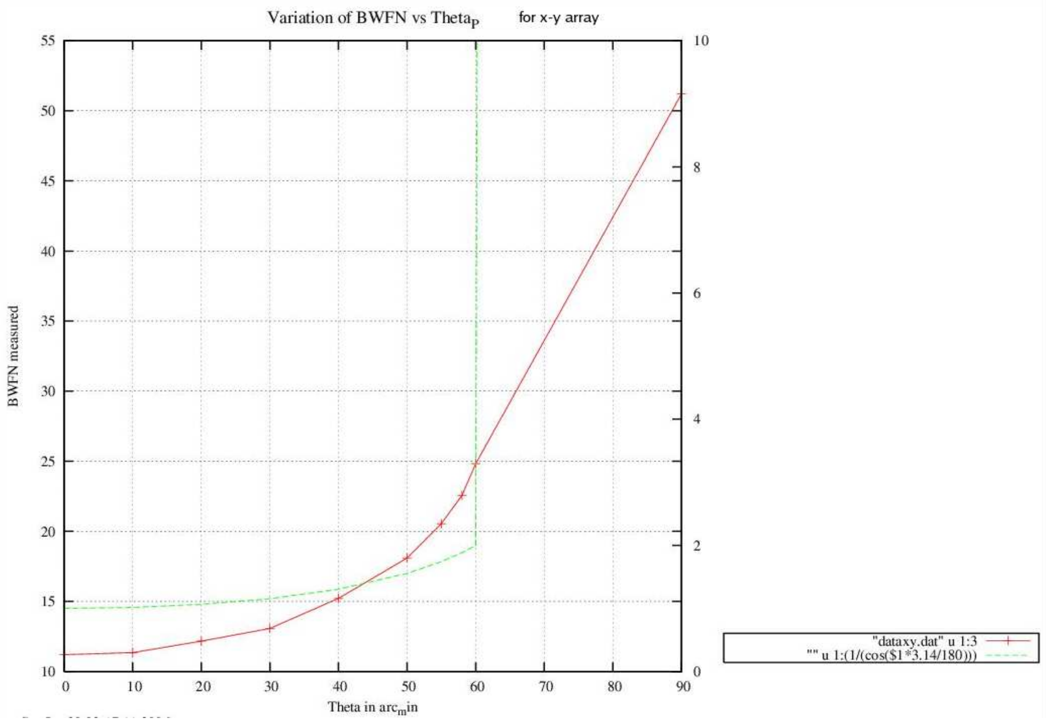
Changing the phasing angle θ_p , with φ_p = 0 kept constant ,we measure the BWFN from the data file . Also the calculated value is noted . A plot of variation BWFN vs theta_p is then made. Array along X axis and X=Y line is considered.

FOR ARRAY ALONG X=Y LINE

θ_p *(in deg)*	ϕ_p (in deg)	***BWFN(in deg)***		***Main lobe position (in deg)***
		measured	***calculated***	
0	45	11.2	11.36	0
10	45	11.35	11.44	10
20	45	12.17	11.86	20.1
30	45	13.07	12.72	30.11
40	45	15.22	14.12	40.15
50	45	18.1	16.32	50
55	45	20.55	17.86	55.3
58	45	22.58	18.98	58
60	45	24.83	19.82	59.9
90	45	51.22	51.42	90

From above data table we note that as the phasing angle drifts more and more away from the value of θ_0 ,the discrepancy in measured and calculated value of BWFN increases . This is because as θ_p increases, the power pattern gets folded about theta =90deg.This is shown in the plots below .Then at e θ_p ,=90deg the pattern again peaks .

fig 2.7



FOR ARRAY ALONG X AXIS

θ_p *(in deg)*	ϕ_p (in deg)	***BWFN(in deg)***		***Main lobe position (in deg)***
		measured	***calculated***	
0	0	15.97	16.08	0
10	0	16.2	16.14	10
20	0	16.97	16.68	29.98
30	0	21.1	19.64	30
50	0	24.73	22.48	49.91
55	0	30.18	24.46	55
58	0	34.48	25.84	57.91
60	0	43.97	26.9	59.9
90	0	60.97	61.38	90

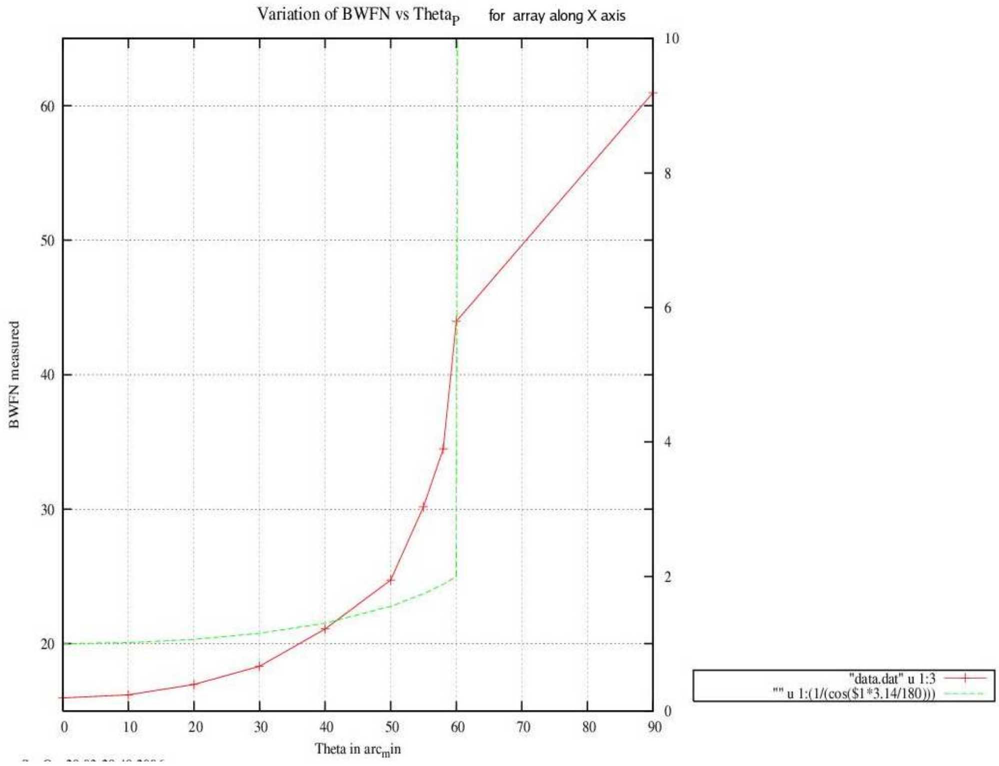

Variation of BWFN vs Theta_p for array along X axis
BWFN measured
Theta in arc_min
"data.dat" u 1:3
"" u 1:(1/(cos($1*3.14/180)))

fig2.8

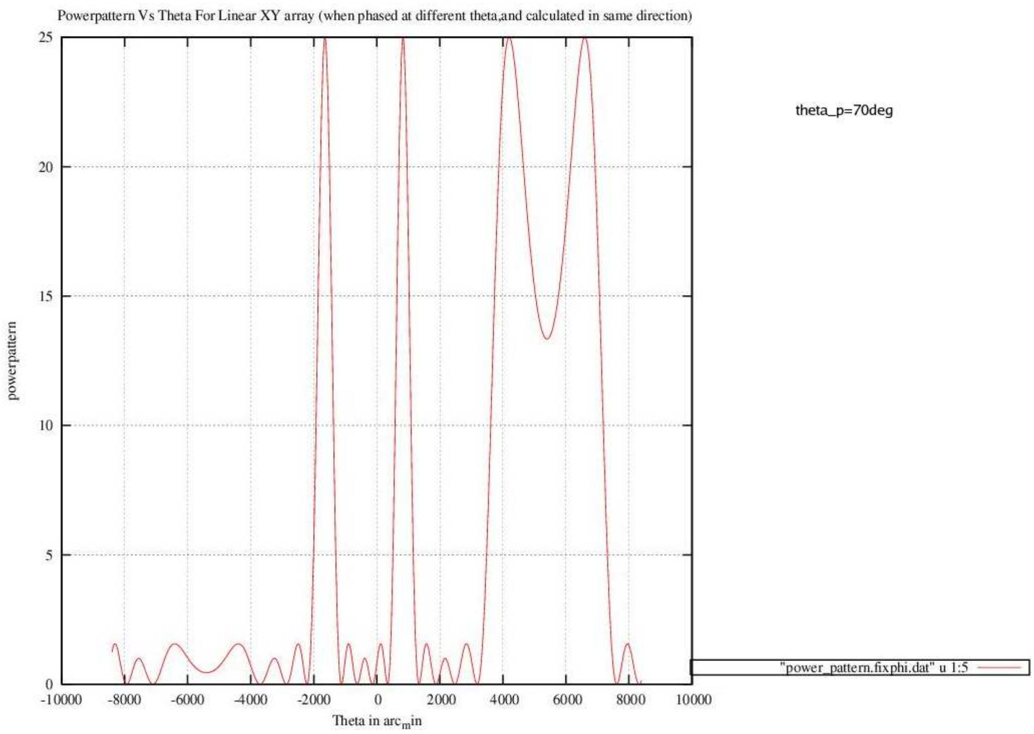
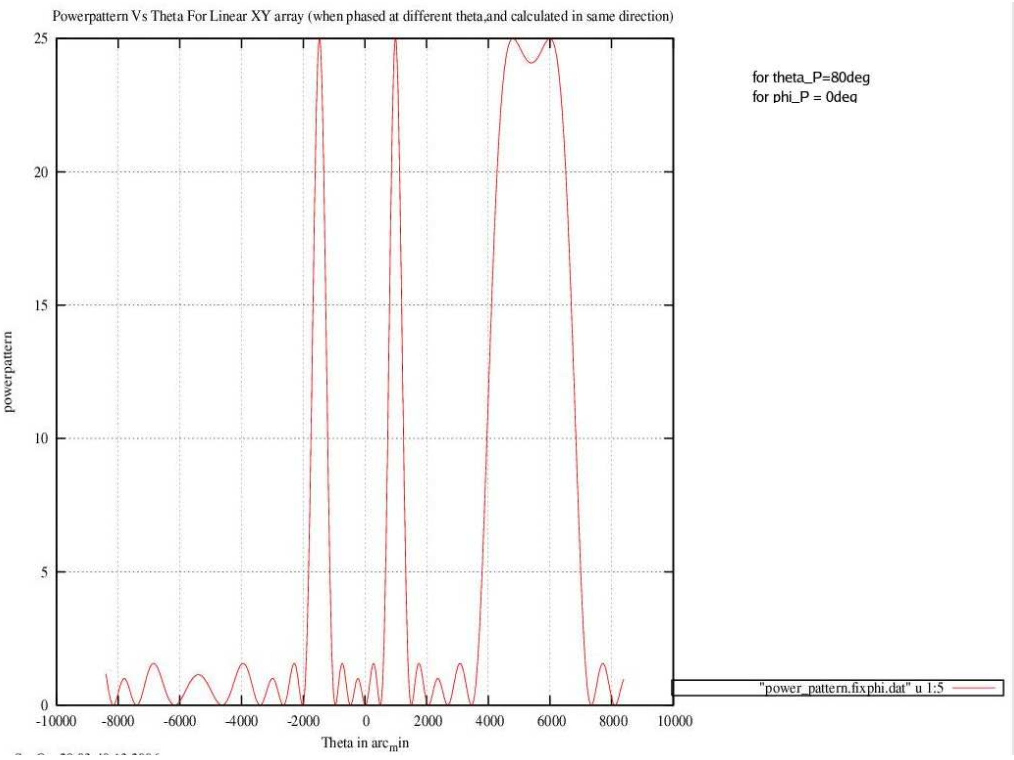

Powerpattern Vs Theta For Linear XY array (when phased at different theta,and calculated in same direction)
theta_p=70deg
powerpattern
Theta in arc_min
"power_pattern.fixphi.dat" u 1:5

fig2.9



Powerpattern Vs Theta For Linear XY array (when phased at different theta,and calculated in same direction)
for theta_P=80deg
for phi_P = 0deg
powerpattern
Theta in arc_min
"power_pattern.fixphi.dat" u 1:5

fig 2.10

CHAPTER 3

OBSERVATIONS ON GMRT ARRAY(CENTRAL SQUARE)

The Giant Meter Wave Radio Telescope(GMRT) consists of an array of 30 antennas,each antenna being 45m in diameter. The GMRT has a hybrid configuration with 14 of the antenna randomly distributed in a central region (approx 1 km across), called central square. The distribution of the antenna in the central square is "randomized" . The remaining antenna is distributed in a roughly Y shaped configuration with the length of each arm of the Y being approx 14km. The maximum baseline length between extreme arm antenna is approx 25km.

The central square antenna provide a large number of relatively short baselines. This is very useful for imaging large extended sources. The arm antennas on other hand are useful in imaging small sources where high angular resolution is essential.

We now study the power pattern vs angle θ for the GMRT array. We consider the 14 antenna of the central square with C2 as reference antenna.
That the central square provides no grating lobes is confirmed on observing the power pattern over a wide range of θ .

The pole star is at 19deg elevation at GMRT . Therefore the zenith is 71deg from the polestar. In that case the maximum projected baseline is d (which is equal to 1km). If we phase the beam at an angle θ_p ,the projected baseline becomes d*(90- θ_p -19)deg.
The Half Power Beam Width (HPBW) is calculated using

[λ / (maximum projected baseline)

where λ =wavelength of observation

The 1st table shows the HPBW (measured and calculated) for different phasing angles of the GMRT array.

WITH C2 AS REFENCE ANTENNA.

θ_p (in deg)	ϕ_p (in deg)	θ_0 (in deg)	ϕ_0 (in deg)	***Range of*** θ *(in deg)*	***Accuracy of*** θ *(in arc min)*	***HPBW (measured)(in arc min)***	***HPBW (calculated)(in arc min)***	***HPBW (cal)/ HPBW (meas)***
10	0	10	0	0.3	0.1	5.4	7	1.3
30	0	30	0	0.3	0.1	3.43	4.5	1.33
45	0	45	0	0.3	0.1	2.82	3.8	1.35
71	0	71	0	0.3	0.1	2.61	3.43	1.3

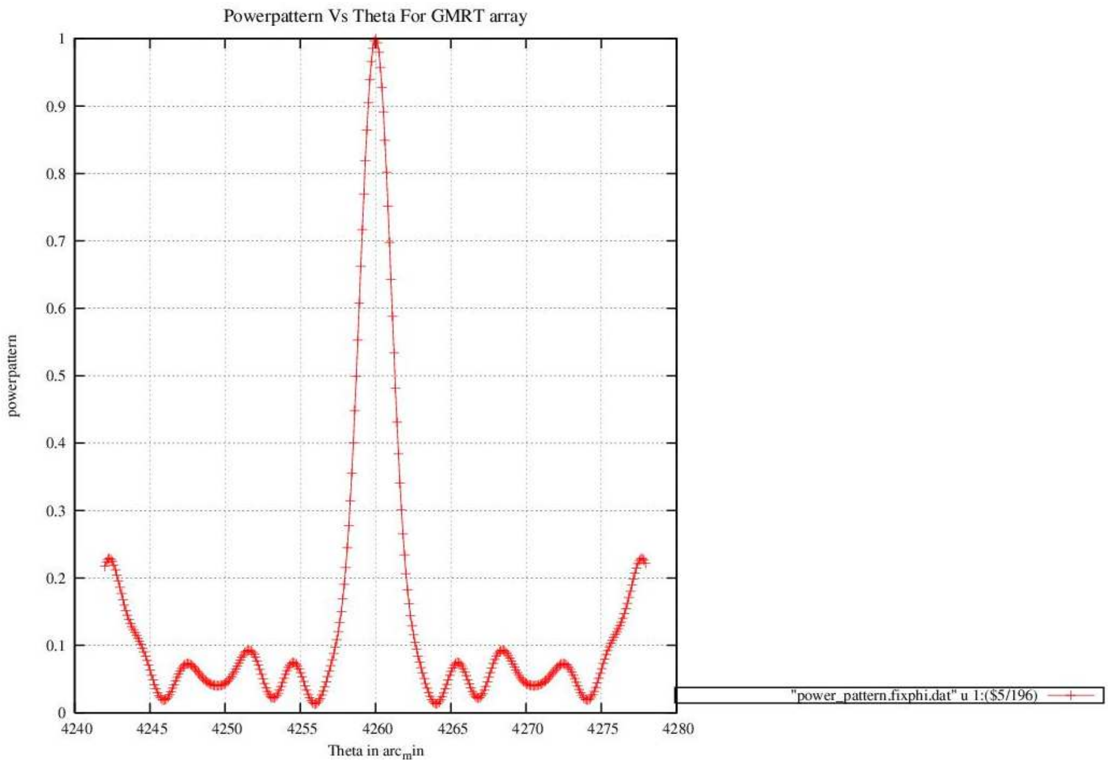


fig 3.1

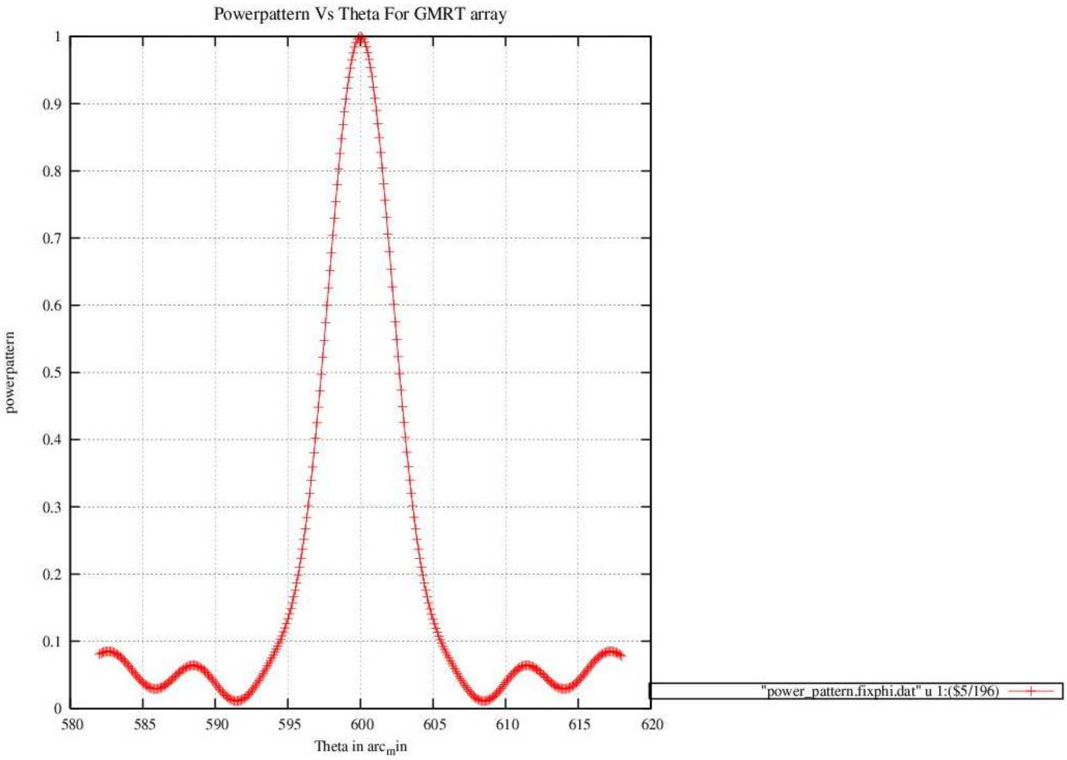

Powerpattern Vs Theta For GMRT array
powerpattern
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
580
585
590
595
600
605
610
615
620
Theta in arc$_m$in
"power_pattern.fixphi.dat" u 1:($5/196)

fig 3.2

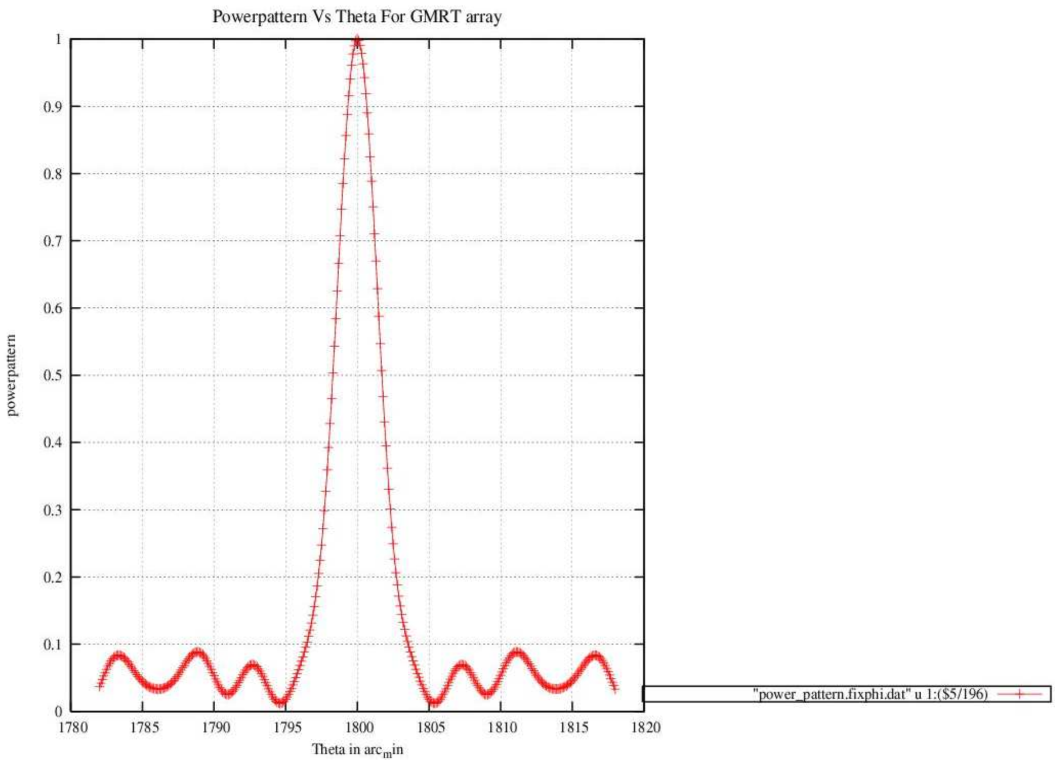
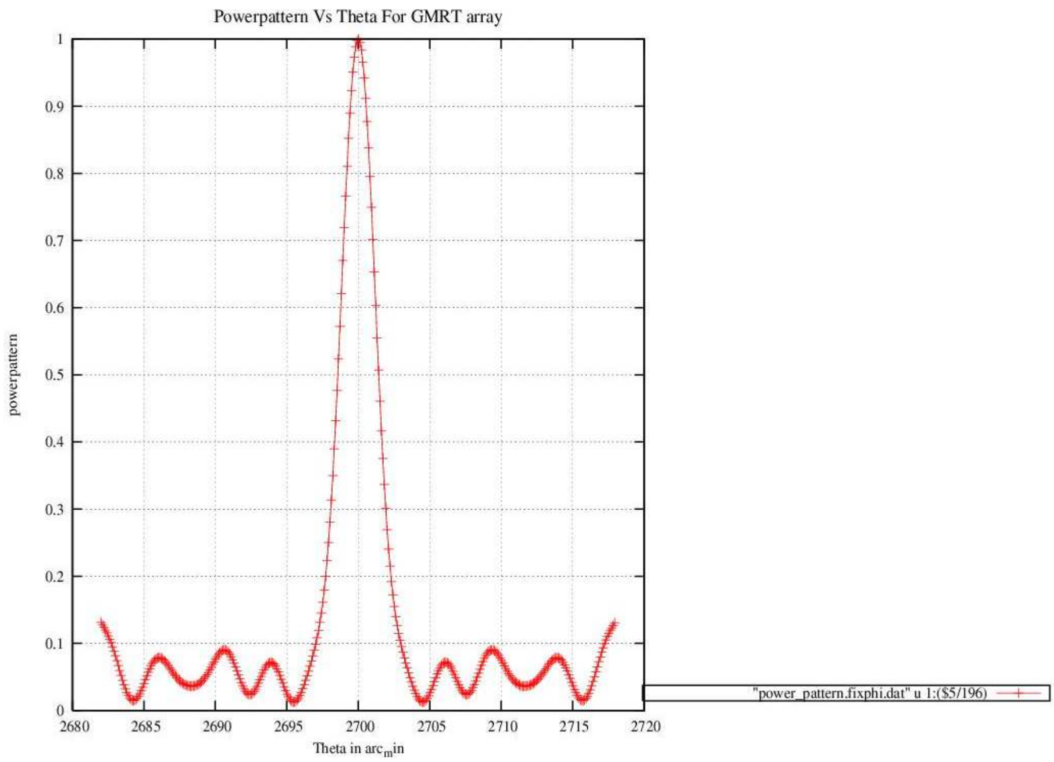

Powerpattern Vs Theta For GMRT array
powerpattern
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1780
1785
1790
1795
1800
1805
1810
1815
1820
Theta in arc$_m$in
"power_pattern.fixphi.dat" u 1:($5/196)

fig3.3



Powerpattern Vs Theta For GMRT array
powerpattern
Theta in arc_min
"power_pattern.fixphi.dat" u 1:($5/196)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
2680
2685
2690
2695
2700
2705
2710
2715
2720

fig 3.4

WITH C13 AS REFERENCE ANTENNA

θ_p (in deg)	ϕ_p (in deg)	θ_0 (in deg)	ϕ_0 (in deg)	***Range of*** θ *(in deg)*	***Accuracy of*** θ *(in arc min)*	***HPBW (measured)(in arc min)***	***HPBW (calculated)(in arc min)***	***HPBW (cal)/ HPBW (meas)***
71	0	71	0	0.3	0.1	2.61	3.43	1.3

Thus we observe that on changing the reference antenna the discrepancy in the value HPBW measured and calculated still remains.

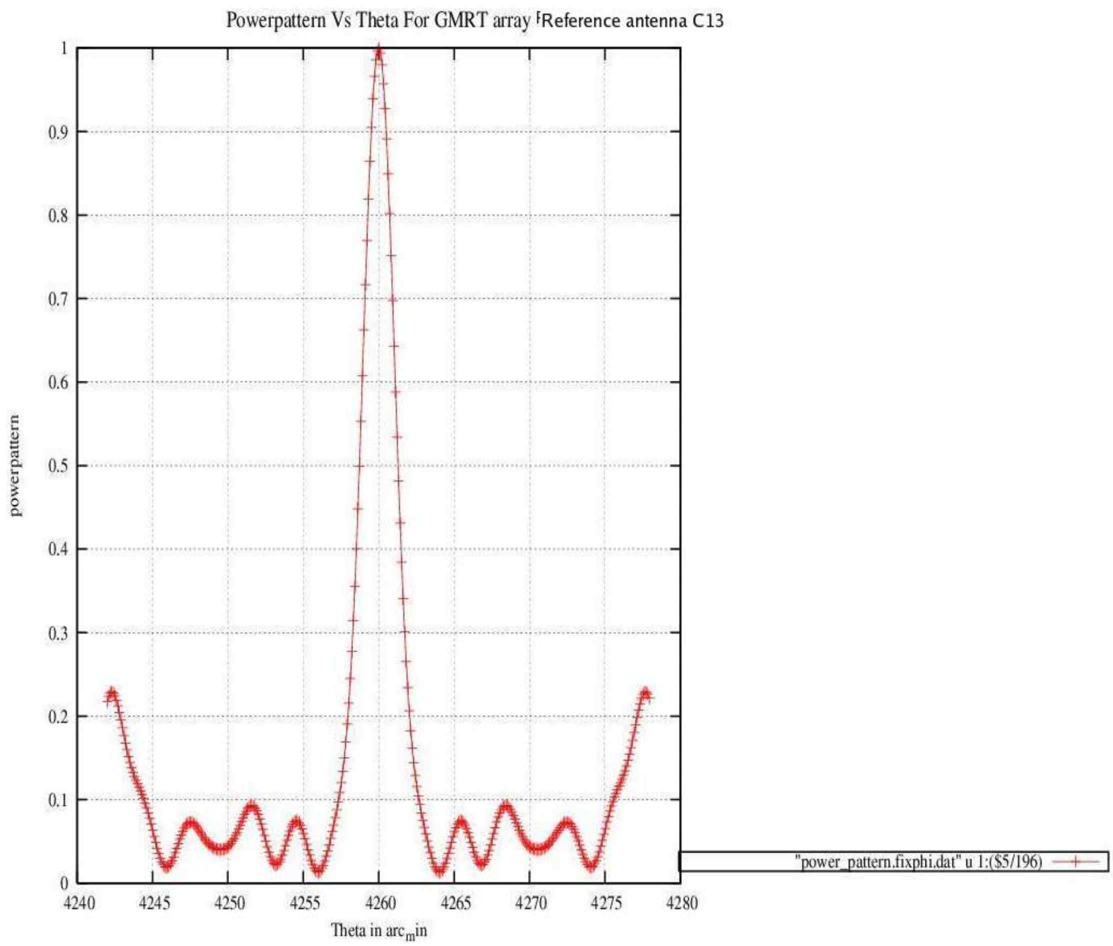


fig 3.5

Now we study the variation in the power pattern of the GMRT array with theta and phi variations,by varying the phasing angles. The pattern is calculated about the phasing angles .The range of theta and phi is kept constant throughout .The plots are taken for fixed phi (i.e variation in theta) and fixed theta (i.e variation in phi).

Range of theta = 5deg   Range of phi = 5 deg
Accuracy of theta = 1'   accuracy of phi = 1'

1.

θ_p=71deg ϕ_p=0 deg

θ_0 = 71 deg ϕ_0 =0 deg

fixed theta=71 deg   fixed phi=0 deg

a.

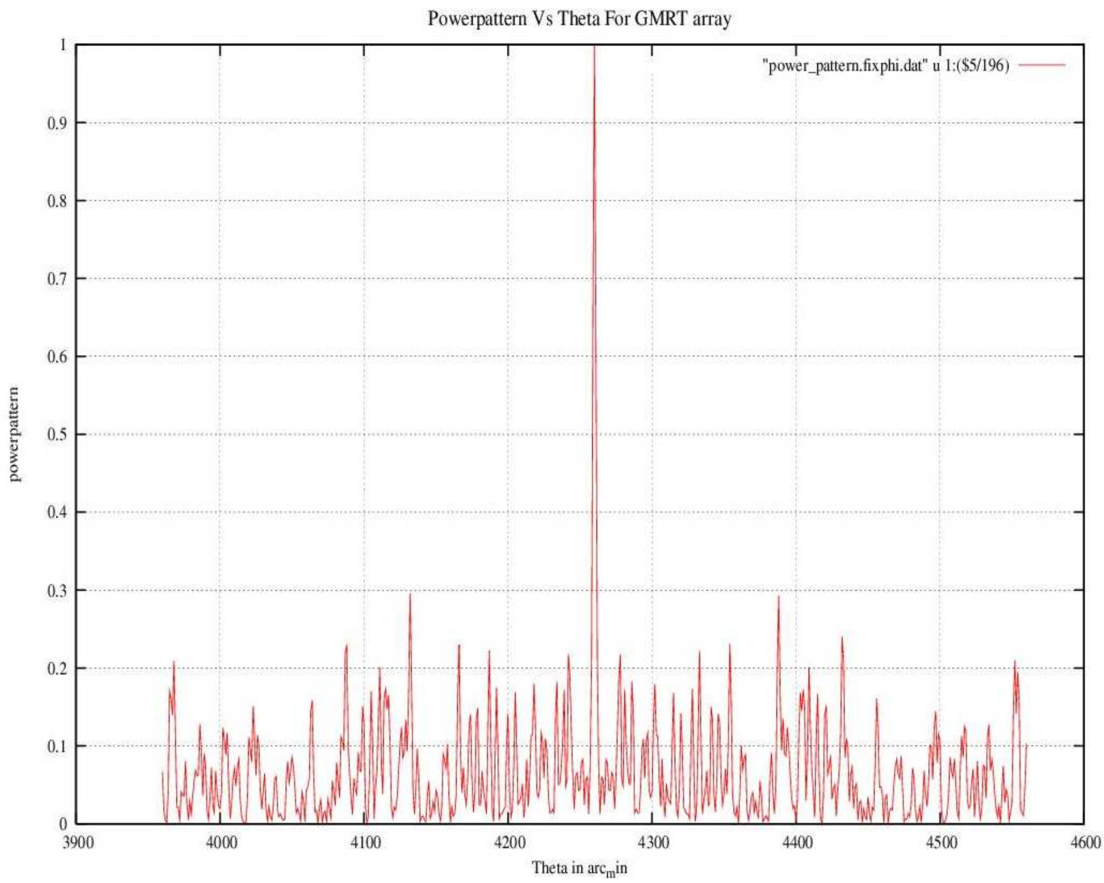


fig3.6

From above plot: side lobes with ~30%power occur 150 ' away from θ_p

b.

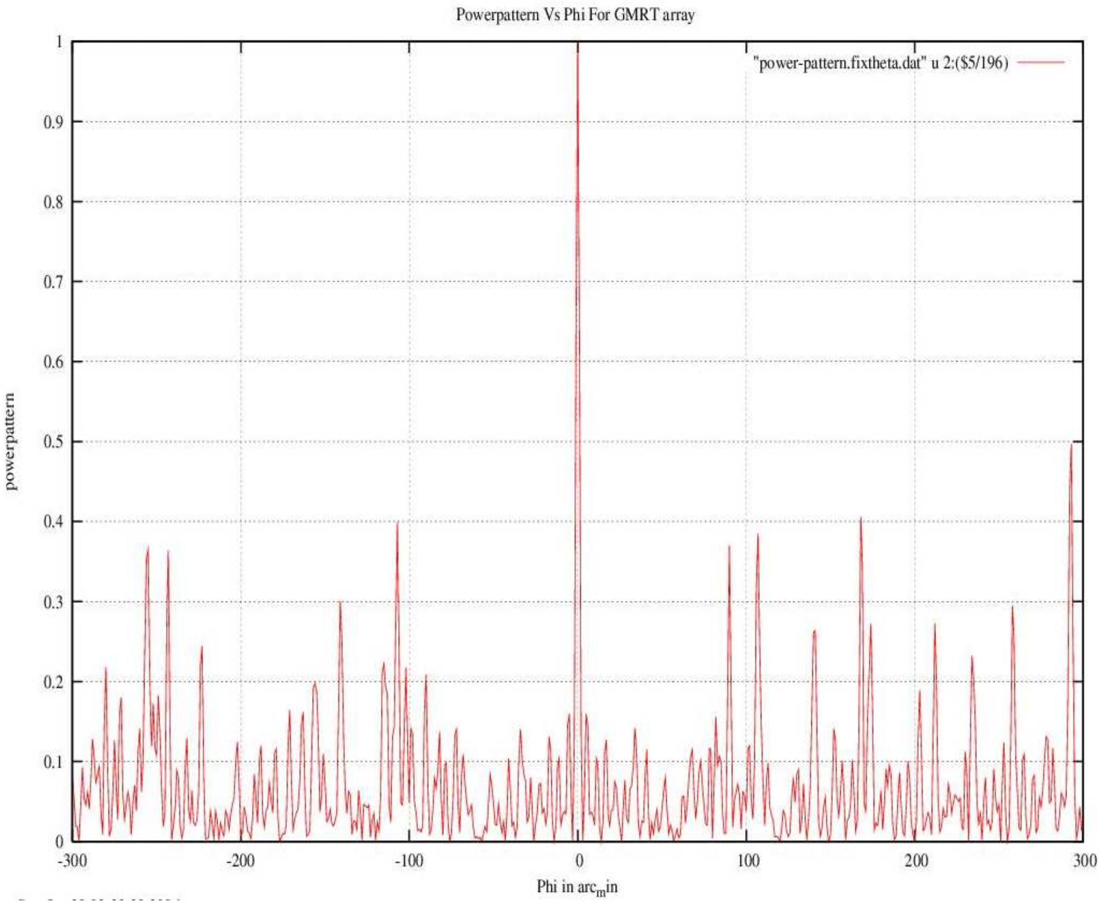


fig 3.7

From above plot:side lobes with ~40%power occur 100' away from ϕ_p

2. $_p$ =71deg $_p$=45deg

$_0$= 71 deg $_0$ =45eg

fixed theta=71 deg fixed phi=45 deg

a.

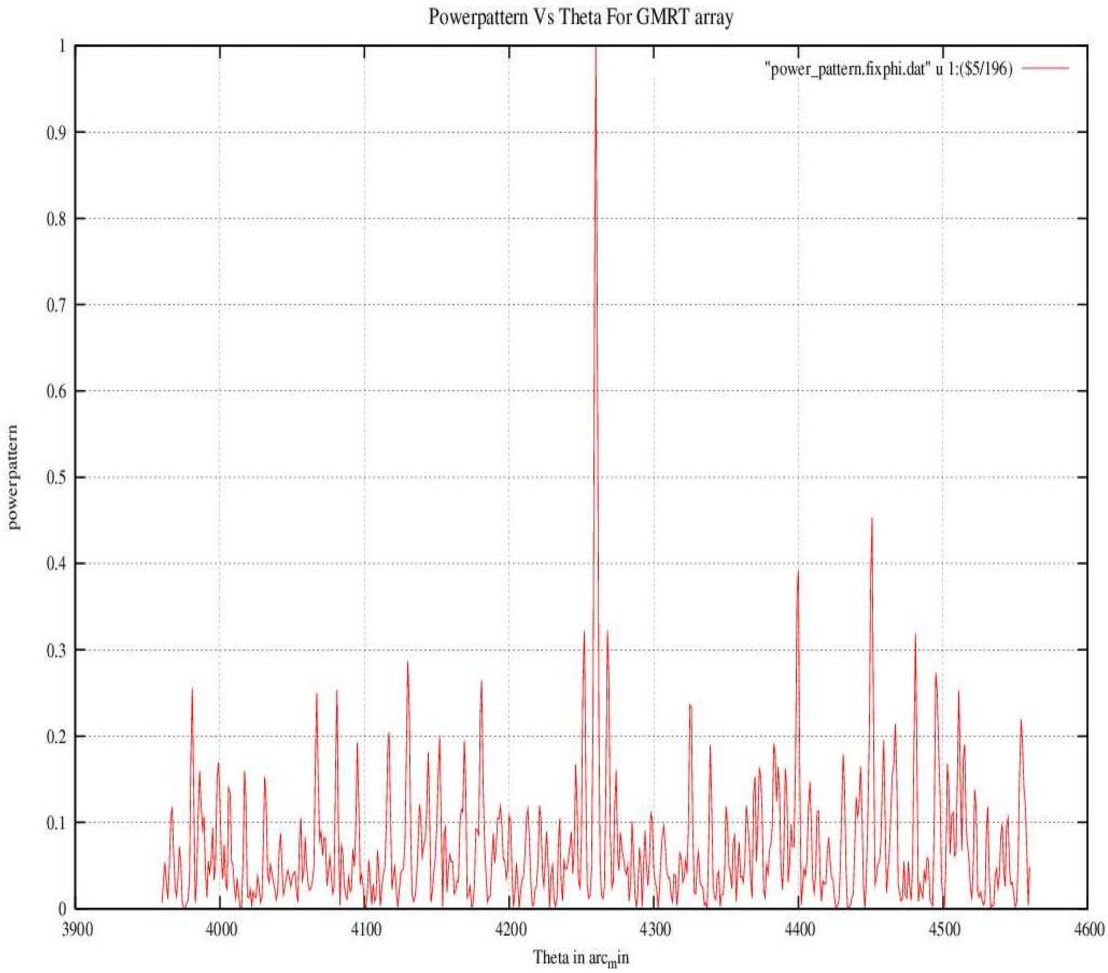


fig 3.8

From above plot:side lobes with ~33% power occur ~10 ' away from θ_p .Side lobe with 40% power occur ~200' away from θ_p

b.

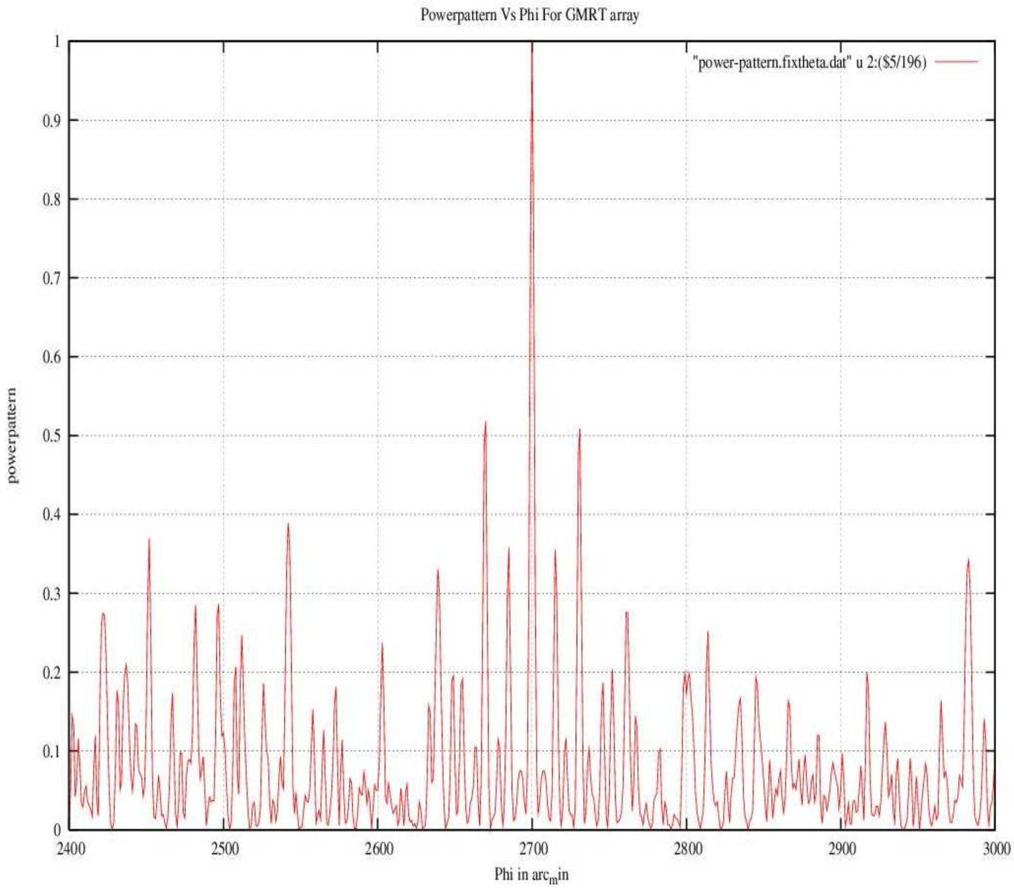


fig 3.9

From above plot: Side lobe with ~52% power occur ~30 ' away from ϕ_p

3. θ_p = 0 deg ϕ_p = 0 deg

θ_0 = 0 deg ϕ_0 =0 deg

fixed theta=0deg fixed phi=0 deg

a.

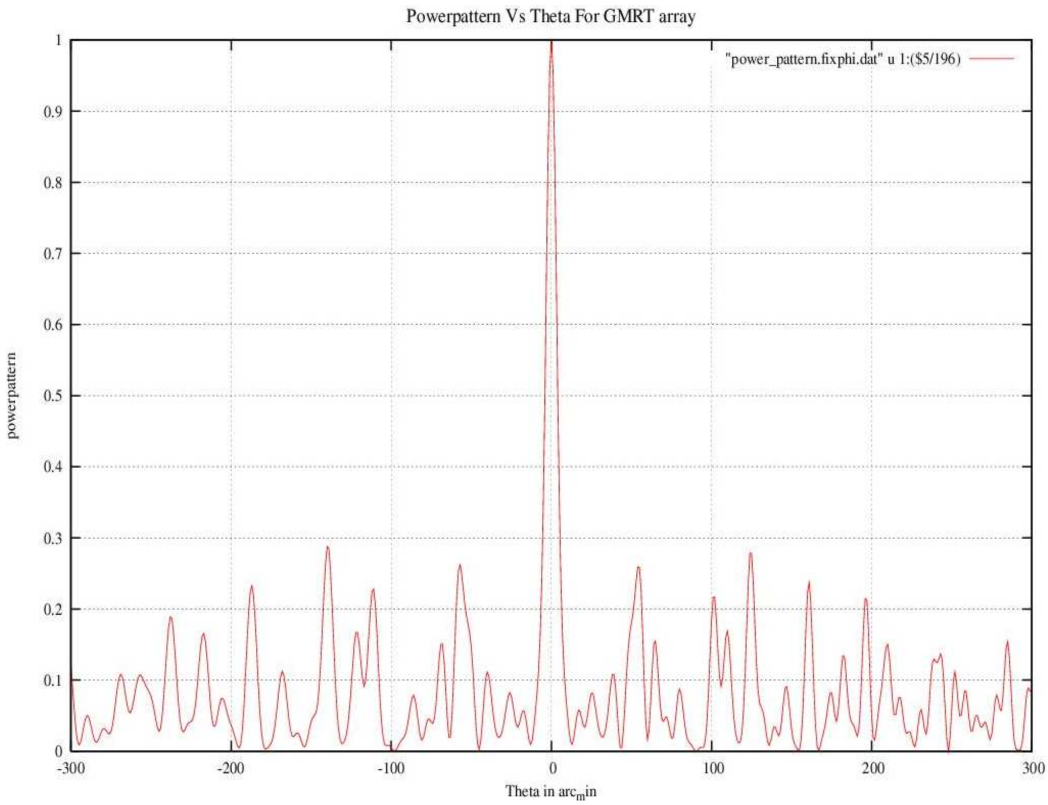


fig 3.10

From above plot:side lobes with 25% power occur 50 ' away from θ_p

b.

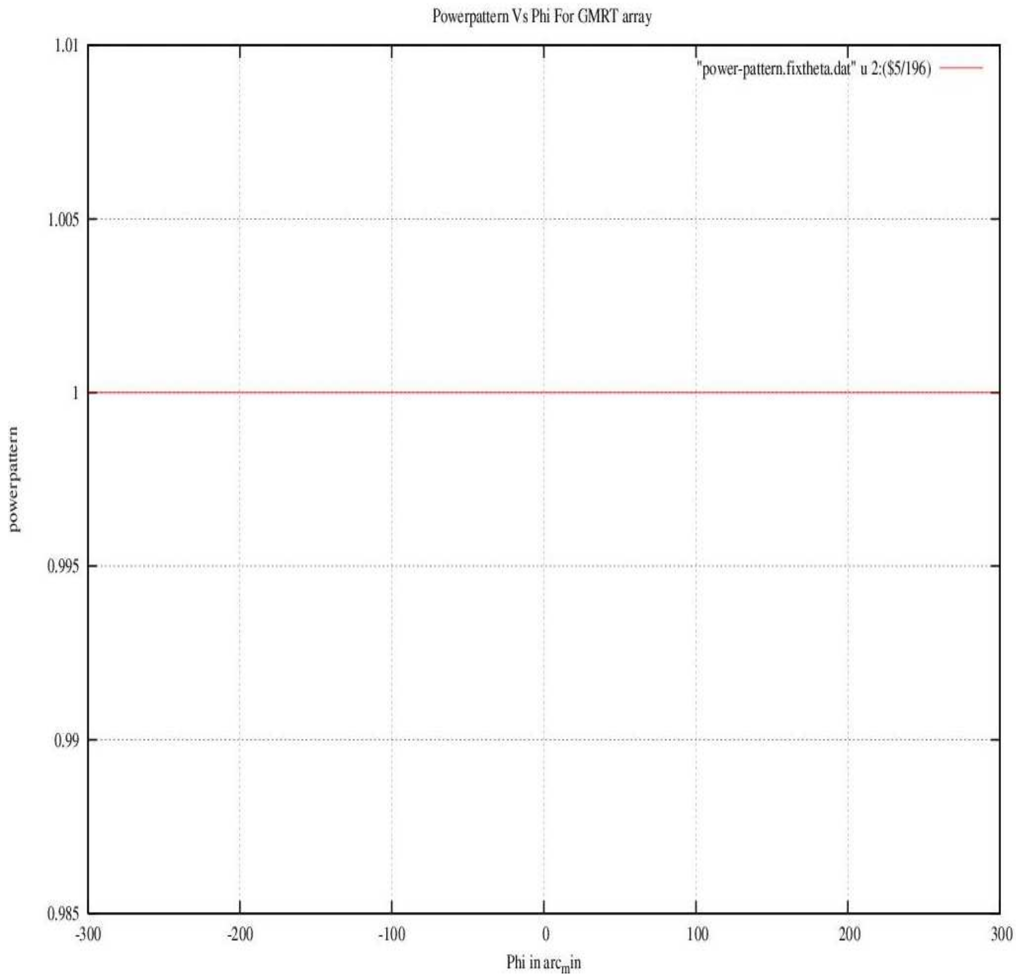


fig 3.11

From above plot: Mathematically When θ_p = 0 deg the contribution of ϕ_p to the power equation is nullified,(as seen in the formula). Also,physically when θ_p =0,on varying phi we obtain a power pattern that is cylindrically symmetric.

4. θ_p = 0deg ϕ_p = 45deg

θ_0 = 0 deg ϕ_0 = 45deg

fixed theta = 0 deg fixed phi= 45deg

a.

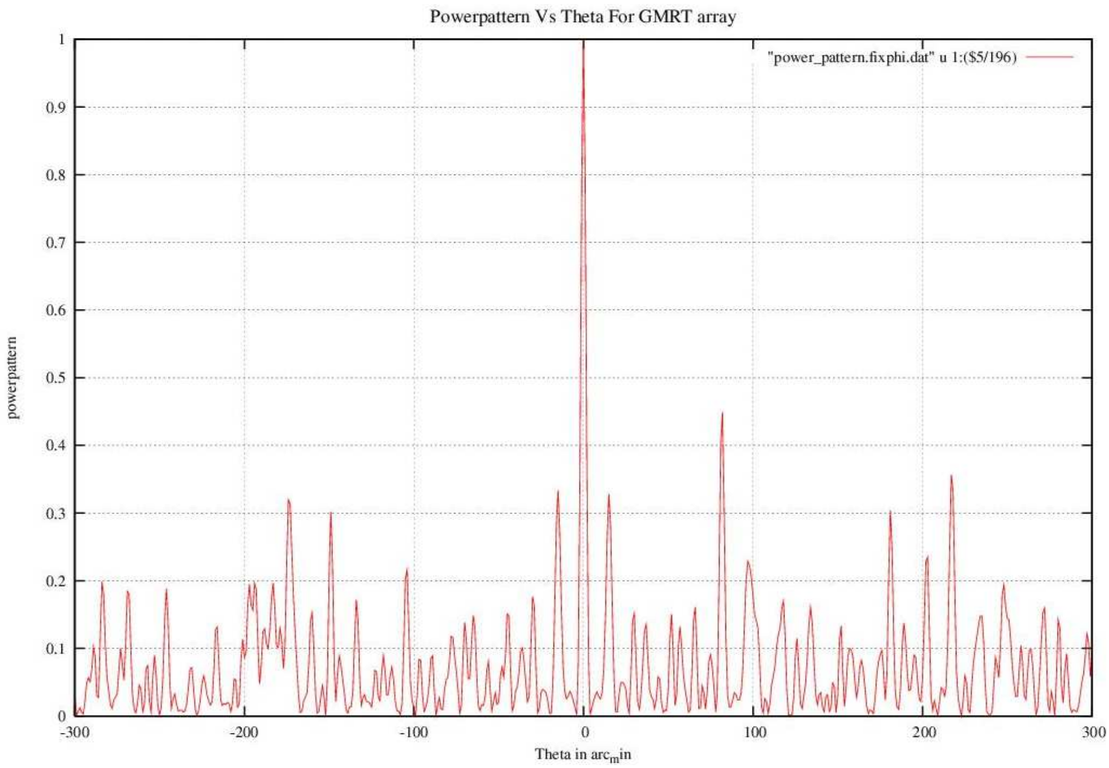


fig 3.12

From above plot: Side lobe with ~33% power occur ~20' away from θ_p

b.

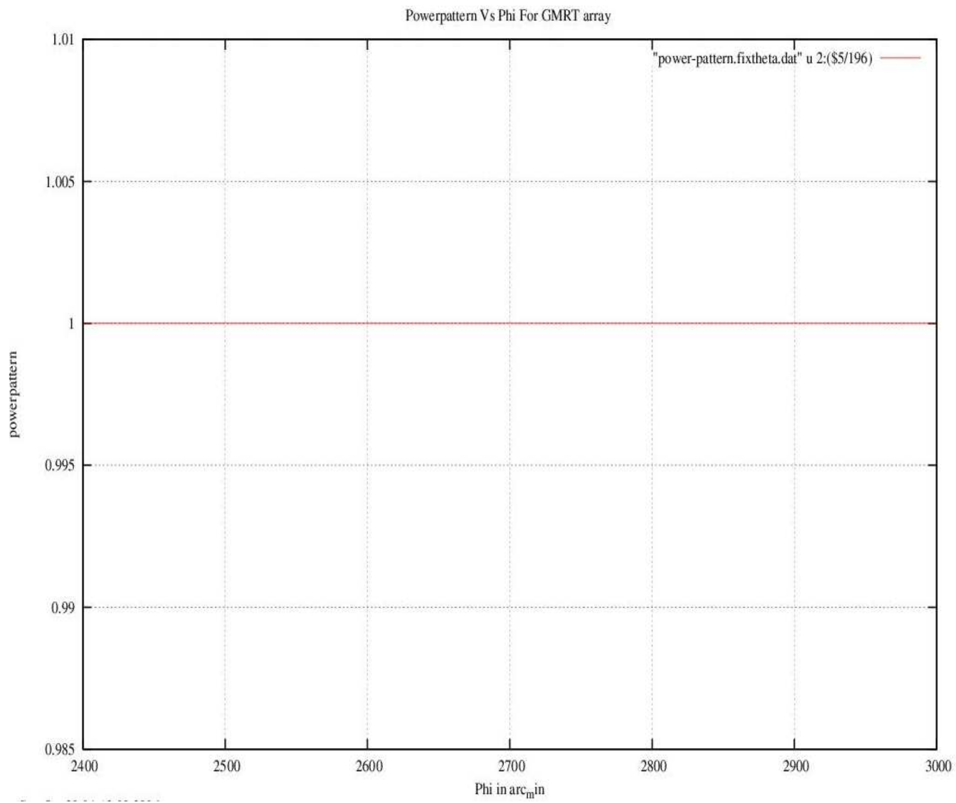


fig 3.13

Same as 3b.

5. θ_p = 45deg ϕ_p = 0 deg

θ_p = 45deg ϕ_0 = 0 deg

fixed theta=45 deg fixed phi=0 deg

a.

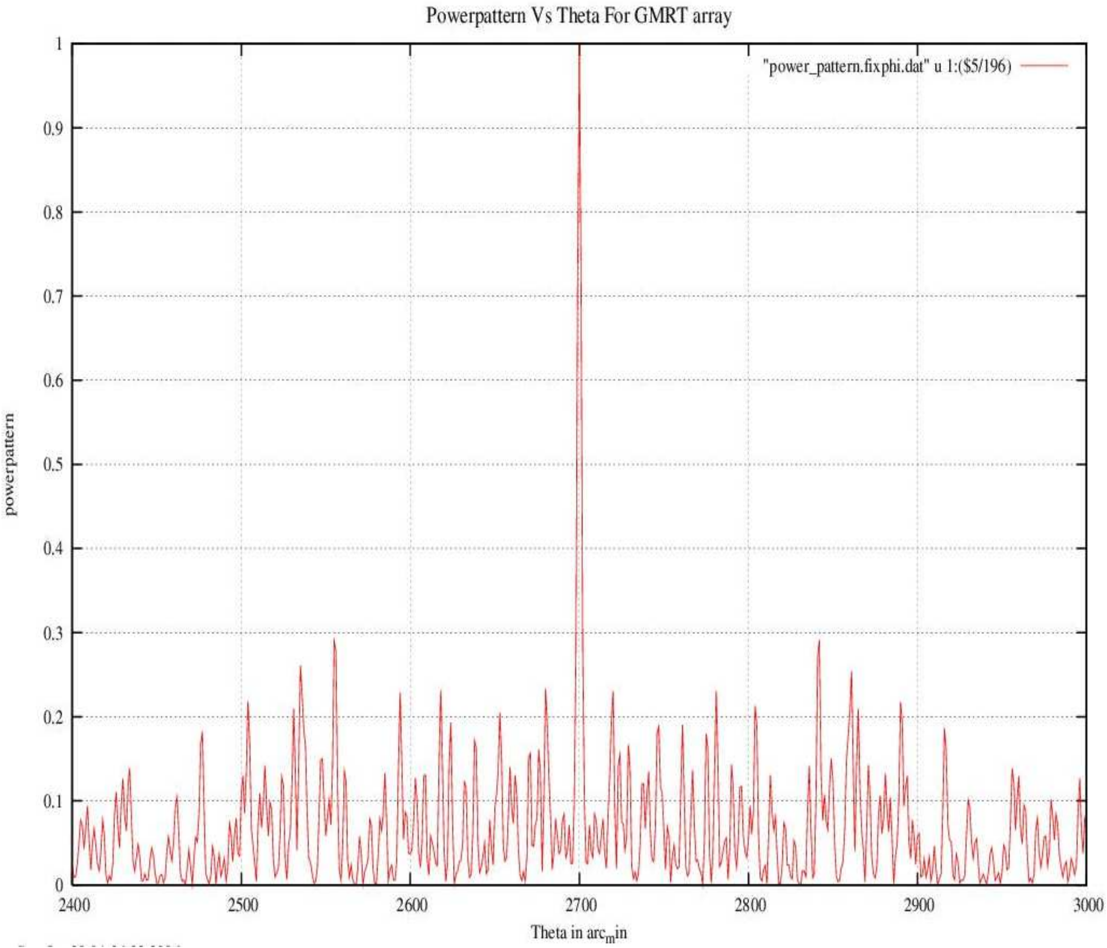


fig3.14

From above plot:side lobe power below 30% over the 5deg range.

b.

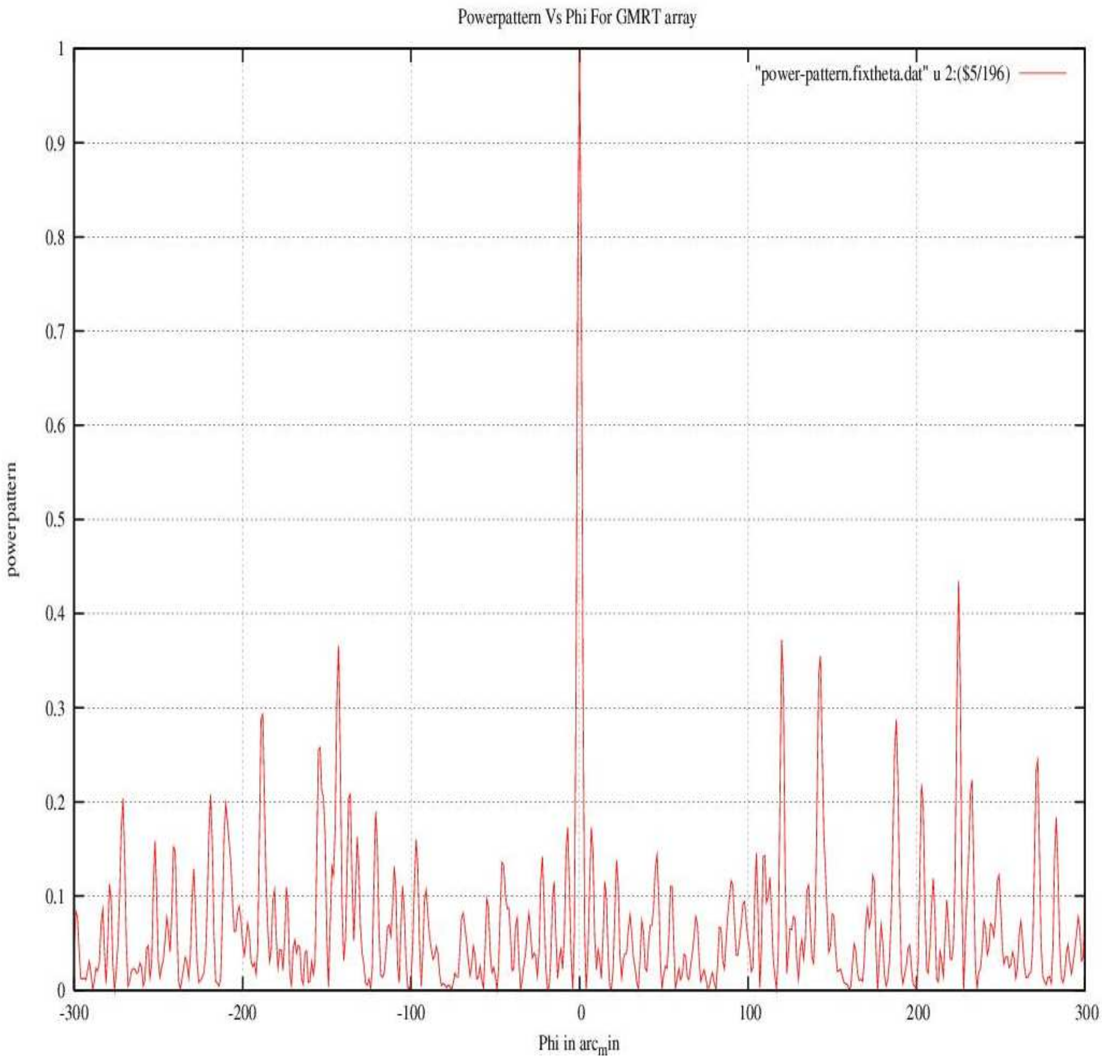


fig 3.15

From above plot:side lobe below 2% power over 100' range from ϕ_p

6. θ_p = 45deg ϕ_p = 45 deg

θ_p = 45 deg ϕ_0 = 45 deg

fixed theta=45 deg fixed phi=45 deg

a.

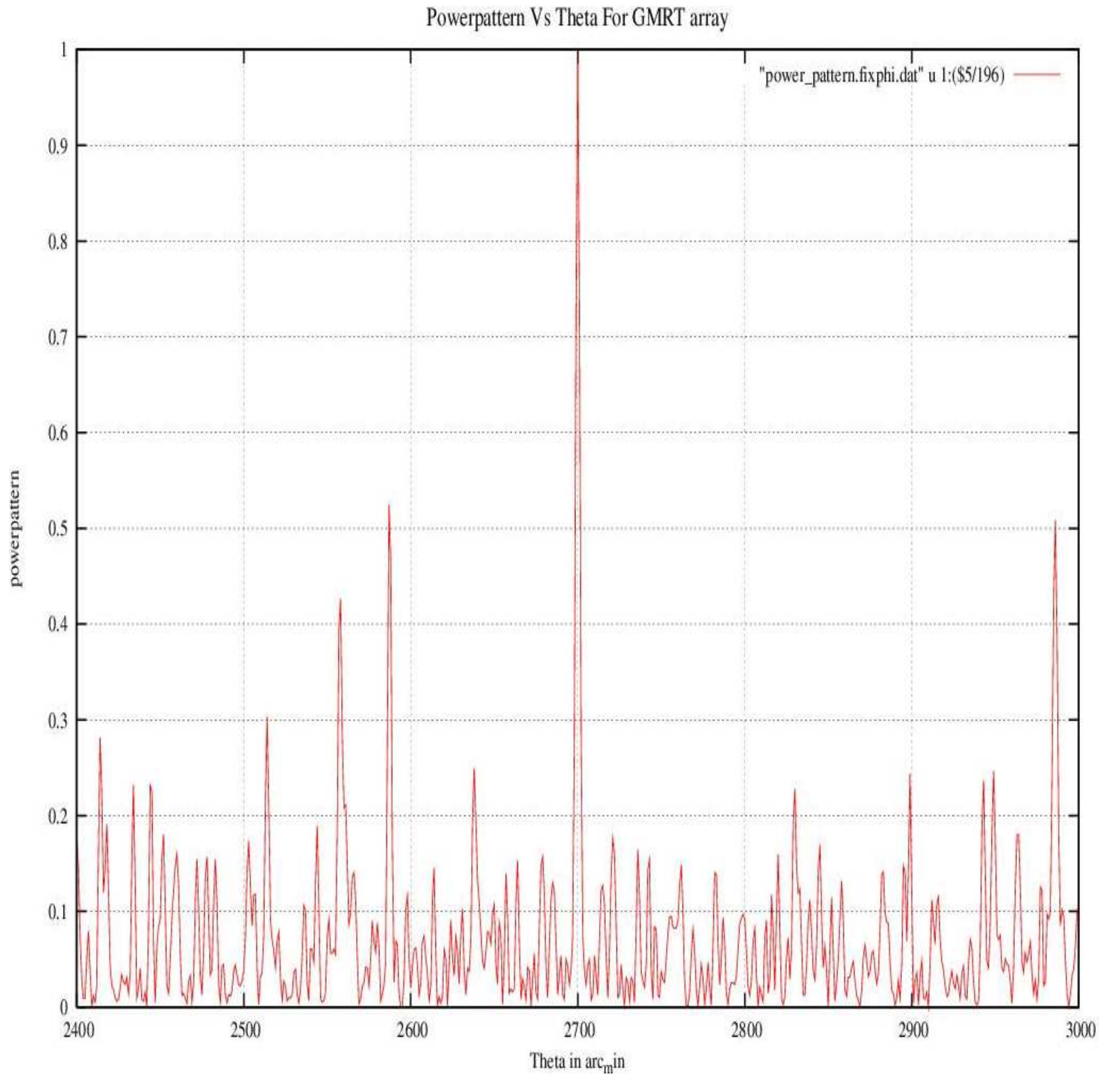


fig3.17

From above plot: side lobe with~52% power 100' away from θ_p

b.

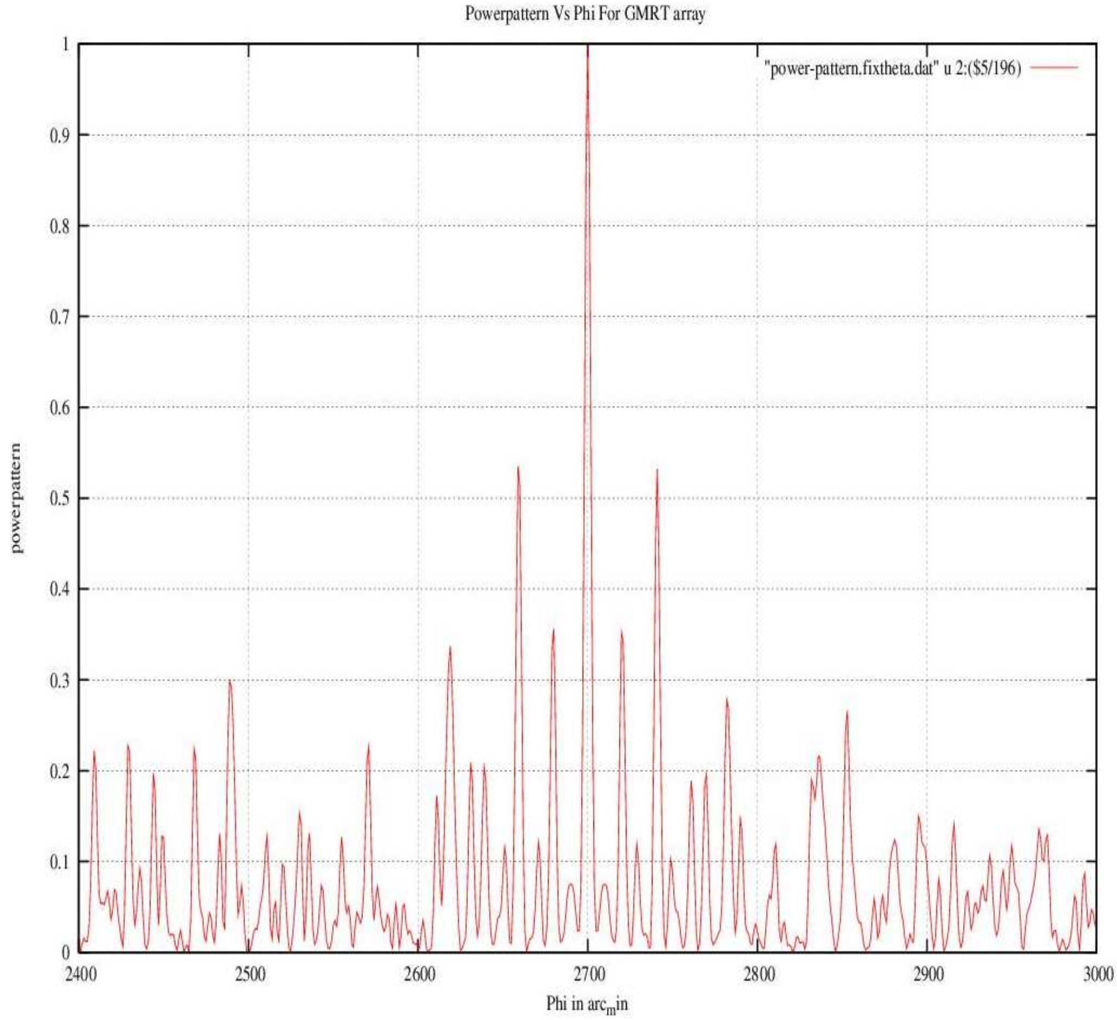


fig3.18

From above plot: side lobe with ~53% power 40' away from ϕ_p

We demonstrate a 3-D plot of the GMRT power pattern and make a contour plot with the given script file.

$\theta_p = \theta_0$ =71 deg , $\Phi_p = \Phi_0$ = 0deg

Range of theta =0.5deg , Range of phi = 1deg

Accuracy of theta=0.2' ,Accuracy of phi=20'

The script file used to plot graph for the above parameters is :

set term post landscape enhanced colour colourtext "Times NewRoman" 10
set out "test.ps"
set autoscale
set xlabel 'Theta in arc_min'
set ylabel 'phi in arc_min'
set zlabel 'powerpattern'
set title 'Powerpattern Vs Theta For GMRT array ' font "Times NewRoman, 12"
show title
set time
set grid
set contour
set hidden3d
splot "power_pattern.dat" u 1:2:($5/196) w l

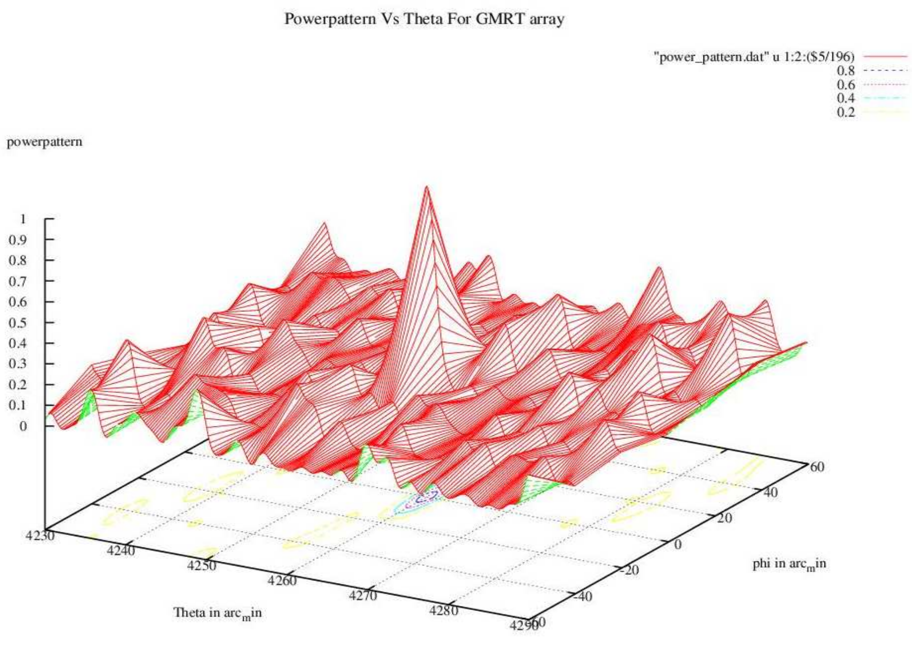


fig3.19

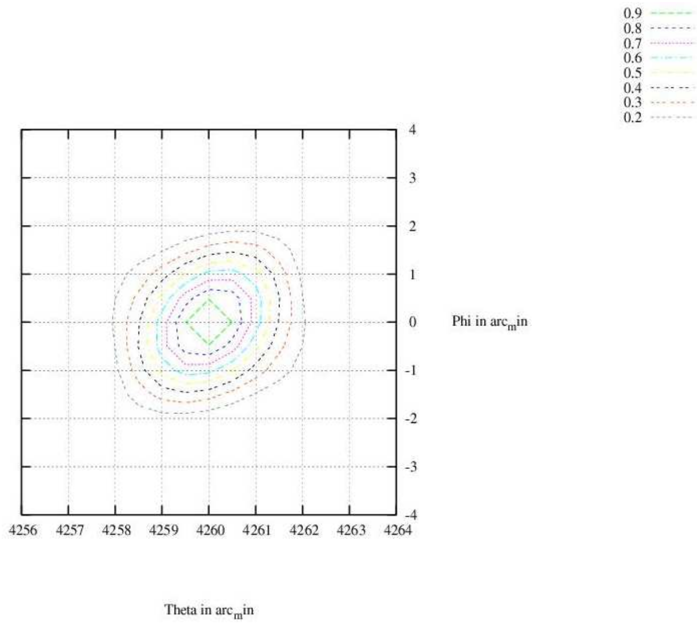


Fig 3.20

set term post landscape enhanced colour colourtext "Times NewRoman" 10
set out "test.ps"
set autoscale
set xlabel 'Theta in arc_min'
set ylabel 'phi in arc_min'
set zlabel 'powerpattern'
set title 'Powerpattern Vs Theta For GMRT array phased in any direction in theta phi plane , showing contour' font "Times NewRoman, 12"
show title
set time
set grid
set contour
set dgrid3d 90,50,8
set cntrparam levels 2
set cntrparam levels incremental .9,.1,1
splot "power_theta_p0.dat" u 3:4:($5/196) w l, "power_theta_p1.dat" u 3:4:($5/196) w l,"power_theta_p2.dat" u 3:4:($5/196) w l,"power_theta_p3.dat" u 3:4:($5/196) w l,"power_theta_p4.dat" u 3:4:($5/196) w l,"power_theta_p5.dat" u 3:4:($5/196) w l,"power_theta_p6.dat" u 3:4:($5/196) w l,"power_theta_p7.dat" u 3:4:($5/196) w l,"power_theta_p8.dat" u 3:4:($5/196) w l

CHAPTER 4

MODIFICTION OF THE C CODE

The program is modified with the concept in mind that multiple beams is needed to be generated to track an extended source. This is done by programming the code for multiple values of $\theta_{p.}$

In the first stage,a for loop of θ_p is run over and above the loop of phi and theta. This generates 3 discrete beams phased at 0 , 0.5 ,1 deg in theta. The power of each of these phasing is put into 3 individual files and their plot taken together as shown.

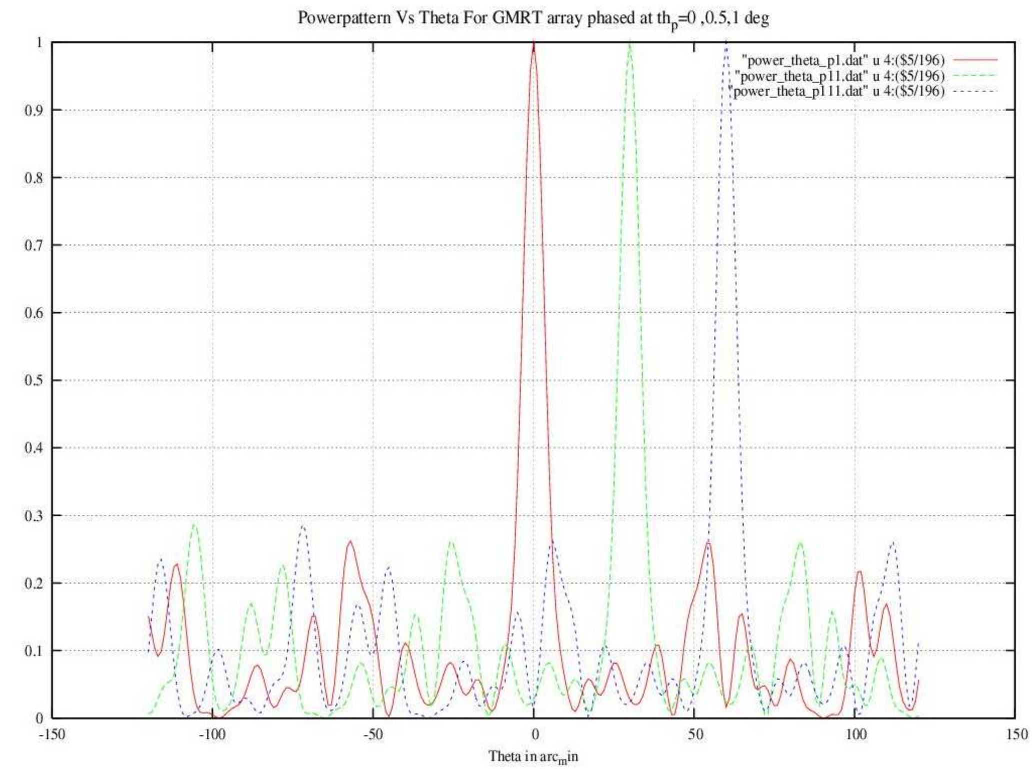


fig 4.1

The variation of power pattern with phi is as shown below.

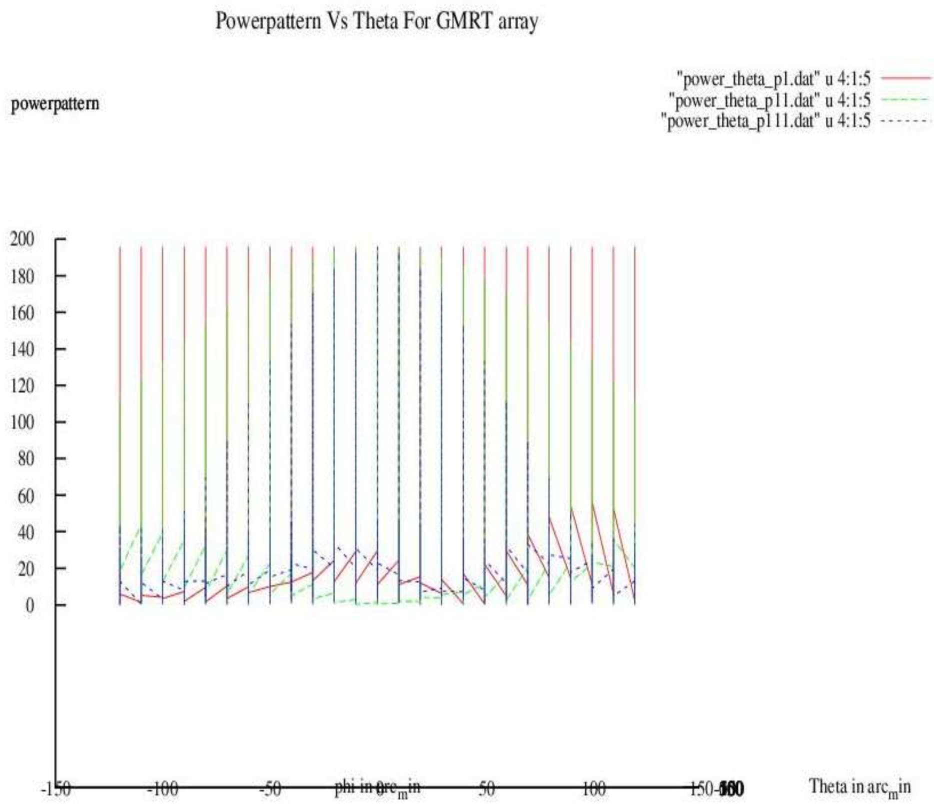


fig4.2

From the above plot we can see that when $\theta_p = 0$ deg there is no variation in the power pattern with phi .This can explained from the equations below.

```
x_comp=x[i]*(sin(min*th)*cos(min*phi)-sin(min*th_P*60)*cos(min*phi_P*60));
y_comp=y[i]*(sin(min*th)*sin(min*phi)-sin(min*th_P*60)*sin(min*phi_P*60));
z_comp=z[i]*(cos(min*th)-cos(min*th_P*60));
```

When $\theta_p = 0$,the contribution of phi part to the power components is constant.
But when $\theta_p \neq 0$,the phi variations is as shown.

To generate multiple beams in 2-D we seek variation both in theta and phi, so that beams phased at any θ_p and ϕ_p can be obtained. The separation of the phased beams in theta and phi is provided by the user. A corresponding data file is generated which stores the value of theta and phi .

If phasing directions are theta_P and phi_P and number of beams to be generated along either side of theta_P , ph_P =X(say) , then total number of multiple beams generated =(2X-1)*(2X-1).

Below are 2 examples of this:

1)Here,
theta_p=0deg,phi_P=0deg
theta_0=0deg,phi_0=0deg.
range of theta=2deg,range of phi=2 deg
accuracy in theta=10' ,accuracy in phi=10'
No. of multiple beams to be generated in theta and phi= 2
separation in between beams =60'.

The input file is as shown: ("input_thp_phip.dat")

# theta_P	phi_P
-60.000000	-60.000000
-60.000000	0.000000
-60.000000	60.000000
0.000000	-60.000000
0.000000	0.000000
0.000000	60.000000
60.000000	-60.000000
60.000000	0.000000
60.000000	60.000000

The 3-D plot is as shown:
The above plot along the axis is as shown:

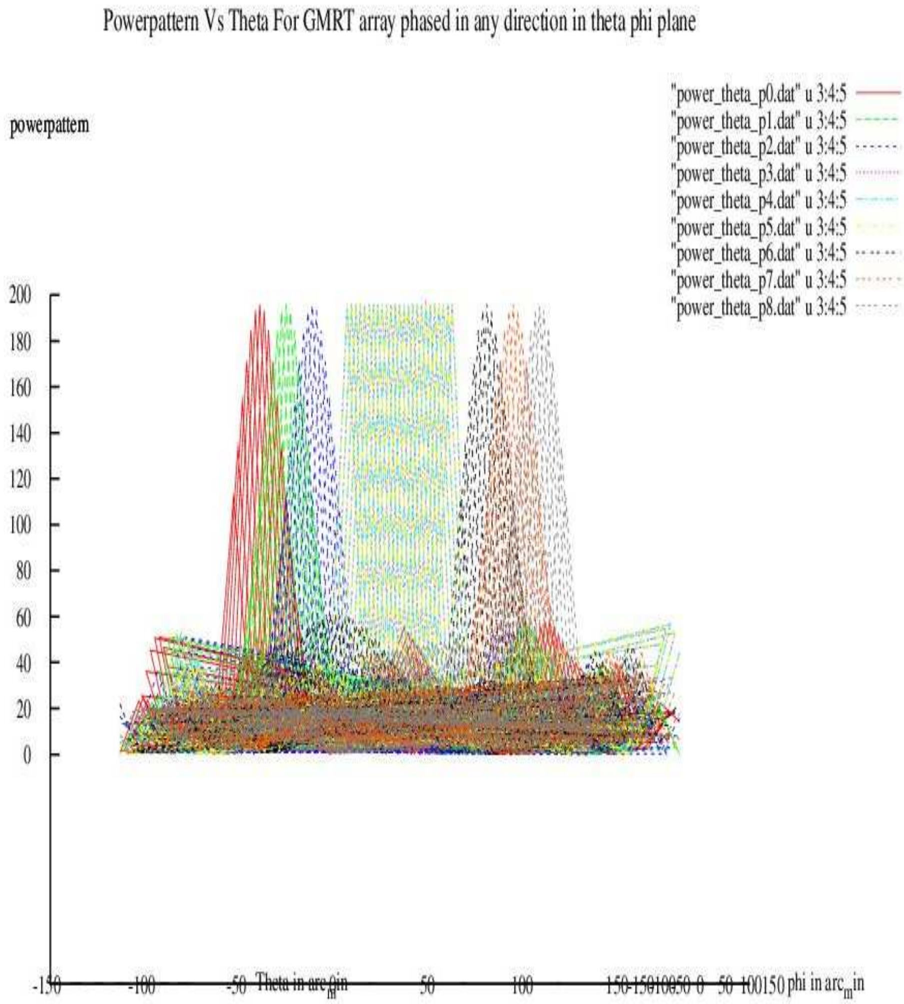

Powerpattern Vs Theta For GMRT array phased in any direction in theta phi plane
powerpattern
"power_theta_p0.dat" u 3:4:5
"power_theta_p1.dat" u 3:4:5
"power_theta_p2.dat" u 3:4:5
"power_theta_p3.dat" u 3:4:5
"power_theta_p4.dat" u 3:4:5
"power_theta_p5.dat" u 3:4:5
"power_theta_p6.dat" u 3:4:5
"power_theta_p7.dat" u 3:4:5
"power_theta_p8.dat" u 3:4:5
Theta in arc min
phi in arc min

fig 4.3

Powerpattern Vs Theta For GMRT array phased in any direction in theta phi plane

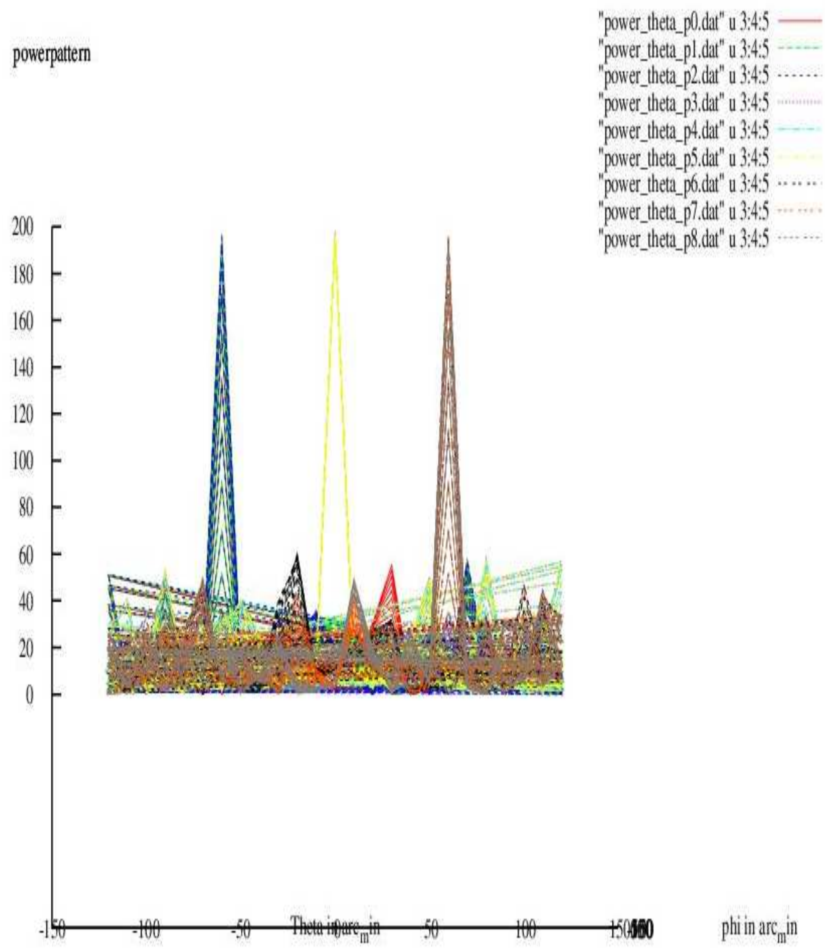


fig 4.4

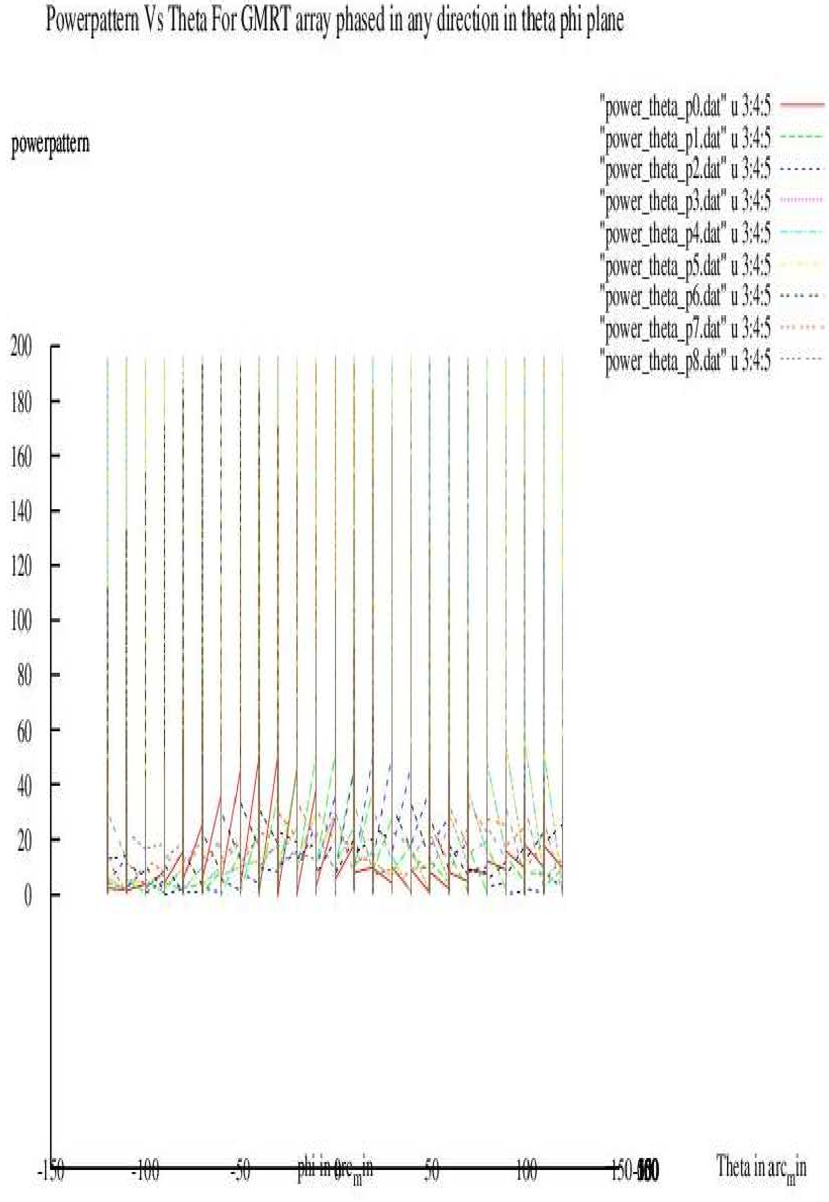


fig 4.5

The plot along phi axis is as shown:

The script file used for contour plot is as given:

```
set term post landscape enhanced colour colourtext "Times NewRoman" 10
set out "test.ps"
set autoscale
set xlabel 'Theta in arc_min'
set ylabel 'phi in arc_min'
set zlabel 'powerpattern'
set title 'Powerpattern Vs Theta For GMRT array phased in any direction in theta phi plane , showing contour' font "Times NewRoman, 12"
show title
set time
set grid
set contour
set dgrid3d 90,50,8
set cntrparam levels 2
set cntrparam levels incremental .9,.1,1
splot "power_theta_p0.dat" u 3:4:($5/196) w l, "power_theta_p1.dat" u 3:4:($5/196) w l,"power_theta_p2.dat" u 3:4:($5/196) w l,"power_theta_p3.dat" u 3:4:($5/196) w l,"power_theta_p4.dat" u 3:4:($5/196) w l,"power_theta_p5.dat" u 3:4:($5/196) w l,"power_theta_p6.dat" u 3:4:($5/196) w l,"power_theta_p7.dat" u 3:4:($5/196) w l,"power_theta_p8.dat" u 3:4:($5/196) w l
```

The contour plot is as shown:

1) For 50% power levels:

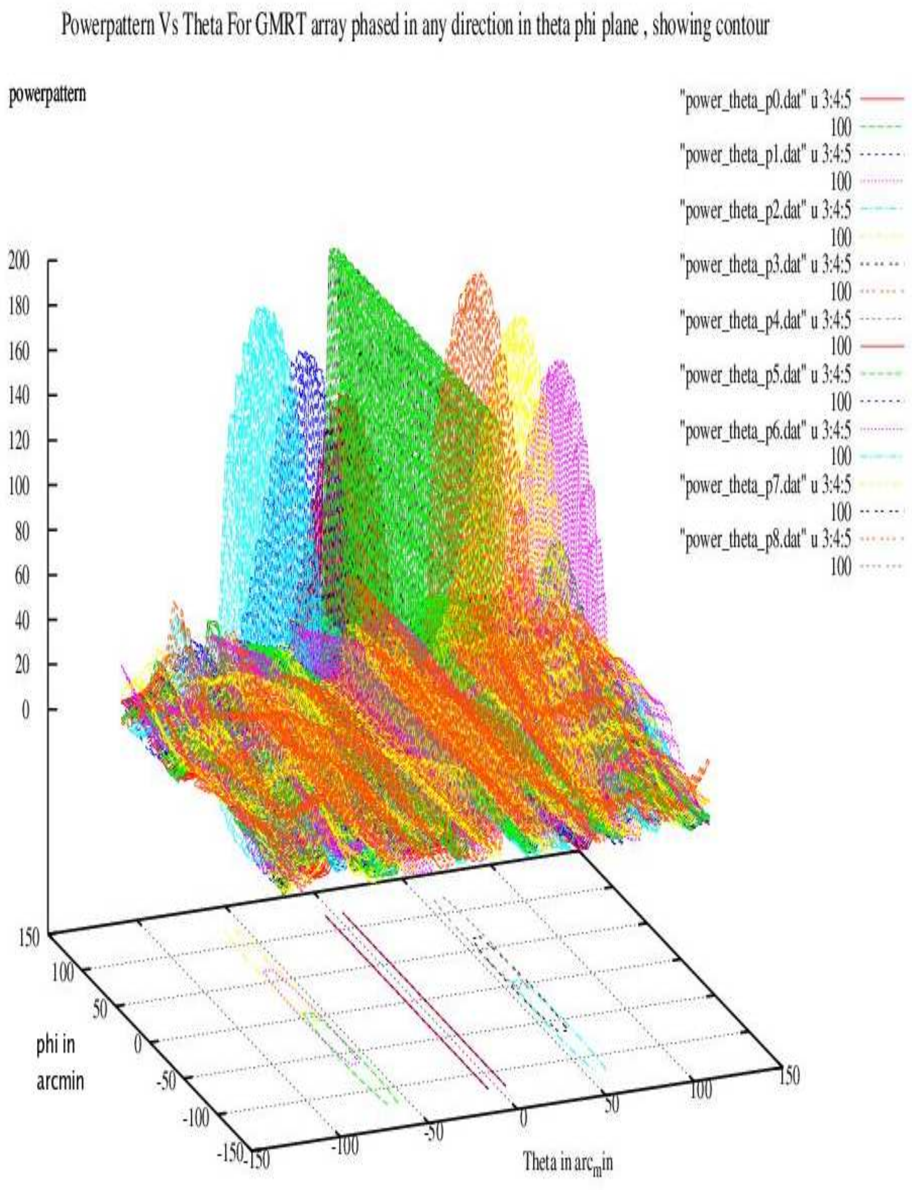


fig 4.6

2) Contour with 90% power levels

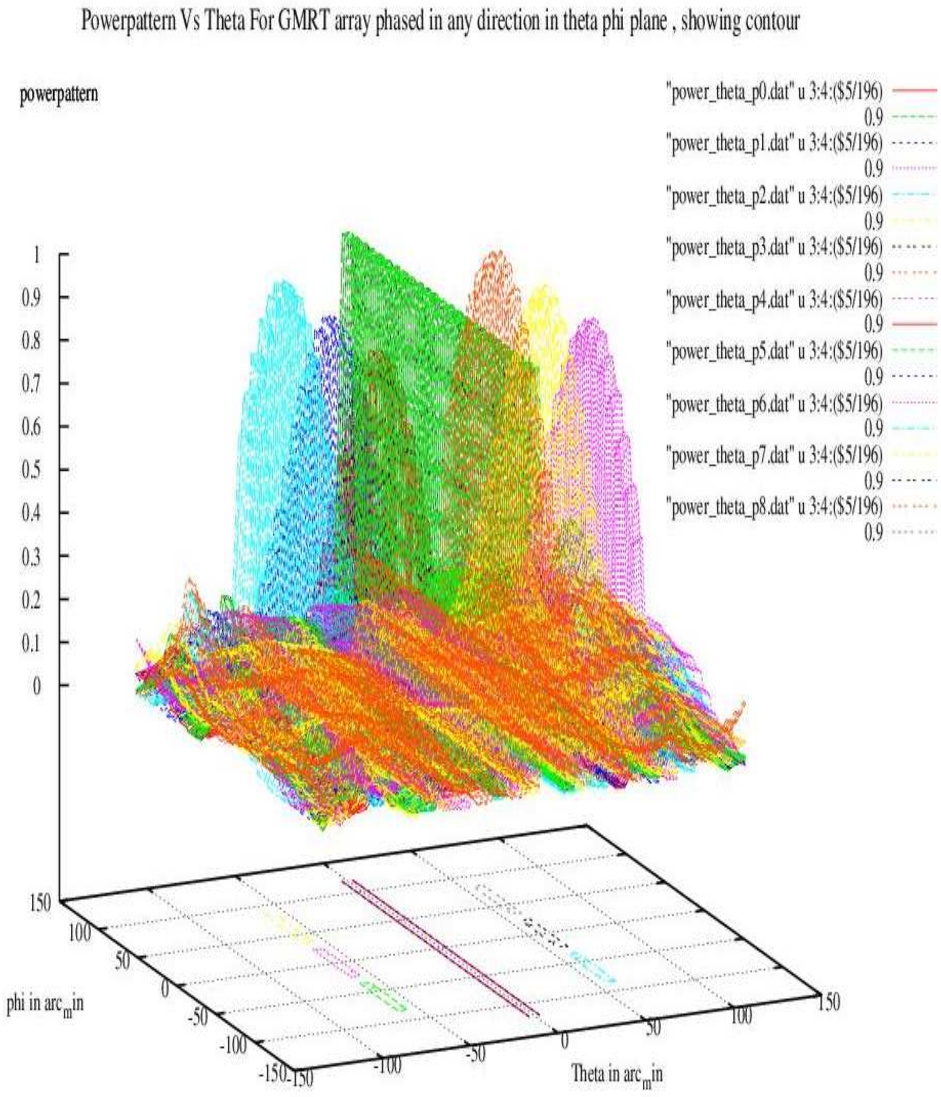


fig 4.7

With unset surface:

Powerpattern Vs Theta For GMRT array phased in any direction in theta phi plane , showing contour

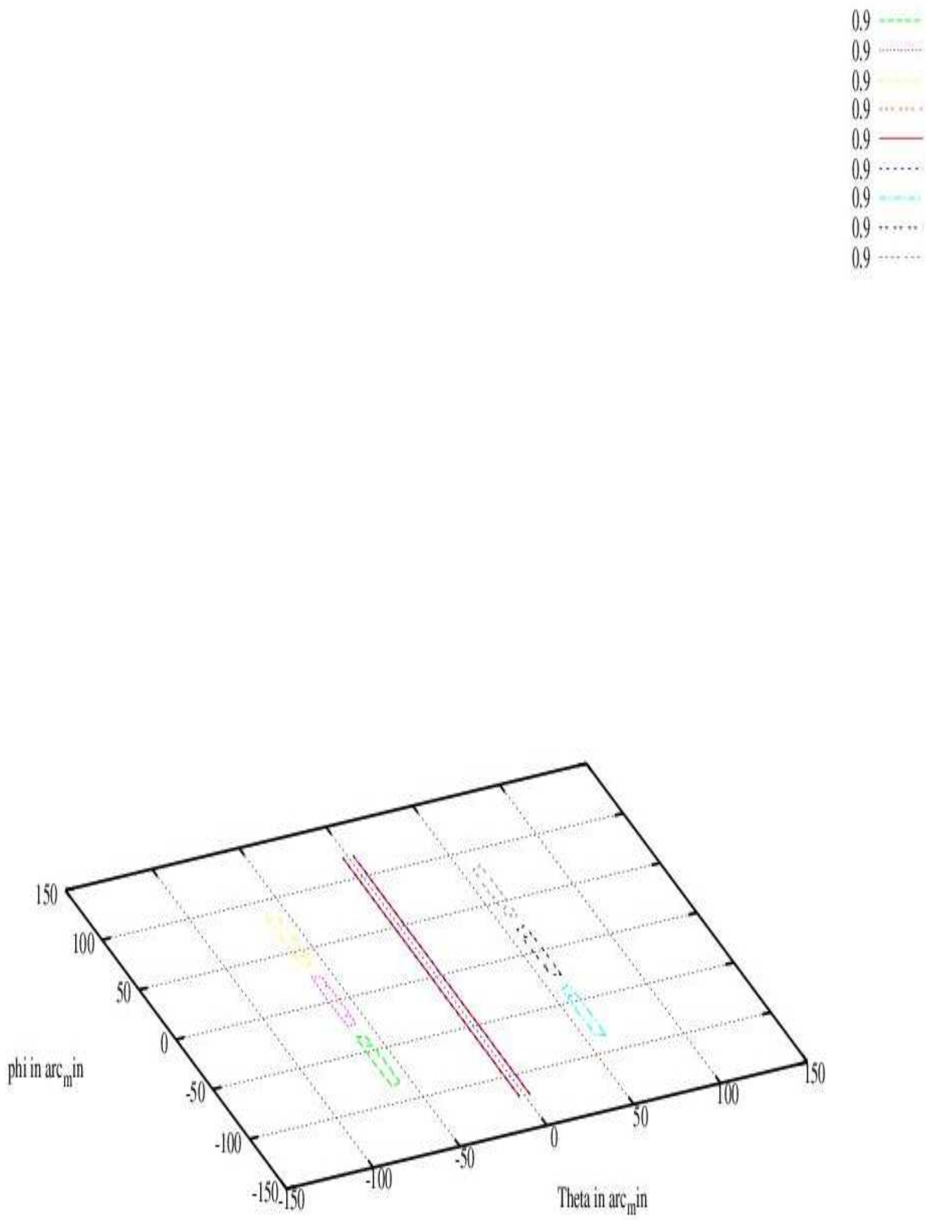


fig:4.8

2a)Here,
theta_p=71deg,phi_P=0deg
theta_0=71deg,phi_0=0deg.
range of theta=0.5deg,   range of phi = 0.5 deg
accuracy in theta=0.2'   ,accuracy in phi=0.2'
No. of multiple beams to be generated in theta and phi= 2
separation in between beams =2'.   (both in theta and phi direction)
The input data file is as shown: ("input_thp_phip.dat")

```
# theta_P         phi_P
4257.000000   -3.000000
4257.000000    0.000000
4257.000000    3.000000
4260.000000    -3.000000
4260.000000    0.000000
4260.000000    3.000000
4263.000000   -3.000000
4263.000000    0.000000
4263.000000    3.000000
```

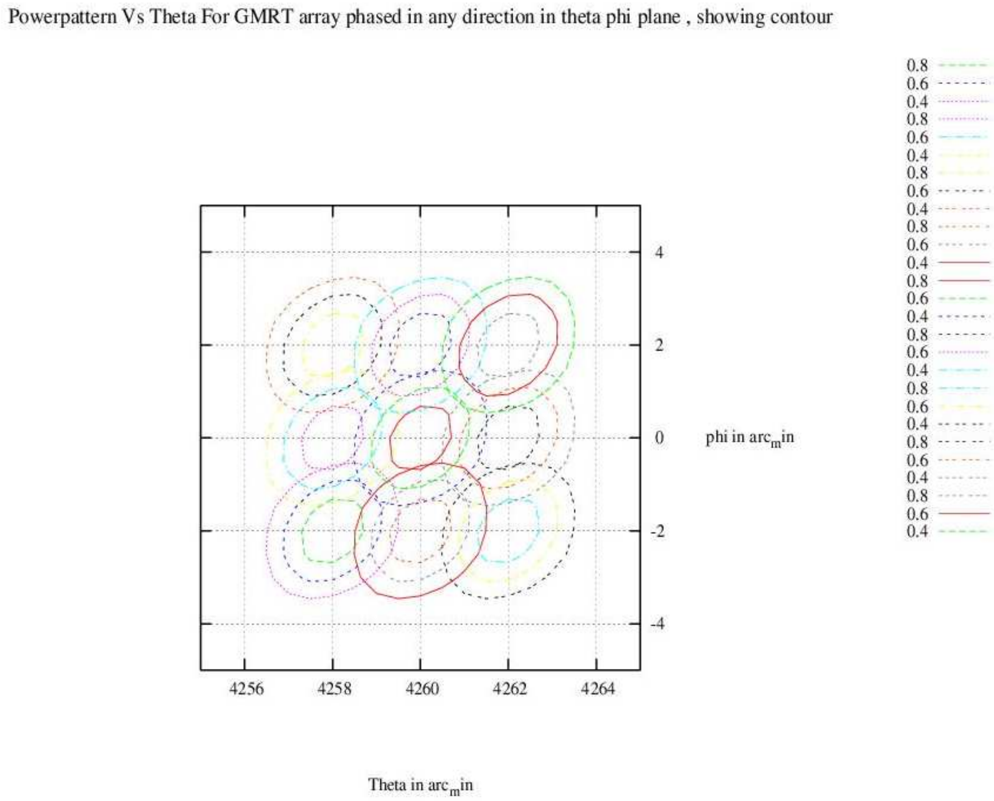

Powerpattern Vs Theta For GMRT array phased in any direction in theta phi plane , showing contour
0.8
0.6
0.4
0.8
0.6
0.4
0.8
0.6
0.4
0.8
0.6
0.4
0.8
0.6
0.4
0.8
0.6
0.4
0.8
0.6
0.4
0.8
0.6
0.4
0.8
0.6
0.4
4
2
0
-2
-4
phi in $arc_{m}in$
4256
4258
4260
4262
4264
Theta in $arc_{m}in$

Fig 4.9

2b) theta_p=71deg,phi_P=0deg

theta_0=71deg,phi_0=0deg.

range of theta=0.2deg, range of phi = 0.2deg

accuracy in theta=0.2' ,accuracy in phi=0.2'

No. of multiple beams to be generated in theta and phi= 2

separation in between beams =5'. (both in theta and phi direction)

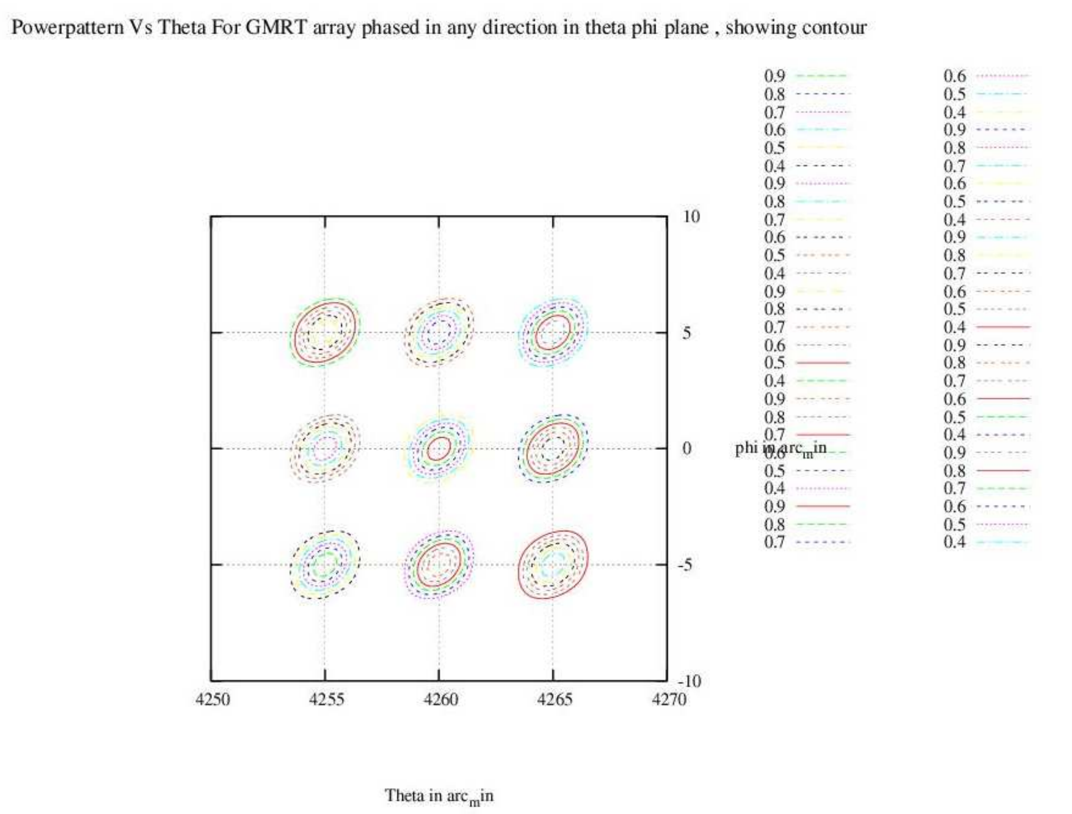


Fig 4.10

2c) theta_p=71deg,phi_P=0deg

theta_0=71deg,phi_0=0deg.

range of theta=0.2deg, range of phi = 0.2deg

accuracy in theta=0.2' ,accuracy in phi=0.2'

No. of multiple beams to be generated in theta and phi= 2

separation in between beams =3'. (both in theta and phi direction)

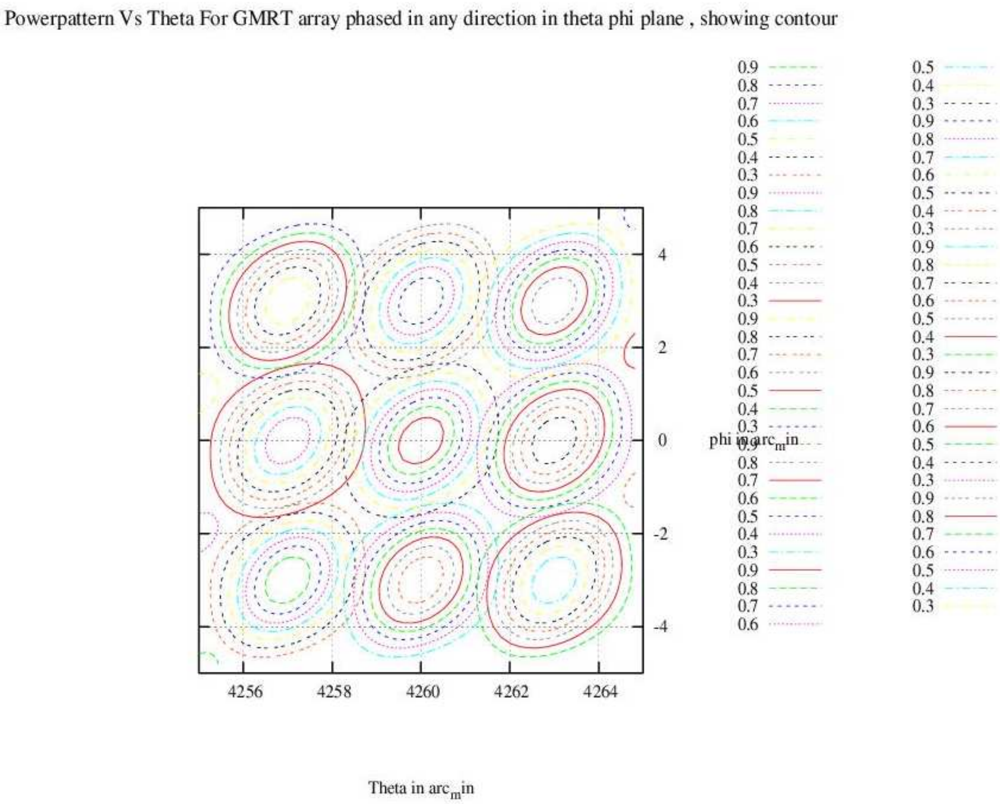


Fig 4.11

The script file for the above thre graphs is:

```
set term post landscape enhanced colour colourtext "Times NewRoman" 10
set out "test.ps"
set autoscale
set xlabel 'Theta in arc_min'
set ylabel 'phi in arc_min'
set zlabel 'powerpattern'
set title 'Powerpattern Vs Theta For GMRT array phased in any direction in theta phi plane , showing contour' font "Times NewRoman, 12"
show title
set time
set grid
set contour
set cntrparam levels 2
set cntrparam levels incremental .3,.1,1
unset surface
set xrange[4255:4265]
set yrange[-5:5]
set view 1,0,1,1
set size square
splot "power_theta_p0.dat" u 3:4:($5/196) w l, "power_theta_p1.dat" u 3:4:($5/196) w l,"power_theta_p2.dat" u 3:4:($5/196) w l,"power_theta_p3.dat" u 3:4:($5/196) w l,"power_theta_p4.dat" u 3:4:($5/196) w l,"power_theta_p5.dat" u 3:4:($5/196) w l,"power_theta_p6.dat" u 3:4:($5/196) w l,"power_theta_p7.dat" u 3:4:($5/196) w l,"power_theta_p8.dat" u 3:4:($5/196) w l
```

CHAPTER 5

OBSERVATION ON GMRT ARRAY(CENTRAL SQUARE + ARM ANTENNA):

We now extend the phased array of the GMRT from the central square symmetrically to the arm antennas. The power pattern are as observed in the plots below. The plot are taken both for fixed theta and fixed phi.

Here : $\lambda = 1m$

$\theta_p = \theta_0 = 45$ deg

$\phi_p = \phi_p = 0$ deg

Range of theta = range of phi = 3 deg

Accuracy of theta = Accuracy of phi = 1'

1) For 17 antenna coordinates (Central Square + E02 +S01+W01)

a)

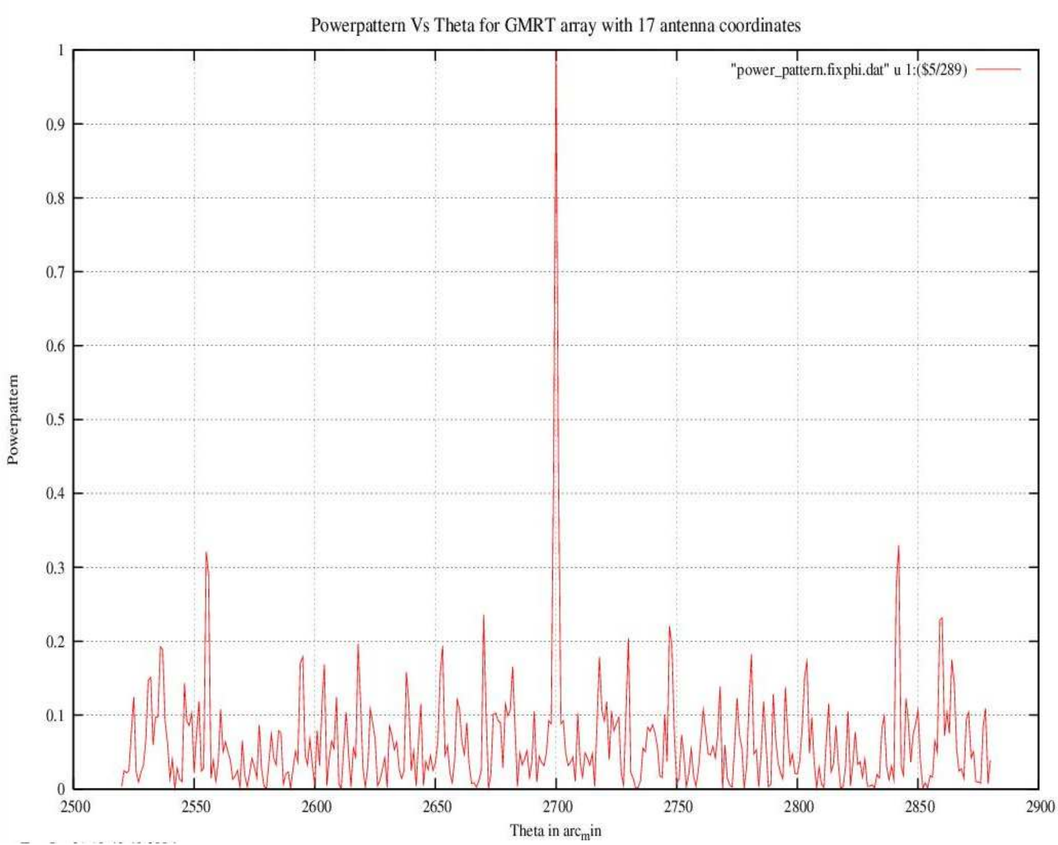


fig5.1

1 b)

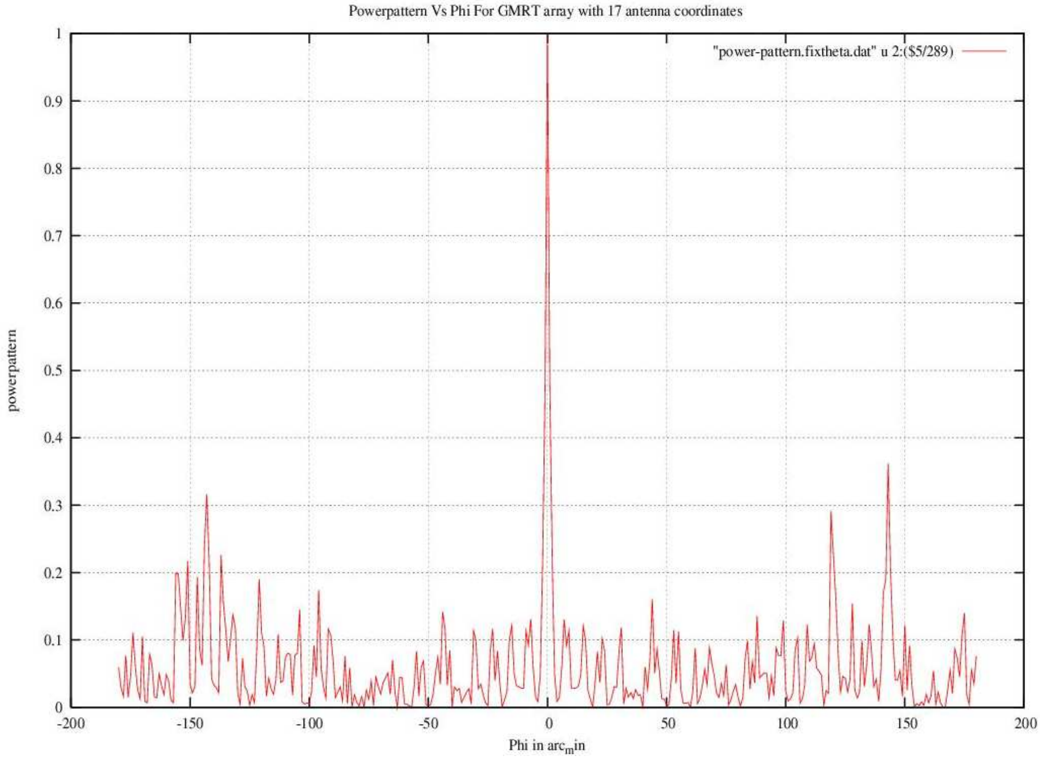

Powerpattern Vs Phi For GMRT array with 17 antenna coordinates
"power-pattern.fixtheta.dat" u 2:($5/289)
powerpattern
Phi in arc_min
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-200
-150
-100
-50
0
50
100
150
200

fig5.2

2) For 20 antenna coordinates (Central Square + E02 +E03+S01+S02+W01+W02)

a)

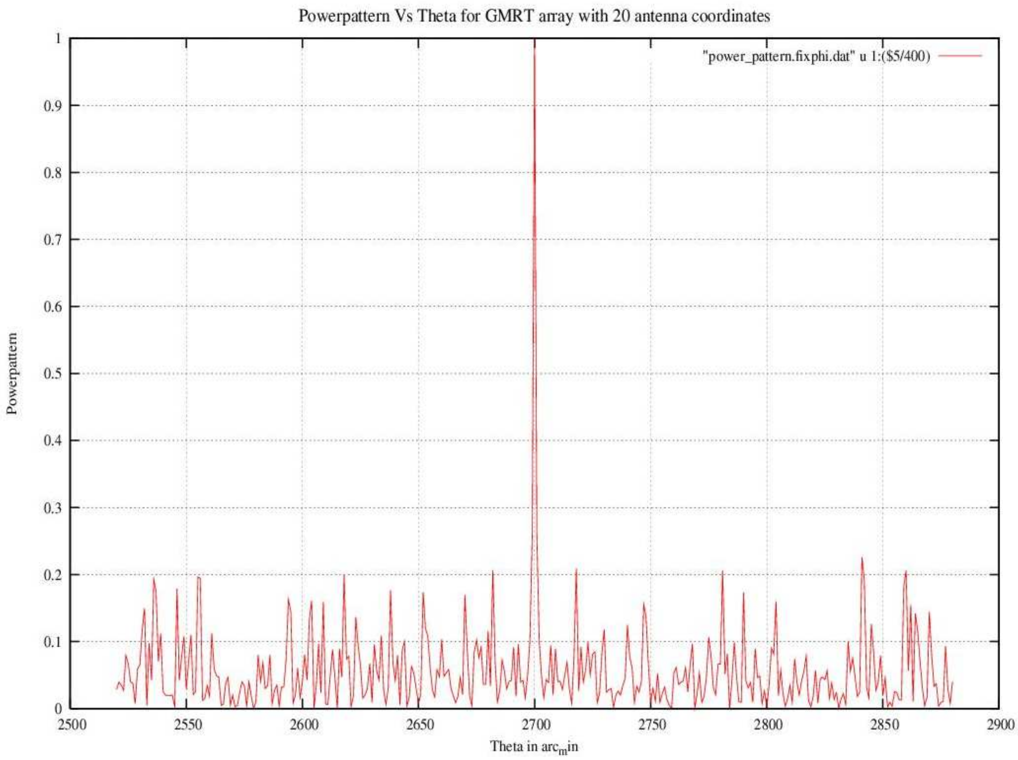


fig5.3

2b)

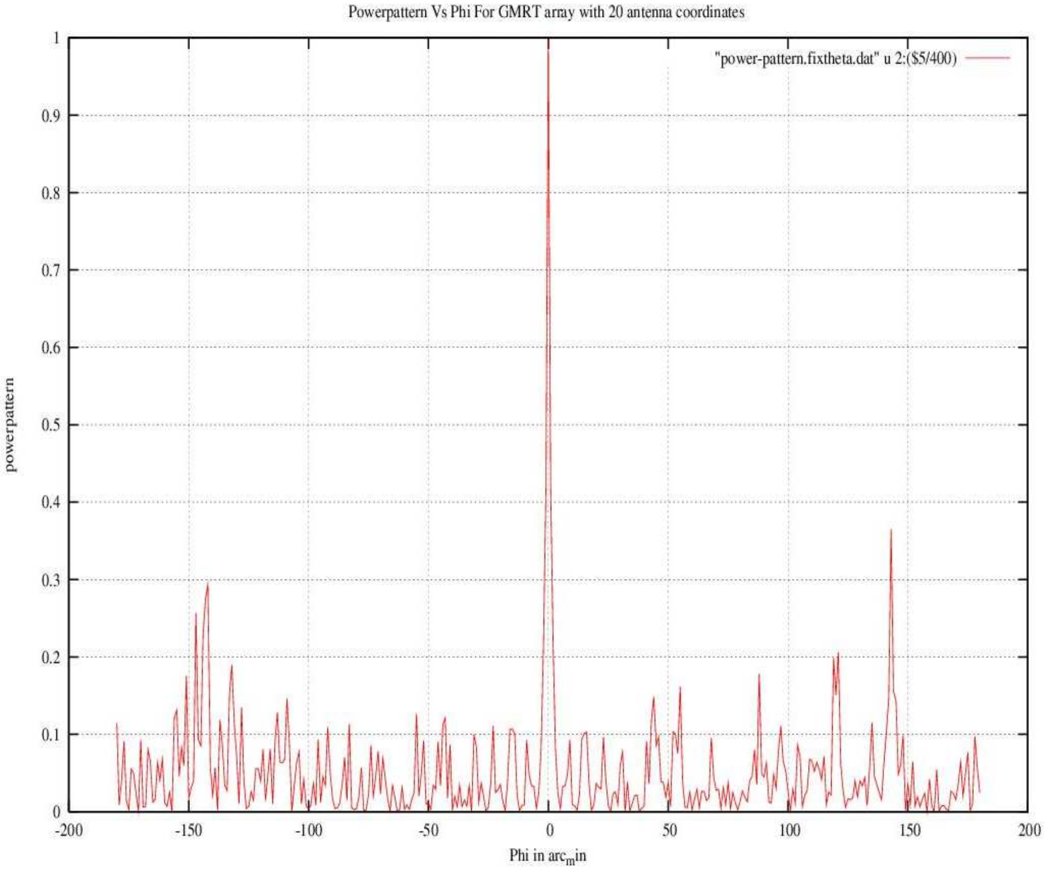

Powerpattern Vs Phi For GMRT array with 20 antenna coordinates
"power-pattern.fixtheta.dat" u 2:($5/400)
powerpattern
Phi in arc_min
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-200
-150
-100
-50
0
50
100
150
200

fig5.4

3)For 23 antenna coordinates (Central Square + E02 E03+E04+S01+S02+S03+W01+W02+W03)

a)

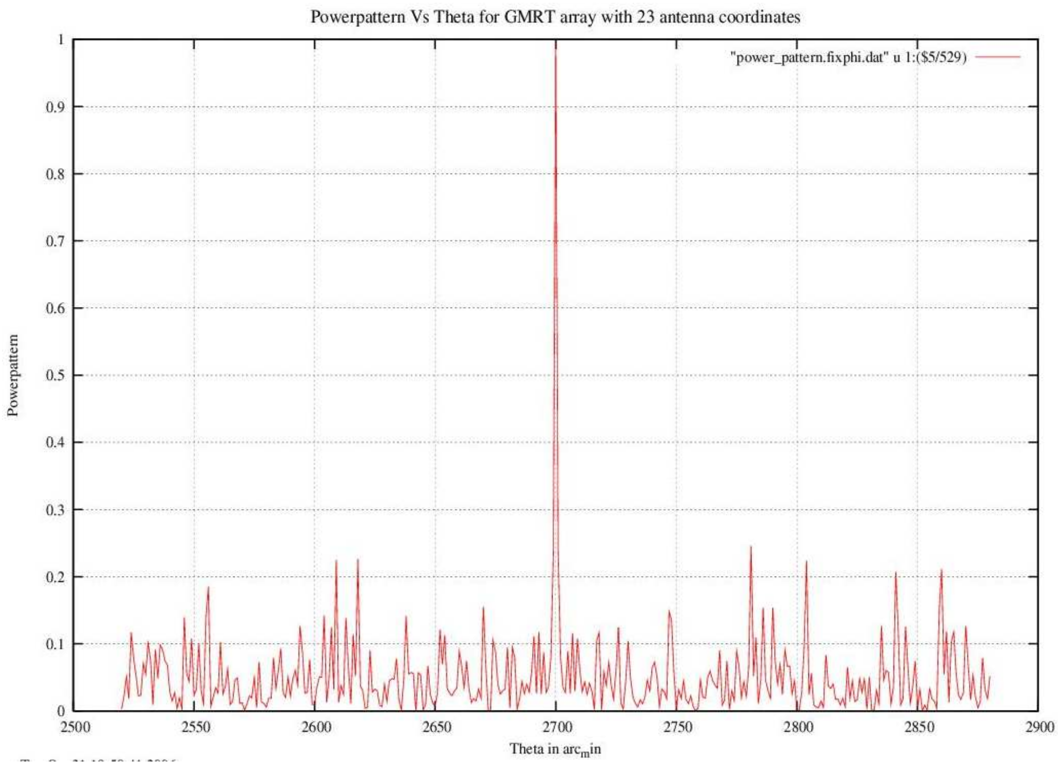
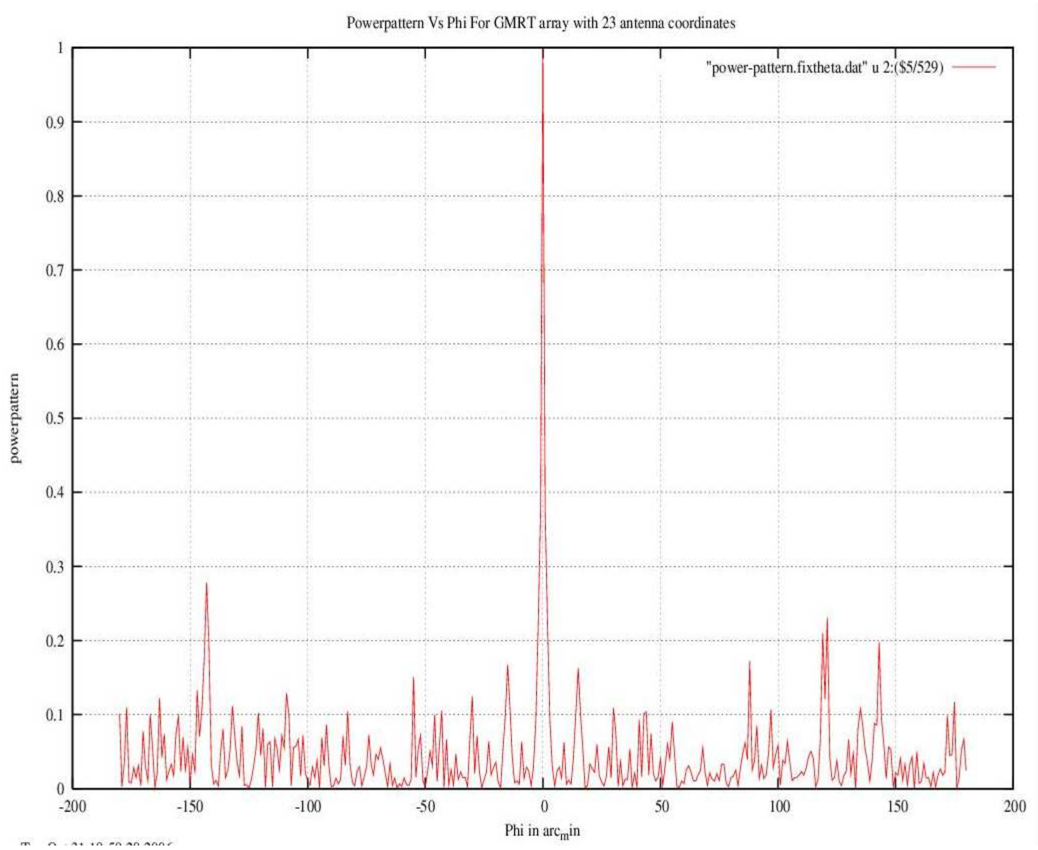


fig5.5

3 b)



Powerpattern Vs Phi For GMRT array with 23 antenna coordinates
"power-pattern.fixtheta.dat" u 2:($5/529)
powerpattern
Phi in arc_min
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-200
-150
-100
-50
0
50
100
150
200

fig5.6

4)For 26 antenna coordinates (Central Square + E02 +E03+E04+E05+S01+S02+S03+S04+W01+W02+W03+W04)

a)

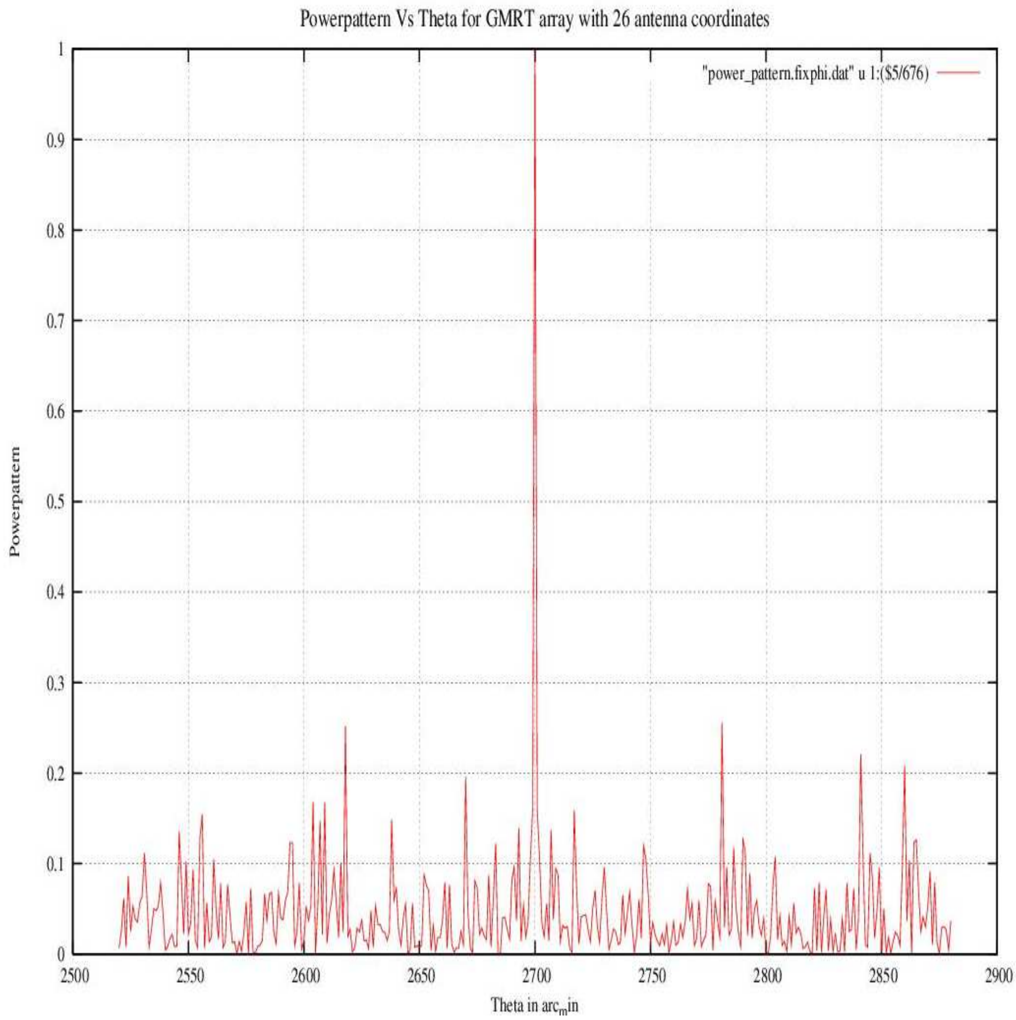
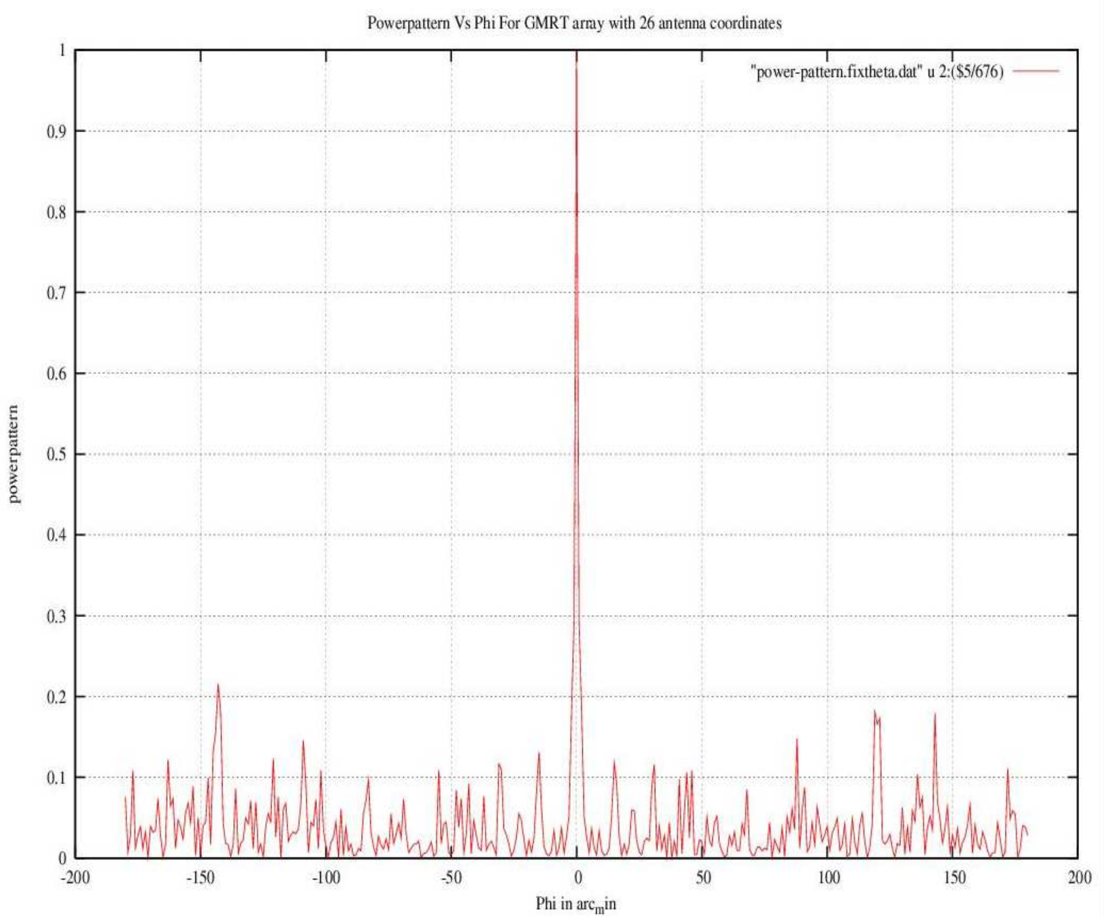


fig5.7

4 b)



Powerpattern Vs Phi For GMRT array with 26 antenna coordinates
"power-pattern.fixtheta.dat" u 2:($5/676)
powerpattern
Phi in arc_min
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-200
-150
-100
-50
0
50
100
150
200

fig5.8

5) For 29 antenna coordinates (Central Square + E02 +E03+E04+E05+E06+S01+S02+S03+S04+S05+W01+W02+W03+W04+W05)

a)

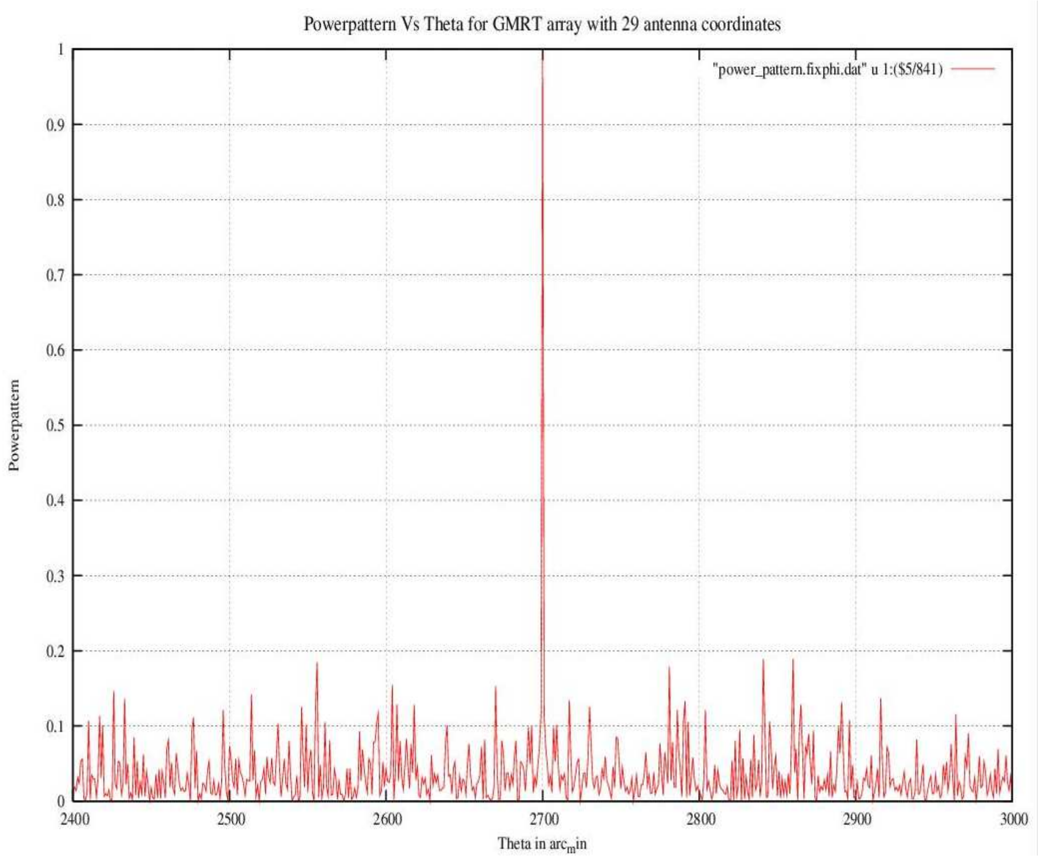


fig5.9

5 b)

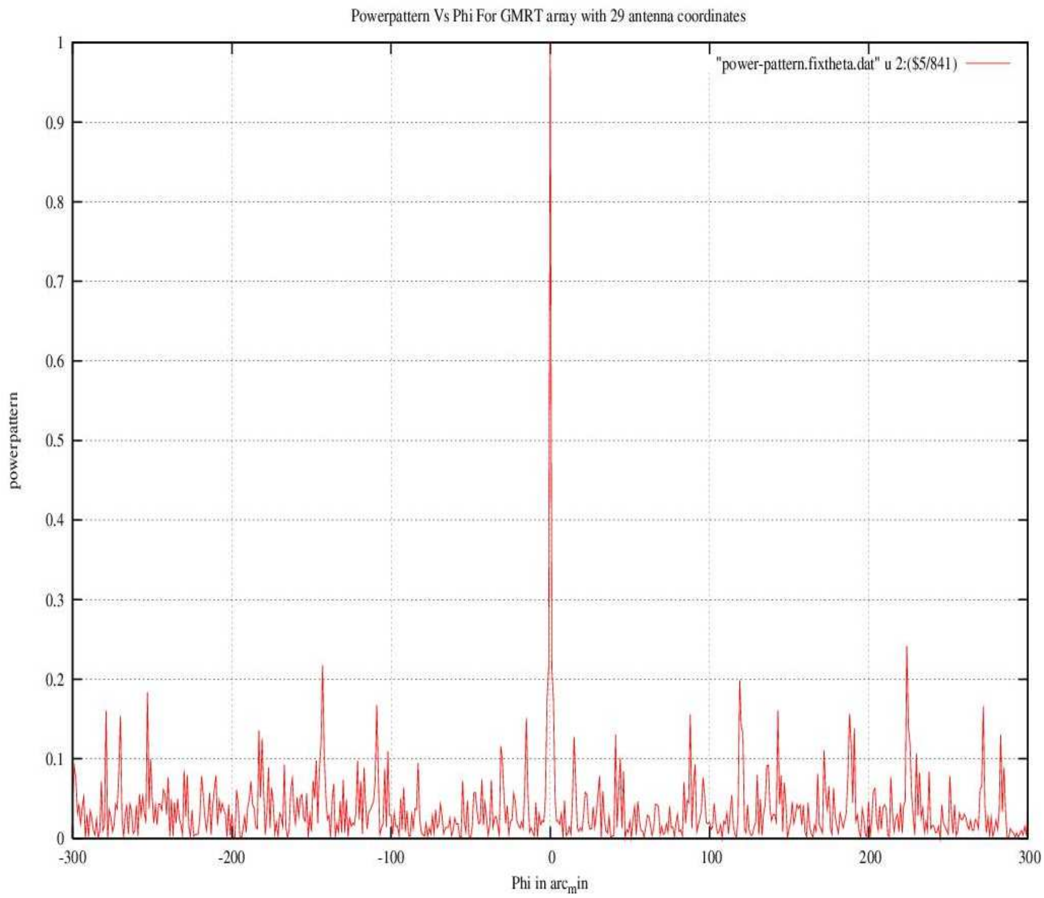


fig5.10

From the above plots it is noted that on increasing the phased antenna from the central square to the arm antenna there is not much change in the added noise levels which remains below 10%.

The Half Power Beam Width of the phased beams are as follows:

1) For 17 antenna coordinates : HPBW = 1.61'

Range of theta = 0.2 deg

Accuracy of theta = 0.05 '

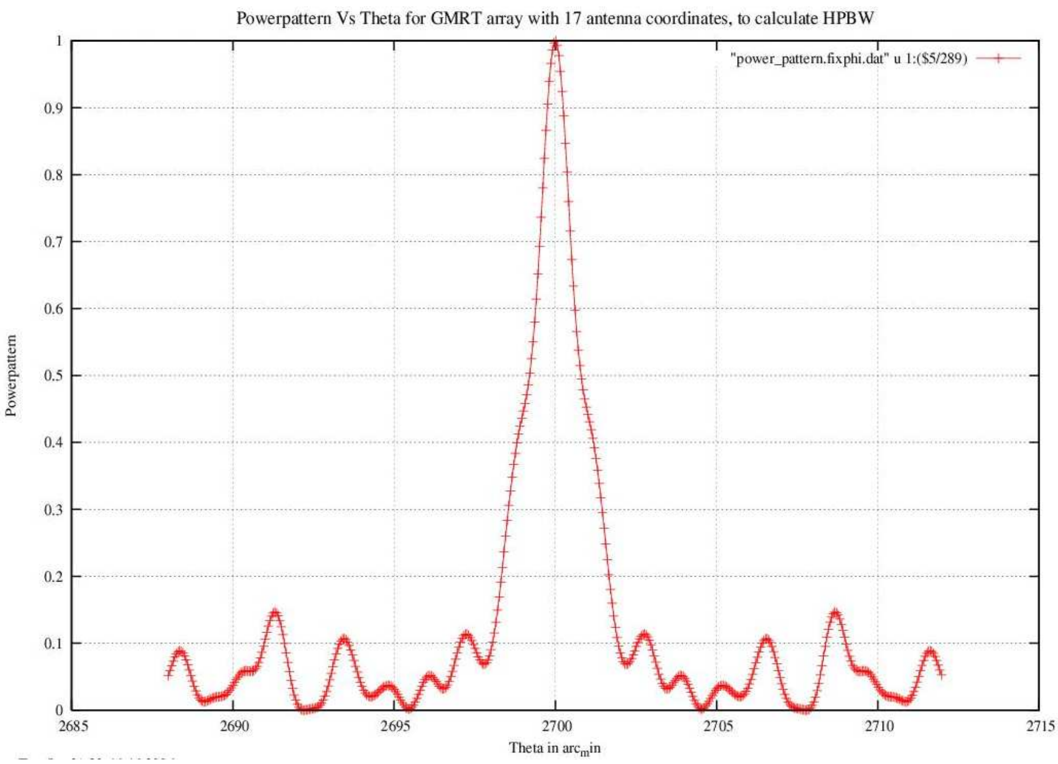


fig5.11

2) For 20 antenna coordinates : HPBW = 0.84 '

Range of theta = 0.2 deg

Accuracy of theta = 0.05 '

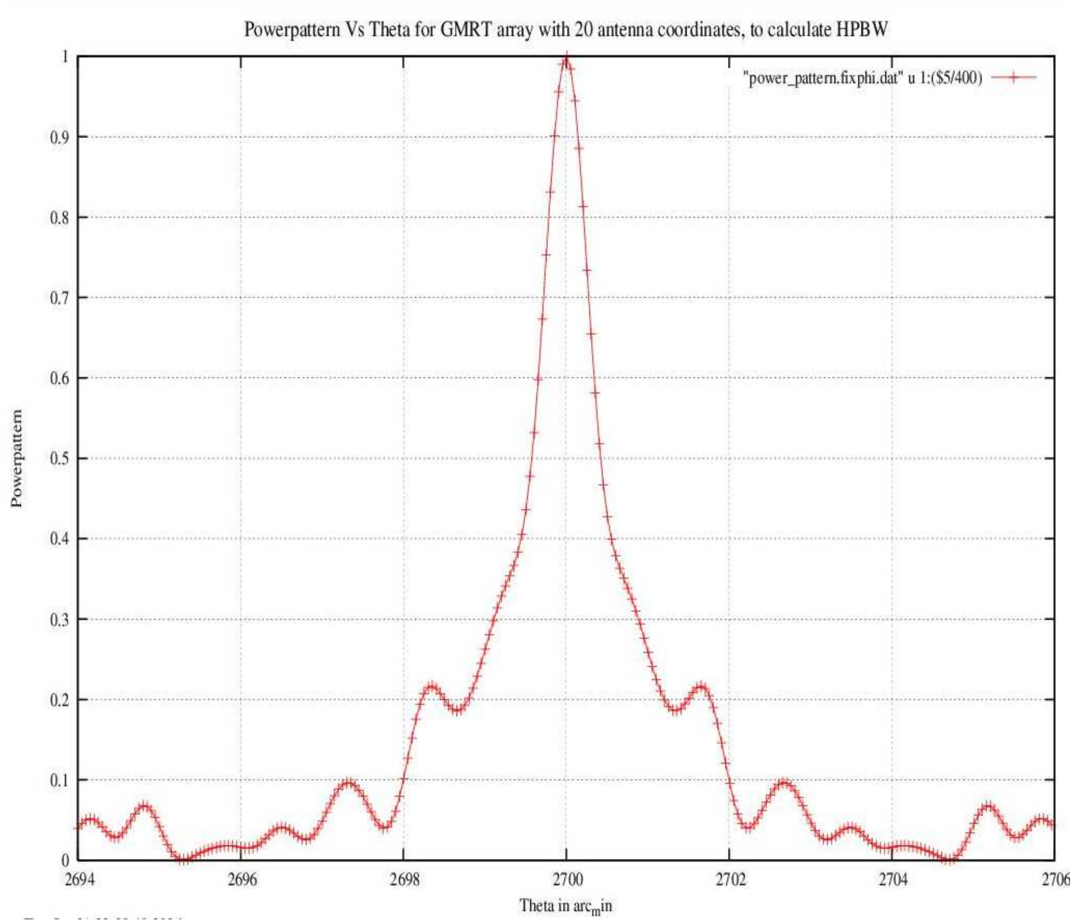


fig5.12

3) For 23 antenna coordinates : HPBW = 0.57 '

Range of theta = 0.1 deg

Accuracy of theta = 0 .05 '

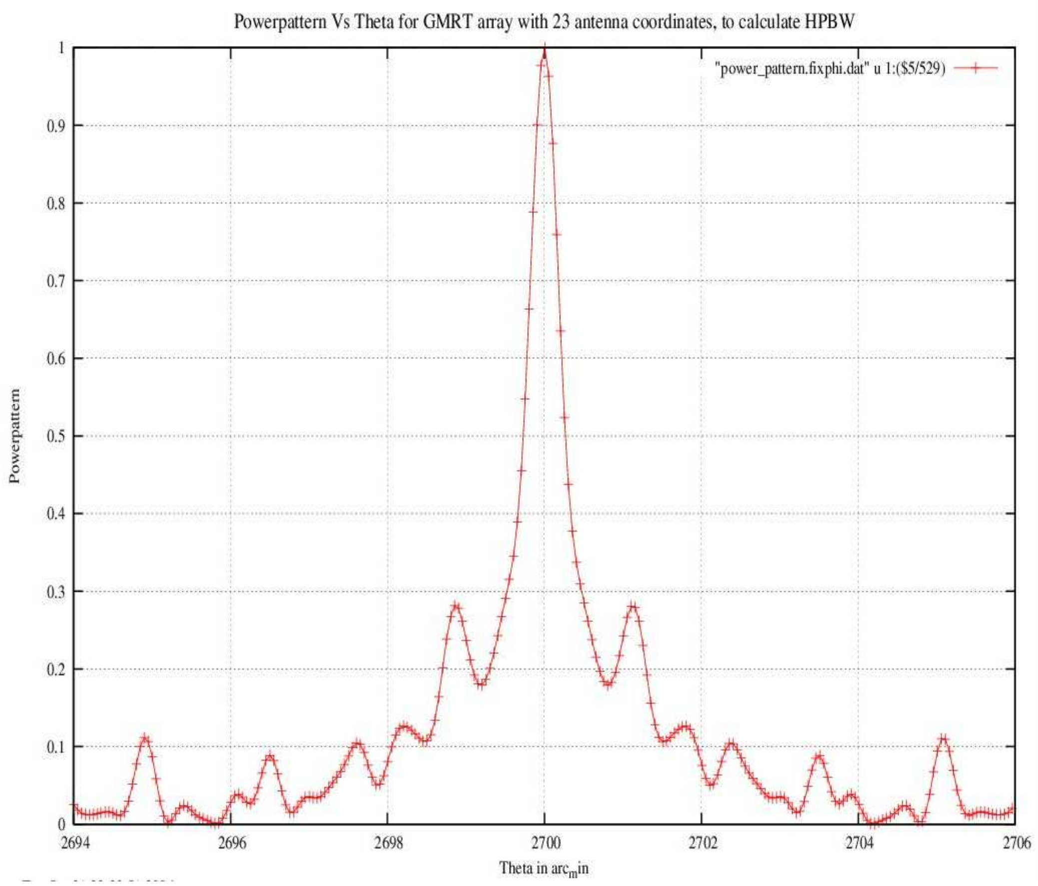


fig5.13

4) For 26 antenna coordinates : HPBW = 0.36 '

Range of theta = 0.05 deg

Accuracy of theta = 0.02 '

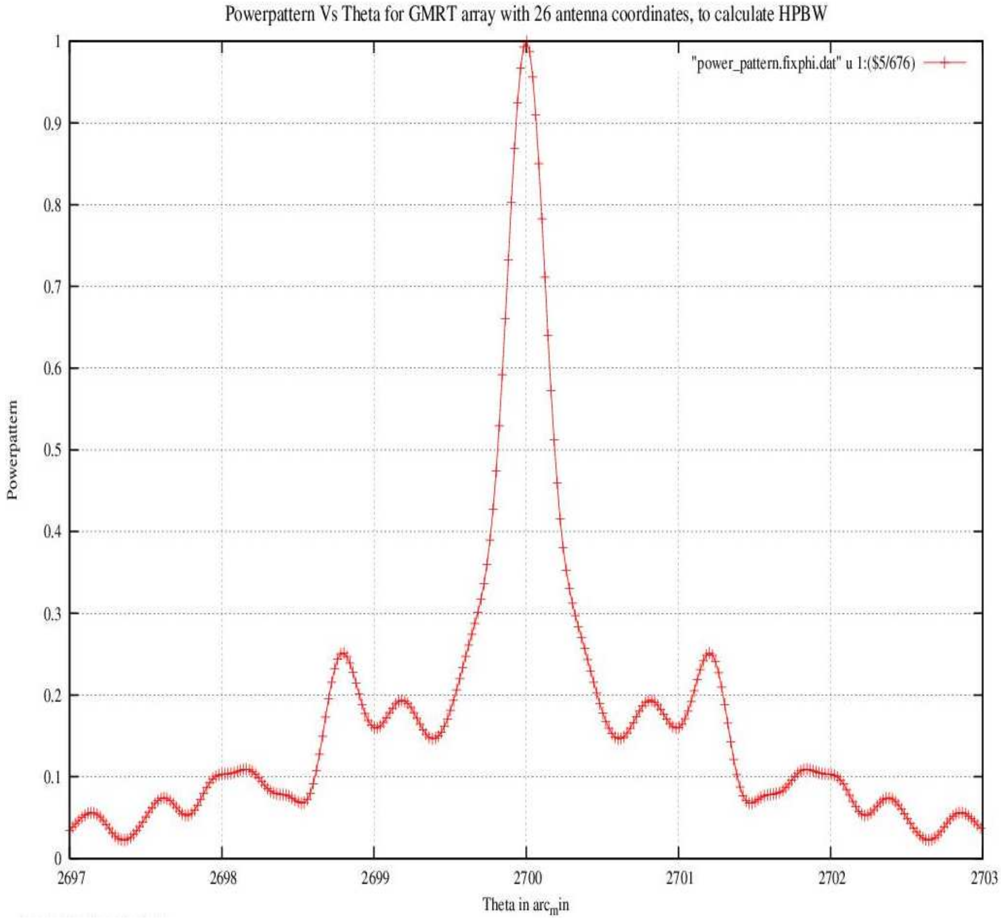


fig5.14

5) For 29 antenna coordinates : HPBW = 0.27 '

Range of theta = 0.02 deg

Accuracy of theta = 0.01 '

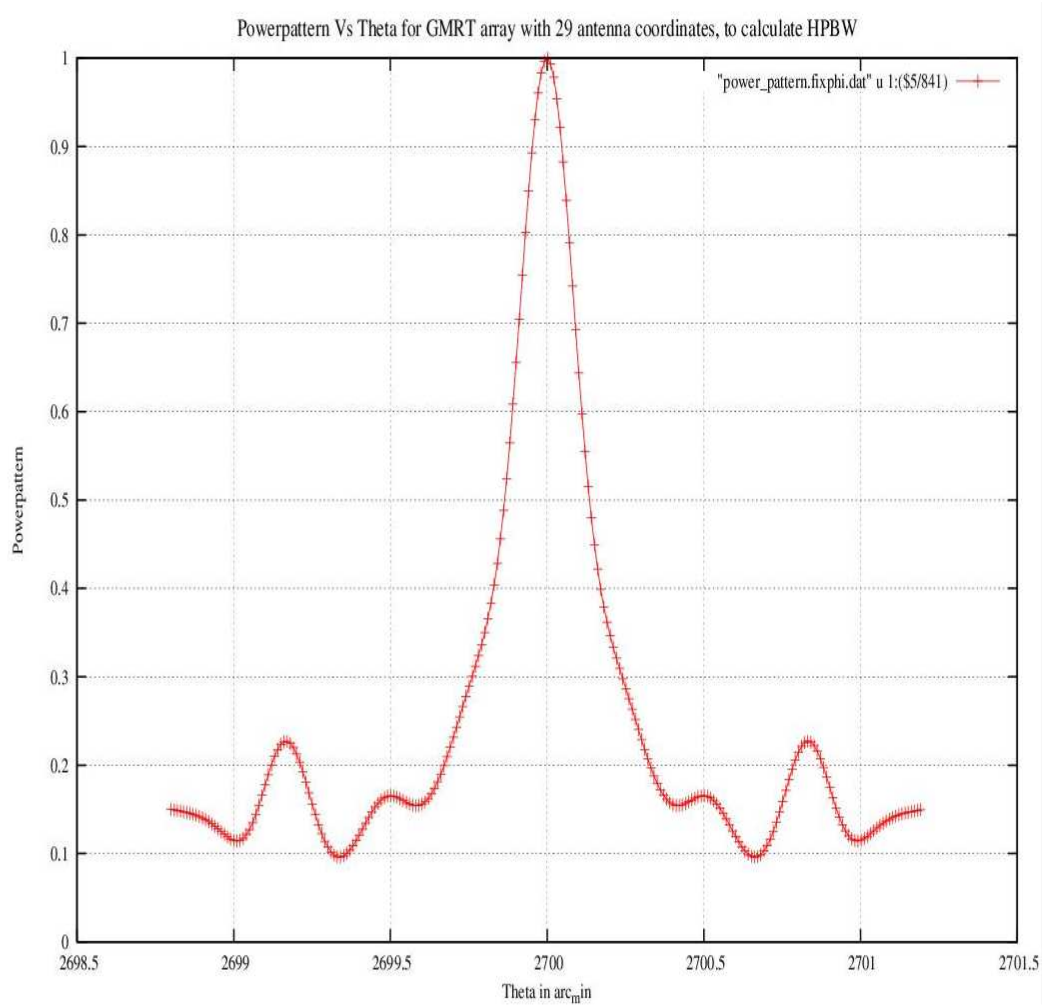


fig5.15

SCOPE OF IMPROVEMENT IN THE CODE

The final code of the program is successful in generating multiple beams in the 2-D sky for any direction of phasing. This program acts as a stand-alone in helping to view how array beam are phased for any linear or random distributed array of antenna.

1) The code can be improved to be made more user friendly in the task of entering the input data for calculation of the beam pattern. This can be done by writing a script file which will have the inputs given as arguments. Thus on running the script file each time the power gets calculated., and the user need not enter the same inputs time and again.

2) The equation for calculating the x component,y component,z component of the power pattern can be theoretically simplified . The incremental phase factors can be incorporated as a linear term over and above the existing phase factor. This makes the algorithm and the calculation of power factor much simpler.

3) The user may need to phase the beams finite beam width away from the central beam. The algorithm need to be modified accordingly

APPENDIX

LIST OF VARIABLES AND FLOWCHART

STRING VARIABLE:

fname[100] : to hold a variable file name

INTEGER VARIABLES:

n : Reference antenna co-ordinate

m : Number of antenna chosen

no_th : Number of beams to be generated along theta axis around phasing direction

no_phi : Number of beams to be generated along phi axis around phasing direction

s : Running index to demarcate the files where the generated multiple beam data is written

i : Running index for multiple use

j[30] : Array to hold the coordinates of the antenna selected in the sequence

cnt , count : Index used to obtain the antenna coordinates of the selected antenna

inp1 : Index for the array th_P when writing into the input data file

inp2 : Index for the array phi_P when writing into the input data file

inp3 : Index for the array th_P when reading into the input data file

inp4 : Index for the array phi_P when writing into the input data file

FLOAT VARIABLES

x[30] , y[30] , z[30] : Arrays to hold the coordinates of the antenna

x1[30] , y1[30] , z1[30] : Holds the antenna coordinates subtracted from the reference antenna coordinates

l : Wavelength

th_P0 : Phasing direction of the beam in theta

phi_P0 : Phasing direction of the beam in phi

th_P[20] : Array to hold multiple value of phasing theta

phi_P[20] : Array to hold multiple value of phasing phi

d_th_P : Separation between phasing beams in direction of theta

d_phi_P : Separation between phasing beams in direction of phi

th_0 : Direction in theta about which beam pattern is calculated

phi_0 : Direction in phi about which beam pattern is calculated

th : Running value of theta according to the loop condition

phi : Running value of phi according to the loop condition

r_th : Range of theta

a_th : Accuracy of theta

r_phi : Range of phi

a_phi : Accuracy of phi

f_th : Fixed theta value ,the phi variation is noted keeping this theta fixed

f_phi : Fixed phi value ,the theta variation is noted keeping this phi fixed

rms1 : RMS value of amplitude of noise

mean1 : Mean value of amplitude of noise

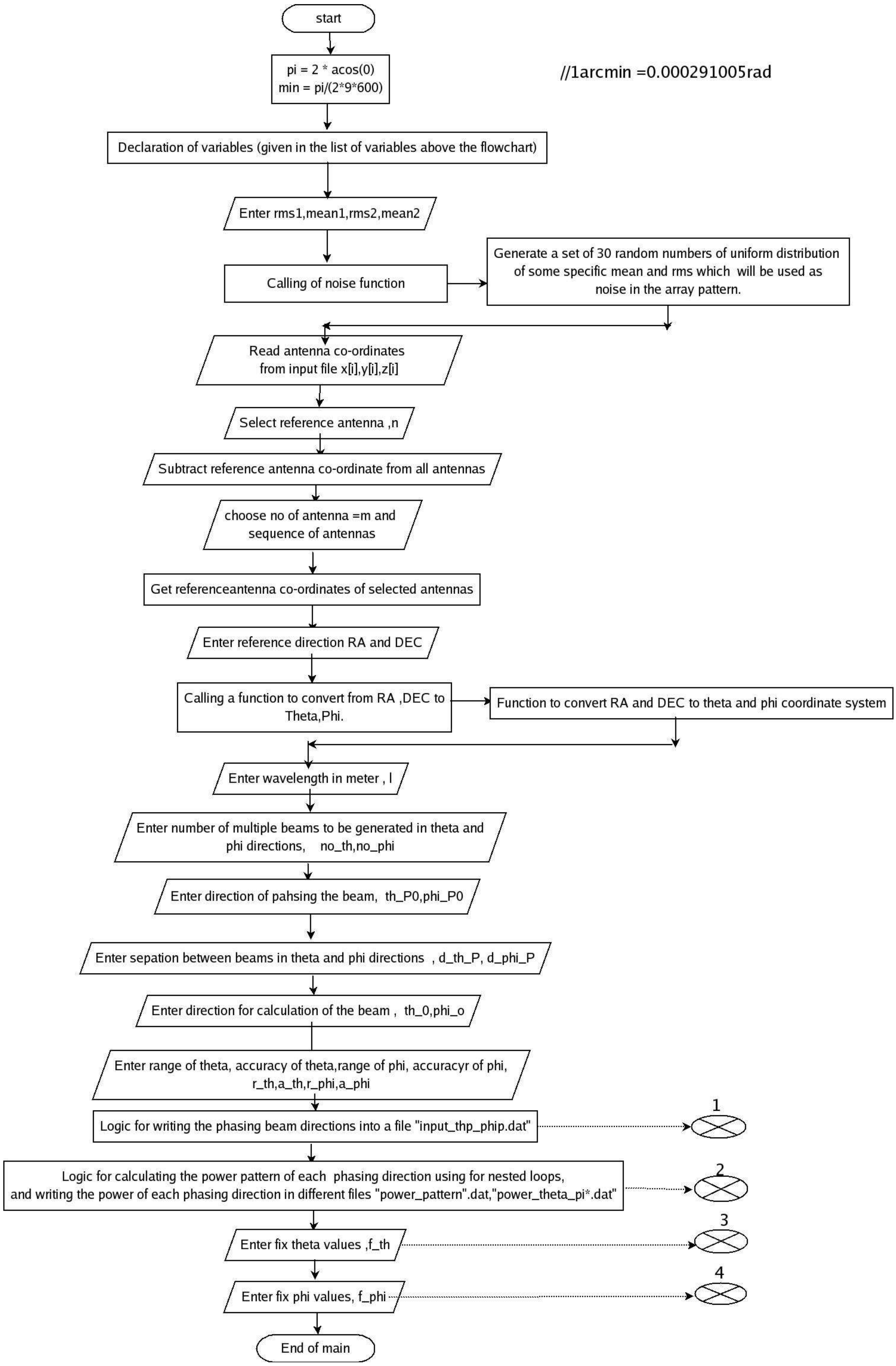
rms2 : RMS value of phase of noise

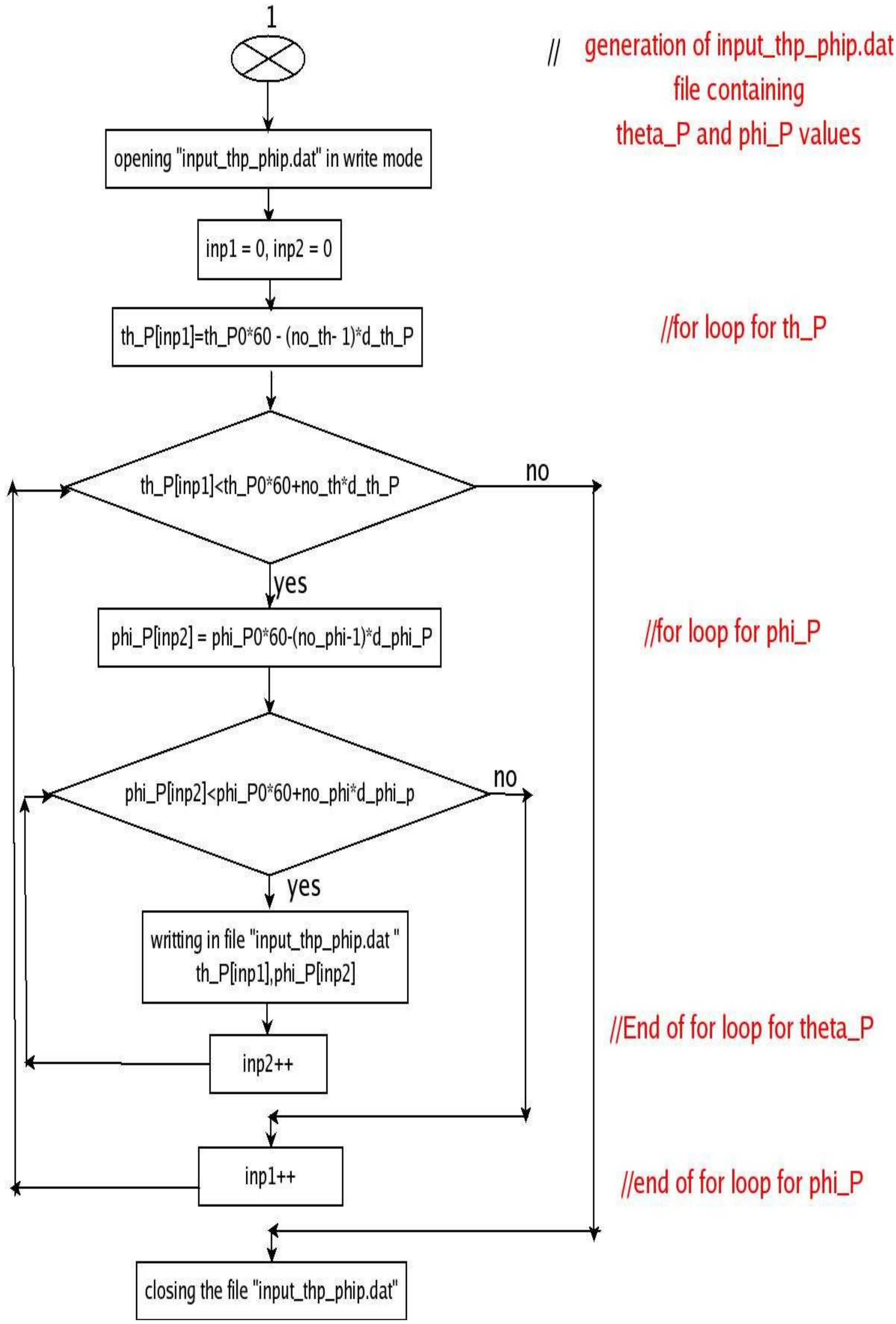
mean2 :Mean value of phase of noise

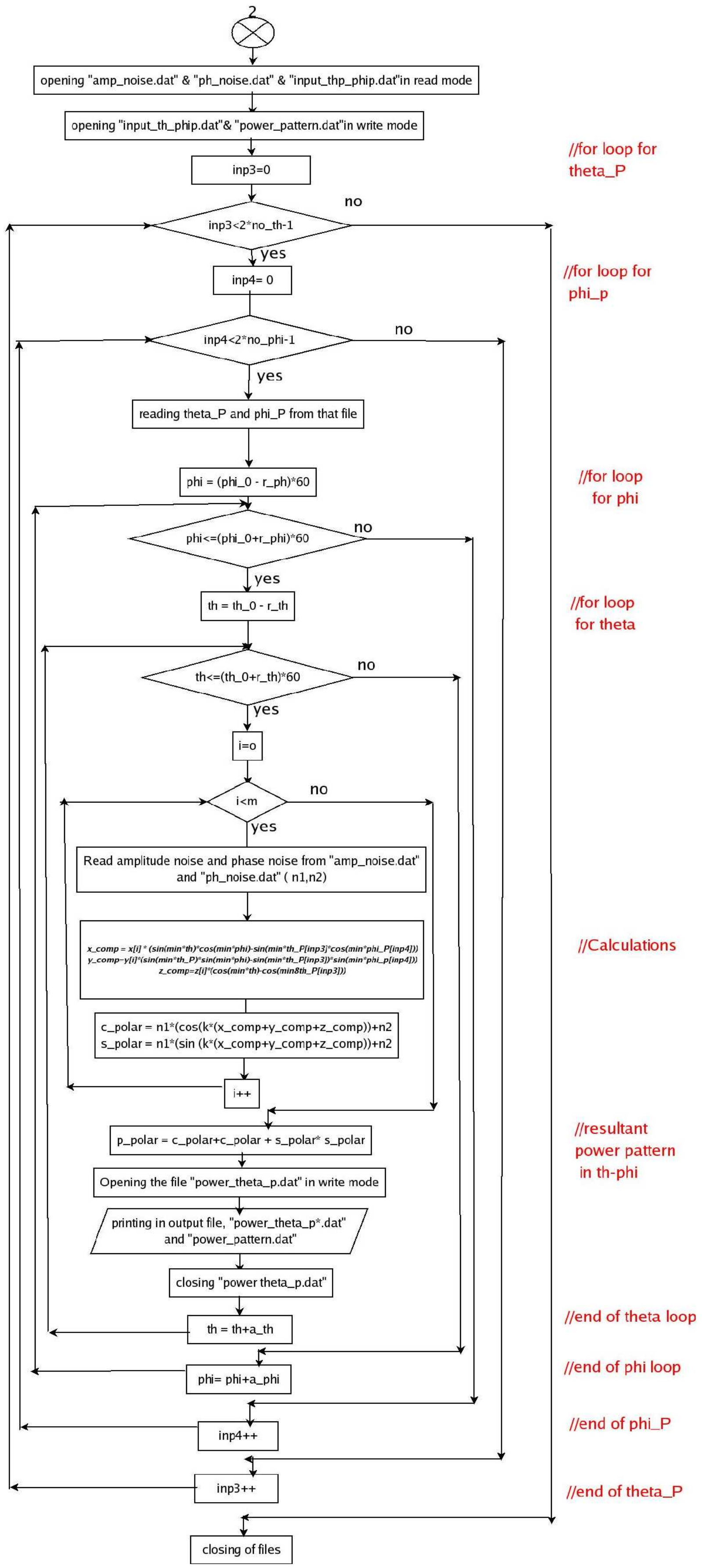
RA(RA_hr,RA_min,RA_sec) ,DEC(DEC_deg,DEC_min,DEC_sec) : Coordinates in the RA-DEC system

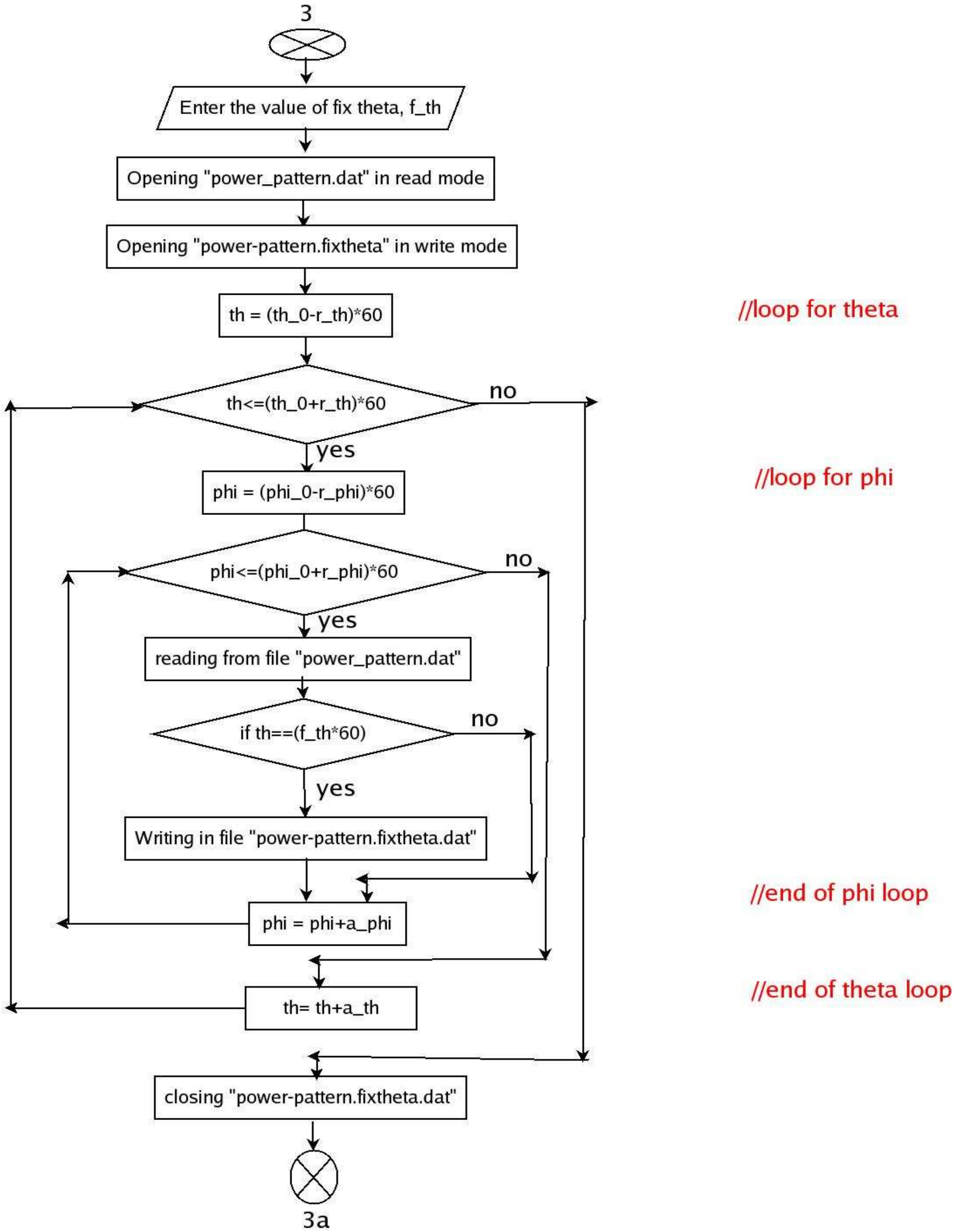
FLOWCHART:

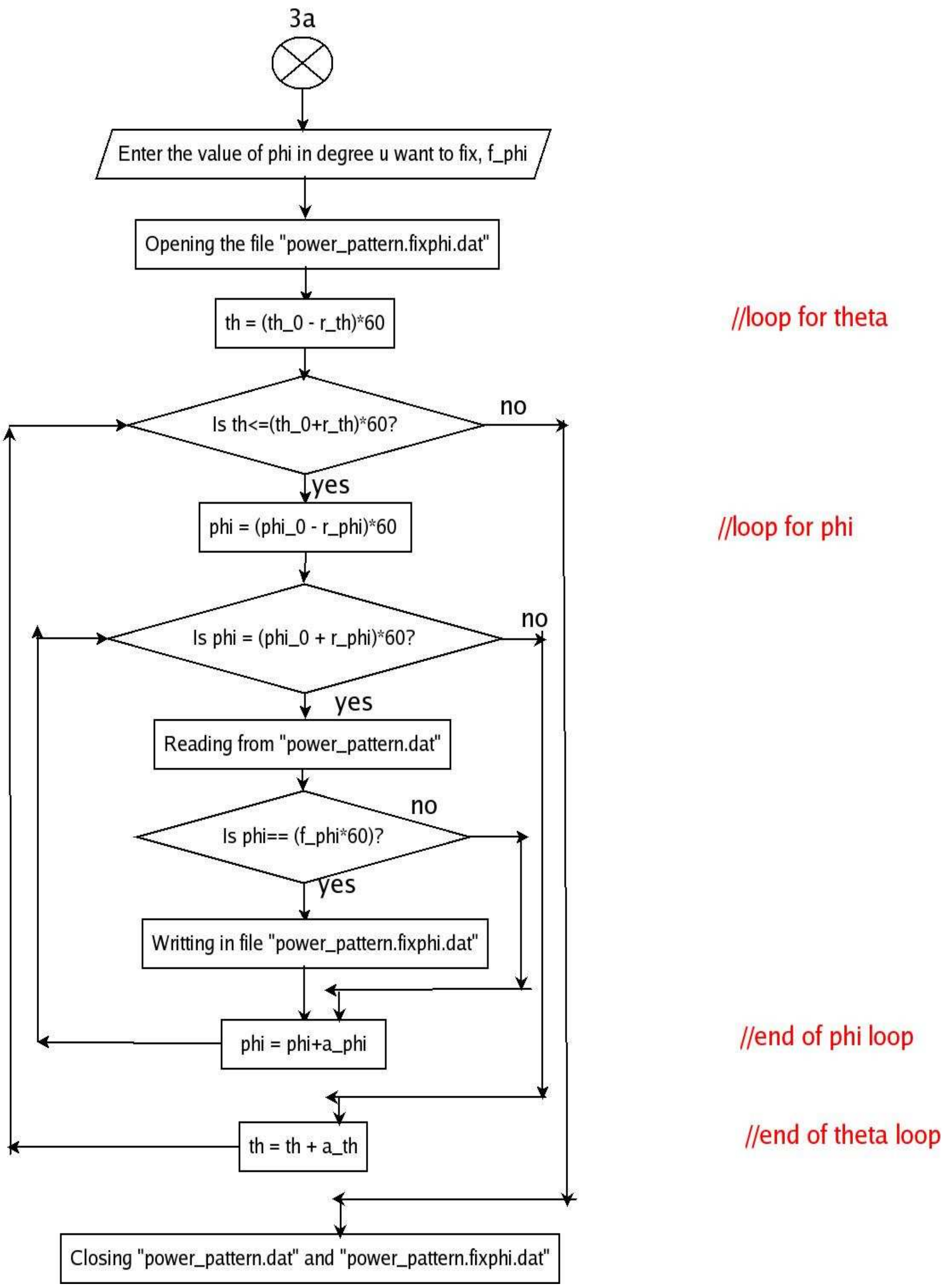
1) The flowchart of the program as an algorithm is given below:




1
// generation of input_thp_phip.dat file containing theta_P and phi_P values
opening "input_thp_phip.dat" in write mode
inp1 = 0, inp2 = 0
th_P[inp1]=th_P0*60 - (no_th- 1)*d_th_P
//for loop for th_P
th_P[inp1]<th_P0*60+no_th*d_th_P
no
yes
phi_P[inp2] = phi_P0*60-(no_phi-1)*d_phi_P
//for loop for phi_P
phi_P[inp2]<phi_P0*60+no_phi*d_phi_p
no
yes
writting in file "input_thp_phip.dat " th_P[inp1],phi_P[inp2]
//End of for loop for theta_P
inp2++
inp1++
//end of for loop for phi_P
closing the file "input_thp_phip.dat"


2
opening "amp_noise.dat" & "ph_noise.dat" & "input_thp_phip.dat"in read mode
opening "input_th_phip.dat"& "power_pattern.dat"in write mode
//for loop for theta_P
inp3=0
inp3<2*no_th-1
no
yes
inp4= 0
//for loop for phi_p
inp4<2*no_phi-1
no
yes
reading theta_P and phi_P from that file
phi = (phi_0 - r_ph)*60
//for loop for phi
phi<=(phi_0+r_phi)*60
no
yes
th = th_0 - r_th
//for loop for theta
th<=(th_0+r_th)*60
no
yes
i=o
i<m
no
yes
Read amplitude noise and phase noise from "amp_noise.dat" and "ph_noise.dat" (n1,n2)
x_comp = x[i] * (sin(min*th)*cos(min*phi)-sin(min*th_P[inp3]*cos(min*phi_P[inp4]))
y_comp-y[i]*(sin(min*th_P)*sin(min*phi)-sin(min*th_P[inp3])*sin(min*phi_p[inp4]))
z_comp=z[i]*(cos(min*th)-cos(min8th_P[inp3]))
//Calculations
c_polar = n1*(cos(k*(x_comp+y_comp+z_comp))+n2
s_polar = n1*(sin (k*(x_comp+y_comp+z_comp))+n2
i++
p_polar = c_polar+c_polar + s_polar* s_polar
//resultant power pattern in th-phi
Opening the file "power_theta_p.dat" in write mode
printing in output file, "power_theta_p*.dat" and "power_pattern.dat"
closing "power theta_p.dat"
th = th+a_th
//end of theta loop
phi= phi+a_phi
//end of phi loop
inp4++
//end of phi_P
inp3++
//end of theta_P
closing of files


3
Enter the value of fix theta, f_th
Opening "power_pattern.dat" in read mode
Opening "power-pattern.fixtheta" in write mode
th = (th_0-r_th)*60
//loop for theta
th<=(th_0+r_th)*60
no
yes
phi = (phi_0-r_phi)*60
//loop for phi
phi<=(phi_0+r_phi)*60
no
yes
reading from file "power_pattern.dat"
if th==(f_th*60)
no
yes
Writing in file "power-pattern.fixtheta.dat"
//end of phi loop
phi = phi+a_phi
//end of theta loop
th= th+a_th
closing "power-pattern.fixtheta.dat"
3a


3a
Enter the value of phi in degree u want to fix, f_phi
Opening the file "power_pattern.fixphi.dat"
th = (th_0 - r_th)*60
//loop for theta
Is th<=(th_0+r_th)*60?
no
yes
phi = (phi_0 - r_phi)*60
//loop for phi
Is phi = (phi_0 + r_phi)*60?
no
yes
Reading from "power_pattern.dat"
Is phi== (f_phi*60)?
no
yes
Writting in file "power_pattern.fixphi.dat"
phi = phi+a_phi
//end of phi loop
th = th + a_th
//end of theta loop
Closing "power_pattern.dat" and "power_pattern.fixphi.dat"

gmrt_arraypattern.c

COMMENTS::

1.Endfire did not work.

2.Beam phasing in one direction and calculation in another direction is not possible.

3.Multiple beam generation is not possible here.

STEPS TO ACHIEVE THE FINAL CODE::

1.Multiple beam generation concept brought in.

2.Initially in three specific directions of theta the beam is phased which is supplied by the programmer and not by the user. The power pattern is written in three individual files and this file generation is not flexible.

PROGRAM NAME :: arraypattern_version1.c

Printed by Somprova Nandy


```c
/*COMMENTS::     1.PHASING OF BEAM FOR ENDFIRE CASE IS POSSIBLE HERE.
                 2.CONCEPT OF PHASING THE BEAM AND CALCULATION OF THE PATTERN IN
ANY DIRECTION BROUGHT IN.

/* Make a program which will calculate the phased array beam in the presence of
noise */


#include<stdio.h>
#include <math.h>

#define pi  2*acos(0)
#define min  pi/(2*9*600) //1 arc-minute = 0.000291005 rad

int RA_DEC_to_th_phi(float RA_hr,float RA_min,float RA_sec,float DEC_deg,float D
EC_min,float DEC_sec);
int noise(float rms1,float rms2,float mean1,float mean2);

int main()
{
 int n,m;
 float x[30],y[30],z[30],l,th_P,phi_P,th_0,phi_0,r_th,a_th,r_phi,a_phi,f_th,f_ph
i,rms1,mean1,rms2,mean2,RA_hr,RA_min,RA_sec,DEC_deg,DEC_min,DEC_sec;

 system("clear");
 printf("enter the rms-s and means you want for uniformly distributed 30 random numbers will be used as noise t
o the beam pattern corresponding to amplitude and phase respectively");
 scanf("%f %f %f %f",&rms1,&mean1,&rms2,&mean2);
 noise(rms1,rms2,mean1,mean2);

 // Read the antenna co-ordinates from the input file
 FILE *ifp = NULL;
 int i;
 float x1[30], y1[30], z1[30];

 ifp = fopen("input.dat","r"); //input.dat is the input file for antenna co-ordinat
es in x,y,z plane.
 for(i=0;i<30;i++)
        fscanf(ifp,"%f %f %f\n",&x[i],&y[i],&z[i]);
 fclose(ifp);

 // Select the reference antenna co-ordinate from the list of the antennas
 printf("choose the antenna number you want as referrence n = ");
 scanf("%d",&n);
 printf("the antenna co-ordinates you select as reference are\n");
 printf("%5.2f    %5.2f    %5.2f\n",x[n-1],y[n-1],z[n-1]);

 // Subtract the ref antenna co-ordinate from the all antennas
 printf("\nthe corrected co-ordinates in meters are:\n");
 for(i=0;i<30;i++)
 { x1[i] = x[i] - x[n-1];
   y1[i] = y[i] - y[n-1];
   z1[i] = z[i] - z[n-1];
   printf("%5.2f    %5.2f    %5.2f\n",x1[i],y1[i],z1[i]);
 }

 // Get the No of antenna as well as the antenna nos
 printf("choose the number (must be <= 30) of antennas m = ");
 scanf("%i",&m);
```

```
int j[30];
printf("\n choose the sequence of antennas. You must input the reference antenna. \n");
for(i=0;i<m;i++)
{
 scanf("%i",&j[i]);
 printf("%i\n",j[i]);
}

// Get the antenna co-ordinates of the selected antennas
int cnt;
int count = 0;
printf("the corresponding antenna co-ordinates are\n");
for(i=0;i<30;i++)
{
 for(cnt=0;cnt<m;cnt++)
 {      if(i == j[cnt])
        {       printf("%5.2f %5.2f %5.2f\n",x1[i],y1[i],z1[i]);
                x[count] = x1[i]; y[count] = y1[i]; z[count] = z1[i];
                count++;
        }
 }
}
printf("enter the reference direction (in sky) in RA(RA_hr,RA_min,RA_sec) and DEC(DEC_deg,DEC_min,DEC
_sec) co-ordinate system");
scanf("%f %f %f %f %f %f",&RA_hr,&RA_min,&RA_sec,&DEC_deg,&DEC_min,&DEC_sec);
RA_DEC_to_th_phi(RA_hr,RA_min,RA_sec,DEC_deg,DEC_min,DEC_sec);

printf("\nenter the wavelength l in meter = ");
scanf("%f",&l);

printf("enter Theta_P and Phi_P which specify the direction for phasing the array ");
scanf("%f %f",&th_P,&phi_P);

printf("enter Theta_0 and Phi_0 which specify the direction for calculation of pattern ");
scanf("%f %f",&th_0,&phi_0);
printf("\nenter the range you want in degree and the accuracy you want in arc-minute in the direction of theta or
polar angle \n");
scanf("%f %f",&r_th,&a_th);
printf("\nenter the range you want in degree and the accuracy you want in arc-minute in the direction of phi or az
imath \n");
scanf("%f %f",&r_phi,&a_phi);



FILE *ifp2;
FILE *ifp3;
FILE *ofp;

float x_comp,y_comp,z_comp,p_polar,p_RA_DEC,th,phi,RA,DEC,RA_0,DEC_0,k,c_polar,
s_polar,c_RA_DEC,s_RA_DEC,n1,n2;
ifp2 = fopen("amp_noise.dat","r");
ifp3 = fopen("ph_noise.dat","r");
ofp = fopen("power_pattern.dat","w");
fprintf (ofp,"#theta phi RA DEC power(th,phi) power(RA,DEC)\n");
k = 2*pi/l;
for(th=(th_0 - r_th)*60;th<=(th_0 + r_th)*60;th=(th + a_th))//so th and py are
in arc-minutes
{
  for(phi=(phi_0 - r_phi)*60;phi<=(phi_0 + r_phi)*60;phi=(phi + a_phi))//1 degr
ee = 60 arc-minute
```

```
    {
        RA = 4 * phi; // RA is expressed in Second since 1 arc-minute = 4 sec
        DEC = (90*60) - th; //DEC is expressed in arc-minute
        RA_0 = 4 * phi_0;//Reference RA direction
        DEC_0 = (90*60) - th_0;//Reference DEC direction
        c_polar = 0;
        s_polar = 0;
        c_RA_DEC = 0;
        s_RA_DEC = 0;
        for(i=0;i<m;i++)
        {
          fscanf(ifp2,"%f",&n1);
          fscanf(ifp3,"%f",&n2);

          x_comp=x[i]*(sin(min*th)*cos(min*phi)-sin(min*th_P*60)*cos(min*phi_P*6
0));
          y_comp=y[i]*(sin(min*th)*sin(min*phi)-sin(min*th_P*60)*sin(min*phi_P*6
0));
          z_comp=z[i]*(cos(min*th)-cos(min*th_P*60));

          c_polar += n1*(cos(k*( x_comp+ y_comp+ z_comp)) + n2);//the external n
 for amp_noise and internal for ph_noise. Here both are taken from an uniform ra
ndom distribution. But for more sophisticated version we can add different distr
ibution, in principle they should be also different for phase and amplitude.
          s_polar += n1*(sin(k*( x_comp+ y_comp+ z_comp)) + n2);//sin in polar c
o-ordinate

          c_RA_DEC += n1*(cos(k*(x[i]*(sin(min*DEC)*cos(min*phi)-sin(min*DEC_0)*
cos(min*phi_0))+y[i]*(sin(min*DEC)*sin(min*phi)-sin(min*DEC_0)*sin(min*phi_0))+z
[i]*(cos(min*DEC)-cos(min*DEC_0)))) + n2);//cosine in RA DEC co-ordinate system

          s_RA_DEC += n1*(sin(k*(x[i]*(sin(min*DEC)*cos(min*phi)-sin(min*DEC_0)*
cos(min*phi_0))+y[i]*(sin(min*DEC)*sin(min*phi)-sin(min*DEC_0)*sin(min*phi_0))+z
[i]*(cos(min*DEC)-cos(min*DEC_0)))) + n2);
        }

        p_polar = (c_polar * c_polar)+(s_polar * s_polar);//resultant power patt
ern in th-phi
        p_RA_DEC = (c_RA_DEC * c_RA_DEC) +(s_RA_DEC * s_RA_DEC);//resultant powe
r pattern in RA-DEC

        fprintf (ofp,"%f %f  %f  %f  %f  %f\n",th,phi,RA,DEC,p_polar,p_RA_DEC);
    }
    fprintf (ofp,"\n");

 }
 fclose(ifp2);
 fclose(ifp3);
 fclose(ofp);

 printf("enter the value of thita in degree u want to fix, fth =");
 scanf("%f",&f_th);

 FILE *ifp_6;
 FILE *ofp_7;
 char str[100];
 ifp_6 = fopen("power_pattern.dat","r");
 ofp_7 = fopen("power-pattern.fixtheta.dat","w");
 fgets(str,100,ifp_6);
```

```
 fprintf (ofp_7,"#fixed–theta phi RA DEC power(th,phi) power(RA,DEC)\n");
 for(th=(th_0-r_th)*60;th<=(th_0+r_th)*60;th=(th+a_th))//so th and py are in arc
-minutes
 {
  for(phi=(phi_0-r_phi)*60;phi<=(phi_0+r_phi)*60;phi=(phi+a_phi))//1 degree = 60
 arc-minute
  {
   fscanf(ifp_6,"%f %f %f %f %f %f",&th,&phi,&RA,&DEC,&p_polar,&p_RA_DEC);
   if(th == (f_th*60))
   {
    fprintf(ofp_7,"%f   %f   %f   %f   %f   %f\n",(f_th*60),phi,RA,DEC,p_polar,p_RA_DE
C);
    printf("%f   %f   %f   %f   %f   %f\n",(f_th*60),phi,RA,DEC,p_polar,p_RA_DEC);

   }
  }
 }
 fclose(ofp_7);

 printf("enter the value of phi in degree u want to fix,fphi =");
 scanf("%f",&f_phi);

 FILE *ofp_8;

 ofp_8 = fopen("power_pattern.fixphi.dat","w");
 rewind(ifp_6);
 fgets(str,100,ifp_6);
 fprintf (ofp_7,"#theta fixed–phi RA DEC power(th,phi) power(RA,DEC)\n");
  for(th=(th_0-r_th)*60;th<=(th_0+r_th)*60;th=(th+a_th))//so th and py are in ar
c-minutes
 {
  for(phi=(phi_0-r_phi)*60;phi<=(phi_0+r_phi)*60;phi=(phi+a_phi))//1 degree = 60
 arc-minute
  {
   fscanf(ifp_6,"%f %f %f %f %f %f",&th,&phi,&RA,&DEC,&p_polar,&p_RA_DEC);
   if(phi == (f_phi*60))
   {
    fprintf(ofp_8,"%f   %f   %f   %f   %f   %f\n",th,(f_phi*60),RA,DEC,p_polar,p_RA_DE
C);
    printf("%f   %f   %f   %f   %f\n",th,(f_phi*60),RA,DEC,p_polar,p_RA_DEC);
   }
  }
 }
 fclose(ifp_6);
 fclose(ofp_8);

 return(0);
}


/* Generate a set of 30 random number of uniform distribution of some specific m
ean and rms which will be used as noise in the array pattern*/

int noise(float rms1,float rms2, float mean1,float mean2)
{
 FILE *ofp1 = NULL;
 FILE *ofp2 = NULL;
 ofp1 = fopen("amp_noise.dat","w");
 ofp2 = fopen("ph_noise.dat","w");
```

```
 int i;
 float a,b;

 for(i=0;i<30;i++)
 {
  a = rms1 * rand() + mean1;
  b = rms2 * rand() + mean2;

  fprintf(ofp1,"%f\n",a);
  fprintf(ofp2,"%f\n",b);
  printf("%d  %f  %f\n",i,a,b);
 }

 fclose(ofp1);
 fclose(ofp2);
}


/* make a program to convert the RA & DEC co-ordinate system in to theta (elivat
ion) and phy (azimath) co-ordinate system */

int RA_DEC_to_th_phi(float RA_hr,float RA_min,float RA_sec,float DEC_deg,float D
EC_min,float DEC_sec)
{
 float th_0, phi_0;
 phi_0 = (RA_sec * 15)/(60*60) + (RA_min * 15)/(60) + (RA_hr * 15);//phi calcula
ted in degree : 1 hr = 15 deg, 1 min = 15 arc-min, 1 sec = 15 arc-sec
 th_0 = 90 - ((DEC_deg) + (DEC_min/60) + (DEC_sec/(60*60)));// th in Degree

 printf("Theta_0 = %f\n",th_0);
 printf("Phi_0 = %f\n",phi_0);

 return(0);
}
```

PROGRAM NAME:: arraypattern_version2.c

Printed by Somprova Nandy


```
/*COMMENTS    : 1.  MULTIPLE BEAM GENERATION IN THETA DIRECTION CONCEPT BROUGHT
IN
               2.  THREE FIXED VALUES OF THETA_P SET IN THE PROGRAM, AND IF-ELS
E LOOP WRITES DATA INTO 3 INDIVIDUAL FILES.     */

#include<stdio.h>
#include <math.h>

#define pi  2*acos(0)
#define min  pi/(2*9*600) //1 arc-minute = 0.000291005 rad



int RA_DEC_to_th_phi(float RA_hr,float RA_min,float RA_sec,float DEC_deg,float D
EC_min,float DEC_sec);
int noise(float rms1,float rms2,float mean1,float mean2);

int main()
{
 int n,m;
 float x[30],y[30],z[30],l,th_P,phi_P,th_0,phi_0,r_th,a_th,r_phi,a_phi,f_th,f_ph
i,rms1,mean1,rms2,mean2,RA_hr,RA_min,RA_sec,DEC_deg,DEC_min,DEC_sec;
 system("clear");
 printf("enter the rms-s and means you want for uniformly distributed 30 random numbers will be used as noise t
o the beam pattern corresponding to amplitude and phase respectively ");
 scanf("%f %f %f %f",&rms1,&mean1,&rms2,&mean2);
 noise(rms1,rms2,mean1,mean2);

 // Read the antenna co-ordinates from the input file
 FILE *ifp = NULL;
 int i;
 float x1[30], y1[30], z1[30];

 ifp = fopen("input.dat","r"); //input.dat is the input file for antenna co-ordinat
es in x,y,z plane.
 for(i=0;i<30;i++)
        fscanf(ifp,"%f %f %f\n",&x[i],&y[i],&z[i]);
 fclose(ifp);

 // Select the reference antenna co-ordinate from the list of the antennas
 printf("choose the antenna number you want as referrence n = ");
 scanf("%d",&n);
 printf("the antenna co-ordinates you select as reference are\n");
 printf("%5.2f    %5.2f    %5.2f\n",x[n-1],y[n-1],z[n-1]);

 // Subtract the ref antenna co-ordinate from the all antennas
 printf("\nthe corrected co-ordinates in meters are:\n");
 for(i=0;i<30;i++)
 { x1[i] = x[i] - x[n-1];
   y1[i] = y[i] - y[n-1];
   z1[i] = z[i] - z[n-1];
   printf("%5.2f    %5.2f    %5.2f\n",x1[i],y1[i],z1[i]);
 }

 // Get the No of antenna as well as the antenna nos
 printf("choose the number (must be <= 30) of antennas m = ");
 scanf("%i",&m);
 int j[30];
 printf("\n choose the sequence of antennas. You must input the reference antenna. \n");
```

```
 for(i=0;i<m;i++)
 {
  scanf("%i",&j[i]);
  printf("%i\n",j[i]);
 }

 // Get the antenna co-ordinates of the selected antennas
 int cnt;
 int count = 0;
 printf("the corresponding antenna co-ordinates are\n");
 for(i=0;i<30;i++)
 {
  for(cnt=0;cnt<m;cnt++)
  {      if(i == j[cnt])
         {        printf("%5.2f %5.2f %5.2f\n",x1[i],y1[i],z1[i]);
                  x[count] = x1[i]; y[count] = y1[i]; z[count] = z1[i];
                  count++;
         }
  }
 }
 printf("enter the reference direction (in sky) in RA(RA_hr,RA_min,RA_sec) and DEC(DEC_deg,DEC_min,DEC
_sec) co-ordinate system");
 scanf("%f %f %f %f %f %f",&RA_hr,&RA_min,&RA_sec,&DEC_deg,&DEC_min,&DEC_sec);
 RA_DEC_to_th_phi(RA_hr,RA_min,RA_sec,DEC_deg,DEC_min,DEC_sec);

 printf("\nenter the wavelength l in meter = ");
 scanf("%f",&l);
 printf("enter  Phi_P which specify the direction for phasing the array ");
 scanf("%f",&phi_P);
 printf("enter Theta_0 and Phi_0 which specify the direction for calculation of pattern ");
 scanf("%f %f",&th_0,&phi_0);
 printf("\nenter the range you want in degree and the accuracy you want in arc-minute in the direction of theta or
polar angle \n");
 scanf("%f %f",&r_th,&a_th);
 printf("\nenter the range you want in degree and the accuracy you want in arc-minute in the direction of phi or az
imath \n");
 scanf("%f %f",&r_phi,&a_phi);



 FILE *ifp2;
 FILE *ifp3;
 FILE *ofp;
 FILE *ofp1;
 FILE *ofp11;
 FILE *ofp111;

 float x_comp,y_comp,z_comp,p_polar,p_RA_DEC,th,phi,RA,DEC,RA_0,DEC_0,k,c_polar,
s_polar,c_RA_DEC,s_RA_DEC,n1,n2;
 ifp2 = fopen("amp_noise.dat","r");
 ifp3 = fopen("ph_noise.dat","r");
 ofp = fopen("power_pattern.dat","w");
 ofp1 =fopen("power_theta_p1.dat","w");
 ofp11=fopen("power_theta_p11.dat","w");
 ofp111=fopen("power_theta_p111.dat","w");

 fprintf (ofp,"#theta theta_P phi RA DEC power(th,phi) power(RA,DEC) \n");
 fprintf (ofp1,"#phi        phi_p       theta_P      theta  power(th,phi)\n");
 fprintf (ofp11,"#phi  phi_p  theta_P theta  power(th,phi)\n");
 fprintf (ofp111,"#phi  phi_p  theta_P theta  power(th,phi)\n");
```

```
 k = 2*pi/l;

for(th_P=0;th_P<=1;th_P+=.5)
{

 for(phi=(phi_0 - r_phi)*60;phi<=(phi_0 + r_phi)*60;phi=(phi + a_phi))//1 degree
 = 60 arc-minute
  {

  for(th=(th_0 - r_th)*60;th<=(th_0 + r_th)*60;th=(th + a_th))//so th and py are
 in arc-minutes
    {

       RA = 4 * phi; // RA is expressed in Second since 1 arc-minute = 4 sec
       DEC = (90*60) - th; //DEC is expressed in arc-minute
       RA_0 = 4 * phi_0;//Reference RA direction
       DEC_0 = (90*60) - th_0;//Reference DEC direction
       c_polar = 0;
       s_polar = 0;
       c_RA_DEC = 0;
       s_RA_DEC = 0;
       for(i=0;i<m;i++)
       {
         fscanf(ifp2,"%f",&n1);
         fscanf(ifp3,"%f",&n2);
         x_comp=x[i]*(sin(min*th)*cos(min*phi)-sin(min*th_P*60)*cos(min*phi_P*6
0));
         y_comp=y[i]*(sin(min*th)*sin(min*phi)-sin(min*th_P*60)*sin(min*phi_P*6
0));
         z_comp=z[i]*(cos(min*th)-cos(min*th_P*60));

         c_polar += n1*(cos(k*(x_comp+y_comp+z_comp)) + n2);//the external n fo
r amp_noise and internal for ph_noise. Here both are taken from an uniform rando
m distribution. But for more sophisticated version we can add different distribu
tion, in principle they should be also different for phase and amplitude.

         s_polar += n1*(sin(k*(x_comp+y_comp+z_comp)) + n2);//sin in polar co-o
rdinate

         c_RA_DEC += n1*(cos(k*(x[i]*(sin(min*DEC)*cos(min*phi)-sin(min*DEC_0)*
cos(min*phi_0))+y[i]*(sin(min*DEC)*sin(min*phi)-sin(min*DEC_0)*sin(min*phi_0))+z
[i]*(cos(min*DEC)-cos(min*DEC_0)))) + n2);//cosine in RA DEC co-ordinate system

         s_RA_DEC += n1*(sin(k*(x[i]*(sin(min*DEC)*cos(min*phi)-sin(min*DEC_0)*
cos(min*phi_0))+y[i]*(sin(min*DEC)*sin(min*phi)-sin(min*DEC_0)*sin(min*phi_0))+z
[i]*(cos(min*DEC)-cos(min*DEC_0)))) + n2);
       }

       p_polar = (c_polar * c_polar)+(s_polar * s_polar);//resultant power patt
ern in th-phi
       p_RA_DEC = (c_RA_DEC * c_RA_DEC) +(s_RA_DEC * s_RA_DEC);//resultant powe
r pattern in RA-DEC



               if(th_P==0)
               {
               fprintf (ofp1,"%f %f    %f %f   %f\n",phi,phi_P,th_P,th, p_polar
 );
```

```
                printf ("IF lOOP1 %f  %f  %f  %f  %f\n",phi,phi_P,th_P,th,p_polar);
    // WRITING DATA INTO 3 INDIVIDUAL FILES ,FOR EACH PHASING ANGLE
                }

                else if (th_P==.5)
                {
                fprintf (ofp11,"%f  %f %f  %f %f \n",phi,phi_P,th_P,th, p_polar );
                printf ("IF lOOP11 %f  %f  %f  %f  %f\n",phi,phi_P,th_P,th,p_polar);
                }
                else if(th_P==1)
                {
                fprintf (ofp111,"%f  %f %f  %f %f \n",phi,phi_P,th_P,th, p_polar );
                printf ("IF lOOP111 %f  %f  %f  %f  %f\n",phi,phi_P,th_P,th,p_polar);
                }


        fprintf (ofp,"%f  %f  %f   %f   %f   %f   %f\n",th,th_P,phi,RA,DEC,p_polar,p_RA
_DEC);

    } //theta
    fprintf (ofp,"\n");

 } //phi

} //theta_p


 fclose(ifp2);
 fclose(ifp3);
 fclose(ofp);
 fclose(ofp1);
 fclose(ofp11);
 fclose(ofp111);

 printf("enter the value of thita in degree u want to fix, fth =");
 scanf("%f",&f_th);

 FILE *ifp_6;
 FILE *ofp_7;
 char str[100];
 ifp_6 = fopen("power_pattern.dat","r");
 ofp_7 = fopen("power-pattern.fixtheta.dat","w");
 fgets(str,100,ifp_6);
 fprintf (ofp_7,"#fixed-theta phi RA DEC power(th,phi) power(RA,DEC)\n");

 for(th=(th_0-r_th)*60;th<=(th_0+r_th)*60;th=(th+a_th))//so th and py are in arc
-minutes
 {
  for(phi=(phi_0-r_phi)*60;phi<=(phi_0+r_phi)*60;phi=(phi+a_phi))//1 degree = 60
 arc-minute
  {
   fscanf(ifp_6,"%f %f %f %f %f %f",&th,&phi,&RA,&DEC,&p_polar,&p_RA_DEC);
   if(th == (f_th*60))
   {
    fprintf(ofp_7,"%f   %f   %f   %f   %f   %f\n",(f_th*60),phi,RA,DEC,p_polar,p_RA_DE
C);
    printf("%f   %f   %f   %f   %f   %f\n",(f_th*60),phi,RA,DEC,p_polar,p_RA_DEC);

   }
```

```
  }
 }
 fclose(ofp_7);
 printf("enter the value of phi in degree u want to fix,fphi =");
 scanf("%f",&f_phi);

 FILE *ofp_8;

 ofp_8 = fopen("power_pattern.fixphi.dat","w");
 rewind(ifp_6);
 fgets(str,100,ifp_6);
 fprintf (ofp_8,"#theta fixed-phi RA DEC power(th,phi) power(RA,DEC)\n");
  for(th=(th_0-r_th)*60;th<=(th_0+r_th)*60;th=(th+a_th))//so th and py are in ar
c-minutes
 {
  for(phi=(phi_0-r_phi)*60;phi<=(phi_0+r_phi)*60;phi=(phi+a_phi))//1 degree = 60
 arc-minute
  {
   fscanf(ifp_6,"%f %f %f %f %f %f",&th,&phi,&RA,&DEC,&p_polar,&p_RA_DEC);
   if(phi == (f_phi*60))
   {
    fprintf(ofp_8,"%f   %f   %f  %f  %f  %f\n",th,(f_phi*60),RA,DEC,p_polar,p_RA_DE
C);
    printf("%f   %f   %f  %f  %f\n",th,(f_phi*60),RA,DEC,p_polar,p_RA_DEC);
   }
  }
 }
 fclose(ifp_6);
 fclose(ofp_8);
 return(0);
}



/* Generate a set of 30 random number of uniform distribution of some specific m
ean and rms which will be used as noise in the array pattern*/

int noise(float rms1,float rms2, float mean1,float mean2)
{
 FILE *ofp1 = NULL;
 FILE *ofp2 = NULL;
 ofp1 = fopen("amp_noise.dat","w");
 ofp2 = fopen("ph_noise.dat","w");
 int i;
 float a,b;

 for(i=0;i<30;i++)
 {
  a = rms1 * rand() + mean1;
  b = rms2 * rand() + mean2;

  fprintf(ofp1,"%f\n",a);
  fprintf(ofp2,"%f\n",b);
  printf("%d  %f  %f\n",i,a,b);
 }

 fclose(ofp1);
 fclose(ofp2);
}
```

```
/* make a program to convert the RA & DEC co-ordinate system in to theta (elivat
ion) and phy (azimath) co-ordinate system */

int RA_DEC_to_th_phi(float RA_hr,float RA_min,float RA_sec,float DEC_deg,float D
EC_min,float DEC_sec)
{
 float th_0, phi_0;
 phi_0 = (RA_sec * 15)/(60*60) + (RA_min * 15)/(60) + (RA_hr * 15);//phi calcula
ted in degree : 1 hr = 15 deg, 1 min = 15 arc-min, 1 sec = 15 arc-sec
 th_0 = 90 - ((DEC_deg) + (DEC_min/60) + (DEC_sec/(60*60)));// th in Degree

 printf("Theta_0 = %f\n",th_0);
 printf("Phi_0 = %f\n",phi_0);

 return(0);
}
```

PROGRAM NAME:: arraypattern_version3.c

Printed by Somprova Nandy


```
/* COMMENTS:    FINAL CODE OF THE PROGRAM
                MULTIPLE BEAM GENERTION OVER 2D SKY PLANE POSSIBLE*/


#include<stdio.h>
#include <math.h>

#define pi  2*acos(0)
#define min  pi/(2*9*600) //1 arc-minute = 0.000291005 rad


int RA_DEC_to_th_phi(float RA_hr,float RA_min,float RA_sec,float DEC_deg,float D
EC_min,float DEC_sec);
int noise(float rms1,float rms2,float mean1,float mean2);

int main()
{
 int n,m,no_th,no_phi,s,inp1,inp2,inp3,inp4;
 float x[30],y[30],z[30],l,th_P0,phi_P0,th_P[20],phi_P[20],d_th_P,d_phi_P,th_0,p
hi_0,r_th,a_th,r_phi,a_phi,f_th,f_phi,rms1,mean1,rms2,mean2,RA_hr,RA_min,RA_sec,
DEC_deg,DEC_min,DEC_sec;
 char fname[100];
 system("clear");
 printf("enter the rms-s and means you want for uniformly distributed 30 random numbers will be used as noise t
o the beam pattern corresponding to amplitude and phase respectively ");
 scanf("%f %f %f %f",&rms1,&mean1,&rms2,&mean2);
 noise(rms1,rms2,mean1,mean2);

 // Read the antenna co-ordinates from the input file
 FILE *ifp = NULL;
 int i;
 float x1[30], y1[30], z1[30];

 ifp = fopen("input.dat","r"); //input.dat is the input file for antenna co-ordinat
es in x,y,z plane.
 for(i=0;i<30;i++)
        fscanf(ifp,"%f %f %f\n",&x[i],&y[i],&z[i]);
 fclose(ifp);

 // Select the reference antenna co-ordinate from the list of the antennas
 printf("choose the antenna number you want as referrence n = ");
 scanf("%d",&n);
 printf("the antenna co-ordinates you select as reference are\n");
 printf("%5.2f    %5.2f    %5.2f\n",x[n-1],y[n-1],z[n-1]);

 // Subtract the ref antenna co-ordinate from the all antennas
 printf("\nthe corrected co-ordinates in meters are:\n");
 for(i=0;i<30;i++)
 { x1[i] = x[i] - x[n-1];
   y1[i] = y[i] - y[n-1];
   z1[i] = z[i] - z[n-1];
   printf("%5.2f    %5.2f    %5.2f\n",x1[i],y1[i],z1[i]);
 }

 // Get the No of antenna as well as the antenna nos
 printf("choose the number (must be <= 30) of antennas m = ");
 scanf("%i",&m);
 int j[30];
 printf("\n choose the sequence of antennas. You must input the reference antenna. \n");
```

```
 for(i=0;i<m;i++)
 {
  scanf("%i",&j[i]);
  printf("%i\n",j[i]);
 }

 // Get the antenna co-ordinates of the selected antennas
 int cnt;
 int count = 0;
 printf("the corresponding antenna co-ordinates are\n");
 for(i=0;i<30;i++)
 {
  for(cnt=0;cnt<m;cnt++)
  {      if(i == j[cnt])
         {       printf("%5.2f  %5.2f  %5.2f\n",x1[i],y1[i],z1[i]);
                 x[count] = x1[i]; y[count] = y1[i]; z[count] = z1[i];
                 count++;
         }
  }
 }
 printf("enter the reference direction (in sky) in RA(RA_hr,RA_min,RA_sec) and DEC(DEC_deg,DEC_min,DEC
_sec) co-ordinate system  ");
 scanf("%f %f %f %f %f %f",&RA_hr,&RA_min,&RA_sec,&DEC_deg,&DEC_min,&DEC_sec);
 RA_DEC_to_th_phi(RA_hr,RA_min,RA_sec,DEC_deg,DEC_min,DEC_sec);

 printf("\nenter the wavelength l in meter = ");
 scanf("%f",&l);
 printf("\nenter the no. of multiple beams to be generated in a given range of theta and given range of phi = ");
 scanf("%d %d",&no_th,&no_phi);

 printf("\nenter  theta_p and Phi_P which specify the direction for phasing the array  ");
 scanf("%f %f",&th_P0,&phi_P0);
 printf("\nenter separation in th_p (in arc min)and phi_p (in arc min) at which the beams are to be separated in thet
a direction ");
 scanf("%f %f",&d_th_P,&d_phi_P);
 printf("\nenter Theta_0 and Phi_0(in deg)which specify the direction for calculation of pattern ");
 scanf("%f %f",&th_0,&phi_0);
 printf("\nenter the range you want (in degree) and the accuracy you want in (arc-minute) in the direction of theta
or polar angle \n");
 scanf("%f %f",&r_th,&a_th);
 printf("\nenter the range you want( in degree) and the accuracy you want (in arc-minute) in the direction of phi o
r azimath \n");
 scanf("%f %f",&r_phi,&a_phi);


 FILE *ifp2;
 FILE *ifp3;
 FILE *ofp;
 FILE *ofpn1;
 FILE *ifpn;

 float x_comp,y_comp,z_comp,p_polar,p_RA_DEC,th,phi,RA,DEC,RA_0,DEC_0,k,c_polar,
s_polar,c_RA_DEC,s_RA_DEC,n1,n2;
 ifp2 = fopen("amp_noise.dat","r");
 ifp3 = fopen("ph_noise.dat","r");
 ofp = fopen("power_pattern.dat","w");
 ifpn = fopen("input_thp_phip.dat","w");
 fprintf (ifpn,"#theta_P    phi_P   \n");
```

```
 inp1=0;
 inp2=0;


 for(th_P[inp1]=th_P0*60-(no_th-1)*d_th_P;th_P[inp1]<th_P0*60+no_th*d_th_P;th_P[
inp1]=th_P[inp1-1]+d_th_P)     //for generation of input dat
                                                                  //file having
thp phip values
 {
     for(phi_P[inp2]=phi_P0*60-(no_phi-1)*d_phi_P;phi_P[inp2]<phi_P0*60+no_phi*d
_phi_P;phi_P[inp2]=phi_P[inp2-1]+d_phi_P)
        {

        fprintf(ifpn,"%f %f\n",th_P[inp1],phi_P[inp2]);
        inp2++;

        }
    inp1++;
 }
 fclose(ifpn);
 fprintf(stderr,"checking....after th phi writting");
 fprintf (ofp,"#theta theta_P phi_P phi RA DEC power(th,phi) power(RA,DEC)\n");
 k = 2*pi/l;
 s=0;

 char spr[300];
 ifpn=fopen("input_thp_phip.dat","r");
 fgets(spr,300,ifpn);
 for(inp3 = 0;inp3 <2*no_th-1; inp3++)  //for obtainin values of th_P,phi_P frm
input_thp_phip file
 {
 for(inp4 = 0;inp4 <2*no_phi-1; inp4++)
 {

  fscanf(ifpn,"%f %f\n",&th_P[inp3],&phi_P[inp4]);

 for(phi=(phi_0 - r_phi)*60;phi<=(phi_0 + r_phi)*60;phi=(phi + a_phi))//1 degree
 = 60 arc-minute
  {

  for(th=(th_0 - r_th)*60;th<=(th_0 + r_th)*60;th=(th + a_th))//so th and py are
 in arc-minutes
     {

        RA = 4 * phi; // RA is expressed in Second since 1 arc-minute = 4 sec
        DEC = (90*60) - th; //DEC is expressed in arc-minute
        RA_0 = 4 * phi_0;//Reference RA direction
        DEC_0 = (90*60) - th_0;//Reference DEC direction
        c_polar = 0;
        s_polar = 0;
        c_RA_DEC = 0;
        s_RA_DEC = 0;
        for(i=0;i<m;i++)
        {
          fscanf(ifp2,"%f",&n1);
          fscanf(ifp3,"%f",&n2);
          x_comp=x[i]*(sin(min*th)*cos(min*phi)-sin(min*th_P[inp3])*cos(min*phi_
P[inp4]));
          y_comp=y[i]*(sin(min*th)*sin(min*phi)-sin(min*th_P[inp3])*sin(min*phi_
```

```
P[inp4]));
          z_comp=z[i]*(cos(min*th)-cos(min*th_P[inp3]));

          c_polar += n1*(cos(k*(x_comp+y_comp+z_comp)) + n2);//the external n fo
r amp_noise and internal for ph_noise. Here both are taken from an uniform rando
m distribution. But for more sophisticated version we can add different distribu
tion, in principle they should be also different for phase and amplitude.

          s_polar += n1*(sin(k*(x_comp+y_comp+z_comp)) + n2);//sin in polar co-o
rdinate

          c_RA_DEC += n1*(cos(k*(x[i]*(sin(min*DEC)*cos(min*phi)-sin(min*DEC_0)*
cos(min*phi_0))+y[i]*(sin(min*DEC)*sin(min*phi)-sin(min*DEC_0)*sin(min*phi_0))+z
[i]*(cos(min*DEC)-cos(min*DEC_0)))) + n2);//cosine in RA DEC co-ordinate system

          s_RA_DEC += n1*(sin(k*(x[i]*(sin(min*DEC)*cos(min*phi)-sin(min*DEC_0)*
cos(min*phi_0))+y[i]*(sin(min*DEC)*sin(min*phi)-sin(min*DEC_0)*sin(min*phi_0))+z
[i]*(cos(min*DEC)-cos(min*DEC_0)))) + n2);
     }

        p_polar = (c_polar * c_polar)+(s_polar * s_polar);//resultant power patt
ern in th-phi
        p_RA_DEC = (c_RA_DEC * c_RA_DEC) +(s_RA_DEC * s_RA_DEC);//resultant powe
r pattern in RA-DEC

                sprintf(fname,"power_theta_p%d.dat", s);
                ofpn1 =fopen(fname,"a");
                fprintf (ofpn1,"%f  %f  %f   %f   %f\n",th_P[inp3],phi_P[inp4],th,
phi ,p_polar );
                printf ("IF lOOP%d  %f  %f          %f  %f  %f  %f\n",s,th_P0,th_P[inp3
],phi_P[inp4],th,phi,p_polar);
                //fprintf(ofpn1,"\n");
                fclose(ofpn1);


      fprintf (ofp,"%f  %f        %f        %f  %f  %f  %f  %f\n",th,th_P[inp3],phi_P[
inp4],phi,RA,DEC,p_polar,p_RA_DEC);

     } //theta
     fprintf (ofp,"\n");
     ofpn1 =fopen(fname,"a");
     fprintf(ofpn1,"\n");
     fclose(ofpn1);  // newline for new value of phi
 } //phi
printf("VALUE OF S %d\n",s);
s++;

} //generation of phi_P

}//generation of theta_P


 fclose(ifp2);
 fclose(ifp3);
 fclose(ofp);
 fclose(ifpn);

 printf("enter the value of thita in degree u want to fix, fth =");
 scanf("%f",&f_th);
```

```
 FILE *ifp_6;
 FILE *ofp_7;
 char str[100];
 ifp_6 = fopen("power_pattern.dat","r");
 ofp_7 = fopen("power-pattern.fixtheta.dat","w");
 fgets(str,100,ifp_6);
 fprintf (ofp_7,"#fixed-theta phi RA DEC power(th,phi) power(RA,DEC)\n");

 for(th=(th_0-r_th)*60;th<=(th_0+r_th)*60;th=(th+a_th))//so th and py are in arc
-minutes
 {
  for(phi=(phi_0-r_phi)*60;phi<=(phi_0+r_phi)*60;phi=(phi+a_phi))//1 degree = 60
 arc-minute
  {
   fscanf(ifp_6,"%f %f %f %f %f %f",&th,&phi,&RA,&DEC,&p_polar,&p_RA_DEC);
   if(th == (f_th*60))
   {
    fprintf(ofp_7,"%f   %f   %f  %f  %f  %f\n",(f_th*60),phi,RA,DEC,p_polar,p_RA_DE
C);
    printf("%f   %f   %f  %f  %f  %f\n",(f_th*60),phi,RA,DEC,p_polar,p_RA_DEC);

   }
  }
 }
 fclose(ofp_7);
 printf("enter the value of phi in degree u want to fix, fphi =");
 scanf("%f",&f_phi);

 FILE *ofp_8;

 ofp_8 = fopen("power_pattern.fixphi.dat","w");
 rewind(ifp_6);
 fgets(str,100,ifp_6);
 fprintf (ofp_8,"#theta fixed-phi RA DEC power(th,phi) power(RA,DEC)\n");
  for(th=(th_0-r_th)*60;th<=(th_0+r_th)*60;th=(th+a_th))//so th and py are in ar
c-minutes
 {
  for(phi=(phi_0-r_phi)*60;phi<=(phi_0+r_phi)*60;phi=(phi+a_phi))//1 degree = 60
 arc-minute
  {
   fscanf(ifp_6,"%f %f %f %f %f %f",&th,&phi,&RA,&DEC,&p_polar,&p_RA_DEC);
   if(phi == (f_phi*60))
   {
    fprintf(ofp_8,"%f   %f   %f  %f  %f  %f\n",th,(f_phi*60),RA,DEC,p_polar,p_RA_DE
C);
    printf("%f   %f   %f  %f  %f\n",th,(f_phi*60),RA,DEC,p_polar,p_RA_DEC);
   }
  }
 }
 fclose(ifp_6);
 fclose(ofp_8);
 return(0);
}



/* Generate a set of 30 random number of uniform distribution of some specific m
ean and rms which will be used as noise in the array pattern*/
```

```
int noise(float rms1,float rms2, float mean1,float mean2)
{
 FILE *ofp1 = NULL;
 FILE *ofp2 = NULL;
 ofp1 = fopen("amp_noise.dat","w");
 ofp2 = fopen("ph_noise.dat","w");
 int i;
 float a,b;

 for(i=0;i<30;i++)
 {
  a = rms1 * rand() + mean1;
  b = rms2 * rand() + mean2;

  fprintf(ofp1,"%f\n",a);
  fprintf(ofp2,"%f\n",b);
  printf("%d  %f  %f\n",i,a,b);
 }

 fclose(ofp1);
 fclose(ofp2);
}


/* make a program to convert the RA & DEC co-ordinate system in to theta (elivat
ion) and phy (azimath) co-ordinate system */

int RA_DEC_to_th_phi(float RA_hr,float RA_min,float RA_sec,float DEC_deg,float D
EC_min,float DEC_sec)
{
 float th_0, phi_0;
 phi_0 = (RA_sec * 15)/(60*60) + (RA_min * 15)/(60) + (RA_hr * 15);//phi calcula
ted in degree : 1 hr = 15 deg, 1 min = 15 arc-min, 1 sec = 15 arc-sec
 th_0 = 90 - ((DEC_deg) + (DEC_min/60) + (DEC_sec/(60*60)));// th in Degree

 printf("Theta_0 = %f\n",th_0);
 printf("Phi_0 = %f\n",phi_0);

 return(0);
}
```

DATA FILES

GMRT COORDINATES:

X[I]	Y[I]	Z[I]
6.95	687.88	-20.04
13.24	326.43	-40.35
0.00	0.00	0.00
-51.17	-372.71	133.59
-51.08	-565.94	123.43
79.09	67.82	-246.59
71.23	-31.44	-220.58
130.77	280.67	-400.33
48.56	41.92	-151.65
191.32	-164.88	-587.49
102.42	-603.28	-321.56
209.28	174.85	-635.54
368.58	-639.53	-1117.92
207.30	-473.71	-628.63
-348.04	2814.55	953.67
-707.58	4576.00	1932.46
-1037.11	7780.69	2903.29
-1177.37	10200.00	3343.20
-1571.32	12073.46	4543.13
942.99	633.92	-2805.93
1452.92	-367.07	-4279.16
2184.54	333.03	-6404.96
3072.86	947.68	-8979.50
4592.71	-369.04	-13382.48
-201.32	-1591.90	591.32
-482.67	-3099.41	1419.39
-992.01	-5199.90	2899.11
-1734.55	-7039.03	5067.53
-2706.09	-8103.13	7817.14
-3102.11	-11245.60	8916.26

LINEAR X-Y ARRAY

X[I]	Y[I]	Z[I]
0000.0000	0000.0000	0000.0000
0100.0000	0100.0000	0000.0000
0200.0000	0200.0000	0000.0000
0300.0000	0300.0000	0000.0000
0400.0000	0400.0000	0000.0000
0500.0000	0500.0000	0000.0000
0600.0000	0600.0000	0000.0000
0700.0000	0700.0000	0000.0000
0800.0000	0800.0000	0000.0000
0900.0000	0900.0000	0000.0000
1000.0000	1000.0000	0000.0000
1100.0000	1100.0000	0000.0000
1200.0000	1200.0000	0000.0000
1300.0000	1300.0000	0000.0000
1400.0000	1400.0000	0000.0000
1500.0000	1500.0000	0000.0000
1600.0000	1600.0000	0000.0000
1700.0000	1700.0000	0000.0000
1800.0000	1800.0000	0000.0000
1900.0000	1900.0000	0000.0000
2000.0000	2000.0000	0000.0000
2100.0000	2100.0000	0000.0000
2200.0000	2200.0000	0000.0000
2300.0000	2300.0000	0000.0000
2400.0000	2400.0000	0000.0000
2500.0000	2500.0000	0000.0000
2600.0000	2600.0000	0000.0000
2700.0000	2700.0000	0000.0000
2800.0000	2800.0000	0000.0000
2900.0000	2900.0000	0000.0000

REFERENCES

1. Low frequency Radio Astronomy
   Edited by : Jayaram N. Chengalur ,Yashwant Gupta , K.S Dwarakanath
2. Radioastronomy
   By: J.D Krauss
3. .Antenna Theory – Analysis and Design
   By: C. A Balanis
4. Spherical Astronomy
   By: R.M Green
5. Programming in ANSI C
   By: E.Balaguruswamy
6. http://www.analyzemath.com/antenna_tutorials/antenna_arrays.html
7. http://plasma.pmo.ac.cn/sources/teaching/jk1/lectures/node84.htm
8. http://www.ee.surrey.ac.uk/Personal/D.Jefferies/antarray.html