C/C++ BASED SOFTWARE FOR RECTANGULAR MICROSRTIP PATCH ANTENNA AND ARRAYS

By

HIMANSHU CHOPRA - 021201

JAIDEEP SINGH - 021211

MAY-2006

Submitted in partial fulfillment of the Degree of Bachelor of Technology

**DEPARTMENT OF
COMPUTER SCIENCE ENGINEERING
JAYPEE UNIVERSITY OF INFORMATION
TECHNOLOGY - WAKNAGHAT**

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT – 173215

CERTIFICATE

This is to certify that the work entitled, "C/C++ based software for Rectangular Microstrip Patch Antenna and Arrays" submitted by Himanshu Chopra (021201) and Jaideep Singh (021211) in partial fulfillment for the award of degree of Bachelor of Technology in Computer Sciences Engineering, Jaypee University of Information Technology has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Sdh 12/5/06

Sunil K. Khah

Assistant Professor

Jaypee University of Information Technology

Waknaghat – Solan (HP)

ACKNOWLEDGEMENT

We would like to sincerely thank our Project Guide Dr. Sunil K. Khah for his inspiring guidance, which led to the conception of this project, and his erudite support, which has seen us to the completion of the project. We would also like to thank him for his timely advice and constant motivation throughout the life span of the project. The comfort level that he always created motivated us to perform at our best. We would like to thank all the staff members of the department and the laboratory assistants who helped us in arranging for the required material, for the implementation of the project. We would also like to thank our colleagues and friends for their vital criticism. They helped to identify the bottlenecks and scope for refinement.

We would also like to thank our parents for giving us immense strength not only to set high goals, but also to achieve them. This effort could not have been possible without their support.

Himanshu Chopra

Jaideep Singh

TABLE OF CONTENTS

Chapter No.	Topic	Page No.
	List of figures	i
	List of abbreviations	ii
	List of Equations	iii
	Abstract	iv
1	Introduction	1
2	Microstrip patch Antenna	3
	2.1 Motivation for Microstrip patch Antenna	4
	2.2. Advantages and Disadvantages	4
	2.3 Design Specifications	5
	2.3.1 Single Rectangular Patch	5
	2.3.2 Basic Equations	8
	2.4 Arraying of elements	10
	2.4.1 Linear Array	11
	2.4.2 Circular Array	12
	2.5 Software Design	13
3	Results and Conclusions	16
	3.1 Results	16
	3.2 Conclusion	26
	3.3 Future Scope	27
	Bibliography	
	Appendices	
	A Design of Rectangular Microstrip Patch Antenna	
	B Far Field Expressions	
	C Arrays of Microstrip Patch Antenna	

LIST OF FIGURES

Figure No.	List of Figures	Page No.
Figure 1	Structure of a Single Rectangular Microstrip patch Antenna	3
Figure 2	Shapes of Microstrip Patch Antenna	6
Figure 3	Linear Array of Rectangular Microstrip Patch Antenna	12
Figure 4	Circular Array of Rectangular Microstrip Patch Antenna	12
Figure 5	Block Diagram of the Simulator Software	14
Figure 6	Flow Chart for the simulator software	15
Figure 7	Theoretical Results	16
Figure 8	Input Parameters - Rectangular Microstrip Patch Antenna	17
Figure 9	Computed Results	17
Figure 10	Arrays of Microstrip Patch Antenna	18
Figure 11	Computation of Radiation Field Components	18
Figure 12	Graph for phi = 0 Plane	19
Figure 13	Graph for phi = 90 Plane	20
Figure 14	Radiation Pattern for Linear Array	21
Figure 15	Radiation Pattern for Circular Array	22
Figure 16	Main page for the Web Portal	23
Figure 17	Profile of a Microstrip Patch Antenna	24
Figure 18	Results for Microstrip Patch Antenna	25

LIST OF SYMBOLS

AF_linear	Array Factor for Linear Array
AF_circular	Array Factor for Circular Array
c	Speed in free space = 3 e 8 m/s
D_g	Directivity of Microstrip Patch Antenna
ϵ_r	Dielectric Constant of the Substrate
ϵ_{reff}	Effective Dielectric Constant of the substrate
E_θ	Horizontal component of the Radiation Field Pattern
E_ϕ	Vertical component of the Radiation Field Pattern
f_0	Frequency of operation
h	Height of the dielectric substrate
ISRO	Indian Space Research Organization
L	Actual Length of a Microstrip Patch
ΔL	Length extension of the Microstrip Patch
L_{eff}	Effective length of the Microstrip Patch
L_g	Ground Plane location ordinate
P	Power Radiated through the Microstrip Patch
R	Radiation Field Pattern of the Microstrip Patch
t	Thickness of the rectangular Microstrip Patch
W	Width of the microstrip patch
W_g	Ground Plane Location coordinate

LIST OF EQUATIONS

Equation No	Equation	Page No.
1.1	Width of the Rectangular Patch	8
1.2	Effective dielectric constant of the substrate	8
1.3	Effective Length of the Rectangular Patch	8
1.4	Length Extension of the Rectangular Patch	9
1.5	Actual Length of the Rectangular Patch	9
1.6	Ground Plane Location (length)	9
1.7	Ground Plane Location (width)	9
1.8	Radiation Field Pattern (horizontal component)	10
1.9	Radiation Field Pattern (vertical component)	10
1.10	Radiated Power	10
1.11	Directivity of Rectangular Microstrip Patch Antenna	10
1.12	Array Factor for Linear Array	12
1.13	Array Factor for Circular Array	13

ABSTRACT

A C/C++ based software for the simulation of Rectangular Microstrip Patch Antenna and their Arrays has been developed. The simulation software is designed to be very user friendly as evident by the fact that only the basic parameters for the single rectangular patch are to be provided by the user. The software calculates the results for various design parameters of an antenna and calculates basic radiation characteristics efficiently. The software displays the results clearly to the user. If the simulation for an array is run, the simulator software configures the single rectangular patch parameters for the array simulation. The design is based on the basic antenna equations of the Rectangular Microstrip Patch Antenna. The arrays are configured using the pattern multiplication approach. The project aims to provide free availability of this software for educational purposes via internet. The web portal is loaded with the basic study material for a single rectangular microstrip patch antenna and some more useful links to more study material available on the internet. The source code for the simulation program is also available on the web portal.

INTRODUCTION

Chapter 1

Scientifically, an antenna is an electronic component designed to transmit or receive radio signals and other electromagnetic waves.

Physically, an antenna is an arrangement of conductors designed to radiate (transmit) an electromagnetic field signal in response to an applied alternating voltage (or the applied alternating electric current). Similarly, if the antenna is placed in an electromagnetic field, the field induces an alternating current in the antenna (or a voltage difference between its terminals), which further drives the connected circuitry.

There are two fundamental types of antenna:

- omni-directional (radiate equally in all directions of the plane)
- directional (radiates more in one direction than in the other).

All antenna radiate some energy in all directions but careful designing and construction results in large directivity in certain directions and negligible energy radiated in other directions. Therefore, the law of conservation of energy is applicable here and satisfied.

By adding additional conducting rods or coils (called elements) and varying their length, spacing, and orientation, an antenna with specific desired properties can be designed.

Some parameters that affect an antenna's performance and should be carefully considered during the design process are:

- Resonant frequency
- Impedance
- Gain aperture
- Radiation Pattern
- Polarization
- Efficiency
- Bandwidth.

The simulator software presented here calculates basic parameters for a single rectangular microstrip patch antenna and arrays.

The results calculated by the simulator software are compared with the standard results. The experimental results are also used to support the software results.

The radiation characteristics of the rectangular patch being designed are also calculated and plotted by the simulation software. The power radiated through the rectangular patch and the directivity of the antenna are also calculated by the presented software.

MICROSTRIP PATCH ANTENNA

Chapter 2

Microstrip Patch Antennae represent one family of compact antennae that offers the benefits of a conformal nature and the capability of ready integration with the printed circuitry of a communication system.

In its most basic form, a microstrip patch antenna consists of a radiating patch on one side of a dielectric slab which has a ground plane on the other side, as shown in the following figure 1.

The patch is made of a conducting material like copper (Cu) or gold (Au) and is photo etched on the dielectric slab, and can take any regular or irregular shape.

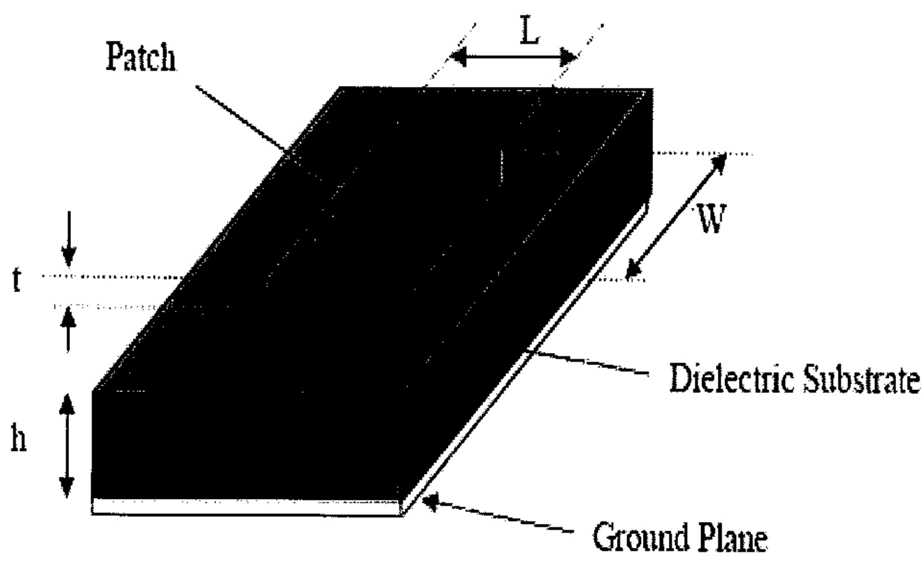


Fig. 1: Structure of a Microstrip Patch Antenna

2.1 MOTIVATION FOR MICROSTRIP PATCH ANTENNA

Microstrip patch antennas have recently gained huge attention for use in wireless applications due to their low-profile structure. Therefore, they are extremely compatible for embedded antennae in handheld wireless devices such as cellular phones, pagers, satellites etc. . Microstrip patch antennas are fast replacing some of the old and bulky antennae used in the telemetry and communication departments, overseas communication services and space communications. ISRO has proposal to replace the bulky and conventional UHF antenna mounted on various GSLVs and other classes of space vehicles for its future applications.

After launching recent satellites indigenously, India has become a big player in the world in the field of space research. Therefore, the present work is supposed to be extremely useful for satellite communication programme of our country. Antenna on missiles need to be thin and are often Microstrip patch antenna..

Since the individual patch antenna have a narrow bandwidth, attempts have also been made to improve the bandwidth available through an individual element.

An easy method is to design the individual patch antenna by using either dual frequency or using a thicker dielectric substrate.

Microstrip patch antenna exhibit low directivity and gain. Hence, they are not very feasible practically, when used individually. However, antenna with high directivity and gain are realized by combining a large number of individual microstrip patches into arrays of various configurations.

2.2 ADVANTAGES AND DISADVANTAGES

Some of the advantages and disadvantages of microstrip patch antenna are listed below. Microstrip patch antennas have numerous advantages over conventional antenna.

Some of their principal advantages are given below:

- Light weight, low volume.
- Low fabrication cost (due to bulk production, cost per unit is low).
- Supports both linear as well as circular polarization.
- Can be easily integrated with microwave integrated circuits (MICs).
- Capable of dual and triple frequency operations.
- Mechanically robust when mounted on rigid surfaces.

Microstrip patch antennas suffer from a number of disadvantages as compared to conventional antenna.

Some of their major disadvantages are given below:

- Narrow bandwidth
- Low efficiency
- Low Gain
- Extraneous radiation from feeds and junctions
- Low power handling capacity.

2.3 DESIGN SPECIFICATIONS - MICROSTRIP PATCH ANTENNA

2.3.1 Single Rectangular Microstrip Patch Antenna

Although conceptually simple, the Microstrip patch Antenna design presented a number of challenges.

The patch may be circular, triangular, elliptical or any other common shape. The antenna may even be contoured as an irregular shape (e.g. Heptagon, Decagon etc.).

Some shapes which can be taken up by the patch of an antenna are shown in figure 1.2

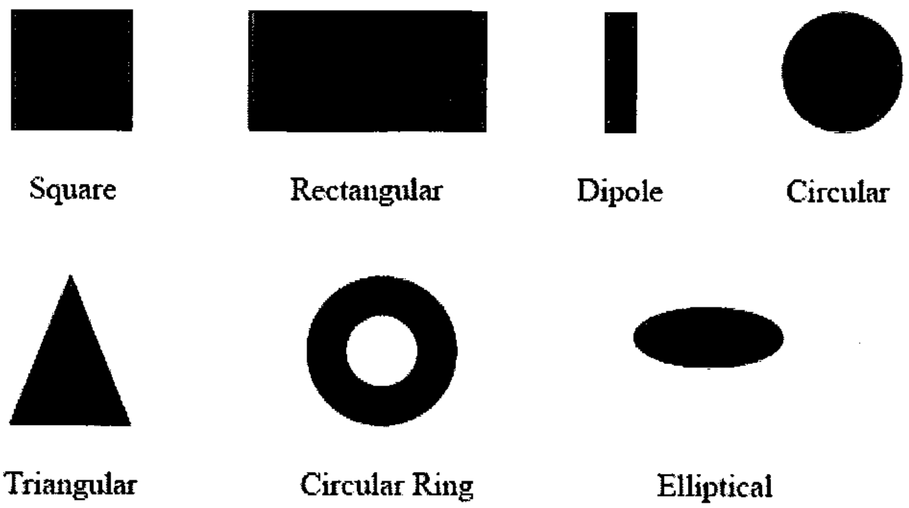


Fig. 2: Shapes & configurations for a Patch Antenna

However, in order to simplify analysis and performance prediction, the patch is generally rectangular.

For a rectangular patch, the length L of the patch is usually maintained between 33%-50% of the free space wavelength. Thus, L typically is = $0.33\lambda < L < 0.5\lambda$, where λ is the free-space wavelength of the signals. The spacing between the conductor and the ground plane is much smaller than the free space wavelength. Thus, the patch is selected to be very thin such that $t \ll \lambda$ (where *t* is the patch thickness and λ is the operating wavelength).

The height *h* of the dielectric substrate is usually $0.003\lambda \leq h \leq 0.05\lambda$.

The dielectric constant of the substrate (ε_r) is typically in the range $2.2 \leq \varepsilon_r \leq 12$.

Microstrip patch antennae radiate primarily because of the fringing fields between the patch edge and the ground plane and are typically used for operation in micrometer and millimeter range of wavelengths of radiations.

For good antenna performance, a thick dielectric substrate having a low dielectric constant is desirable since this provides better efficiency, larger bandwidth and better radiation patterns. However, such a configuration leads to a larger antenna size.
In order to design a compact Rectangular Microstrip patch antenna, higher dielectric constants must be used. The constraint for the designer here is that these dielectrics with high values of dielectric constants are less efficient and lead to narrower bandwidths.

There are certain parameters which are to be supplied by the user in order to actuate the designing of a Rectangular Microstrip patch antenna.
It is these specifications that form the user requirements for our simulator program.

The essential parameters that constitute the user requirements are:

- Frequency of operation (f_o):

  The resonant frequency of the antenna must be selected appropriately. The resonant frequency selected should be in the GHz range.

- Dielectric constant of the substrate (ϵ_r):

  A substrate with a high dielectric constant is preferable since it reduces the dimensions of the antenna.

- Height of dielectric substrate (h):

  For the microstrip patch antenna to be used in wireless handheld devices, it is essential that the antenna is not bulky. Hence, the height of the dielectric substrate is selected in the mm range.

Hence a compromise must be reached between antenna dimensions and antenna performance. Based on the user requirements, an antenna with specific properties can be designed.

2.3.2 Basic Equations

The basic design parameters for microstrip patch antenna are obtained using the equations that follow.

The parameters like width (W), length (L), effective dielectric constant (ϵ_{reff}) effective length (L_{eff}) are calculated using the given equations.

The width of the Microstrip patch antenna is given by equation (1.1) as:

$$W = \frac{c}{2f_o\sqrt{\frac{(\varepsilon_r + 1)}{2}}} \quad (1.1)$$

The dielectric constant of the dielectric substrate is provided by the user. This dielectric constant is used in Equation (1.2) and to get the effective dielectric constant.

$$\varepsilon_{reff} = \frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2}\left[1 + 12\frac{h}{W}\right]^{-\frac{1}{2}} \quad (1.2)$$

Equation (1.3) gives the effective length of the rectangular patch as:

$$L_{eff} = \frac{c}{2f_o\sqrt{\varepsilon_{reff}}} \quad (1.3)$$

Equation (1.4) gives the length extension ΔL for the patch of a microstrip patch antenna as:

$$\Delta L = 0.412h \frac{(\varepsilon_{reff} + 0.3)\left(\frac{W}{h} + 0.264\right)}{(\varepsilon_{reff} - 0.258)\left(\frac{W}{h} + 0.8\right)}$$

(1.4)

The actual length L is obtained by equation (1.5) as:

$$L = L_{eff} - 2\Delta L$$

(1.5)

For the rectangular microstrip patch antenna, the ground plane dimensions are:

$$L_g = 6h + L = 6(1.5) + 22.8 = 31.8mm$$

$$W_g = 6h + W = 6(1.5) + 31.1 = 40.1mm$$

(1.6 & 1.7.)

A Rectangular Microstrip patch antenna radiates either due to the current distribution on the patch, or due to the field on the slot edges.

The field accumulated at the slot edges may be represented by equivalent magnetic currents. The basic equations for radiation through a microstrip patch antenna are obtained using vector wave functions approach.

The radiation field components E_θ and $E_\varnothing$ are given as:

$$E_\theta = k_2 \exp^{(X \sin\theta)} \cos(X \sin\theta)$$

$$E_\phi = \frac{k_2 \exp(X \sin\theta) \sin(Y \sin\theta) \cos\theta}{4}$$

$$\text{where},$$

$$X = \frac{k_o h \sin\theta \cos\phi}{2}$$

$$Y = \frac{k_0 W}{2} \cos\phi$$

(1.8-1.9)

The power radiated through the patch is given by equation 1.10 as:

$$p = \frac{1}{240\pi} \left[\int_0^{\pi} \int_0^{2\pi} R \sin\theta \, d\theta d\phi \right] \quad (1.10)$$

The directive gain D_g for a single rectangular microstrip patch antenna is given by:

$$D_g = \frac{4\pi P_{max}}{P_r} \quad (1.11$$

2.4 Arraying of Microstrip Patch Antenna

Microstrip patch antennas have a few disadvantages like low gain, narrow bandwidth and poor efficiency.

Arraying of microstrip patch antenna is the simplest technique to overcome these drawbacks and significantly improve the practical feasibility of microstrip patch antenna. The essence of arraying is that multiple microstrip patch antennae are put together (on a single dielectric substrate) in a predetermined order. The basic idea is that these elements will work in a constructive manner, thus improving the bandwidth, gain and efficiency characteristics of the cluster.

Microstrip patch antenna can be arrayed in the following ways.

- Linear Array
- Planar Array
- Circular Array

In our present project, we have worked on linear and circular arrays. Planar arrays have not been covered.

The Radiation equations of the array are given by the multiplication approach.

$$E_\theta = E_{\theta 1} + E_{\theta 2} + E_{\theta 3} + \ldots\ldots\ E_{\theta N}$$

$$E_\theta = |E_\theta| * AF$$

, where $E\theta 1$, $E_{\theta 2}$, $E_{\theta 3 \ldots}$ $E_{\theta N}$ are the radiation equations of individual elements of the array. The summation of these basic parameters is done by vector addition. They have a constant phase difference between them. The distance between elements is generally kept λ/2 (approx.) in order to avoid coupling between elements.

2.4.1 Linear Array

In this type of arrangement, all the individual elements are put in a linear order.

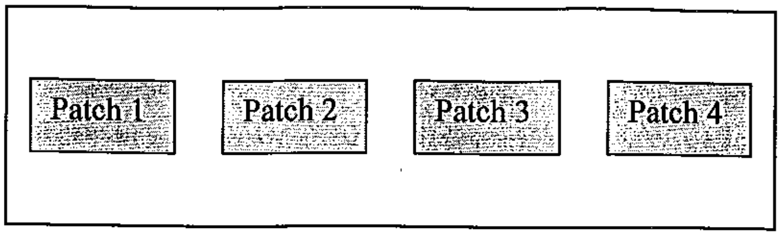


Fig.3: Linear Array Configuration of Microstrip Patch Antenna

The following equation 1.12 describes the Array factor for a linear array of patch antenna.

$$AF\_linear = N \cos\left(\beta_0 d \cos\theta + \beta\right)\cos\left(\frac{\beta_0 d \cos\theta}{2}\right) \quad (1.12)$$

2.4.2 Circular Array

Circular Array is another famous and popular arrangement of arraying microstrip patch antenna.

The circular array arrangement is given in the following figure.

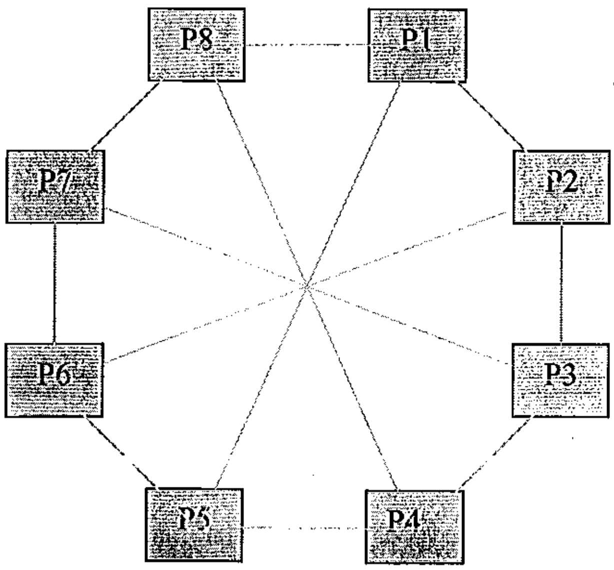


Fig. 4: Circular Array of Microstrip Patch Antenna

The equation to calculate the Array Factor for Circular Array arrangement is given below.

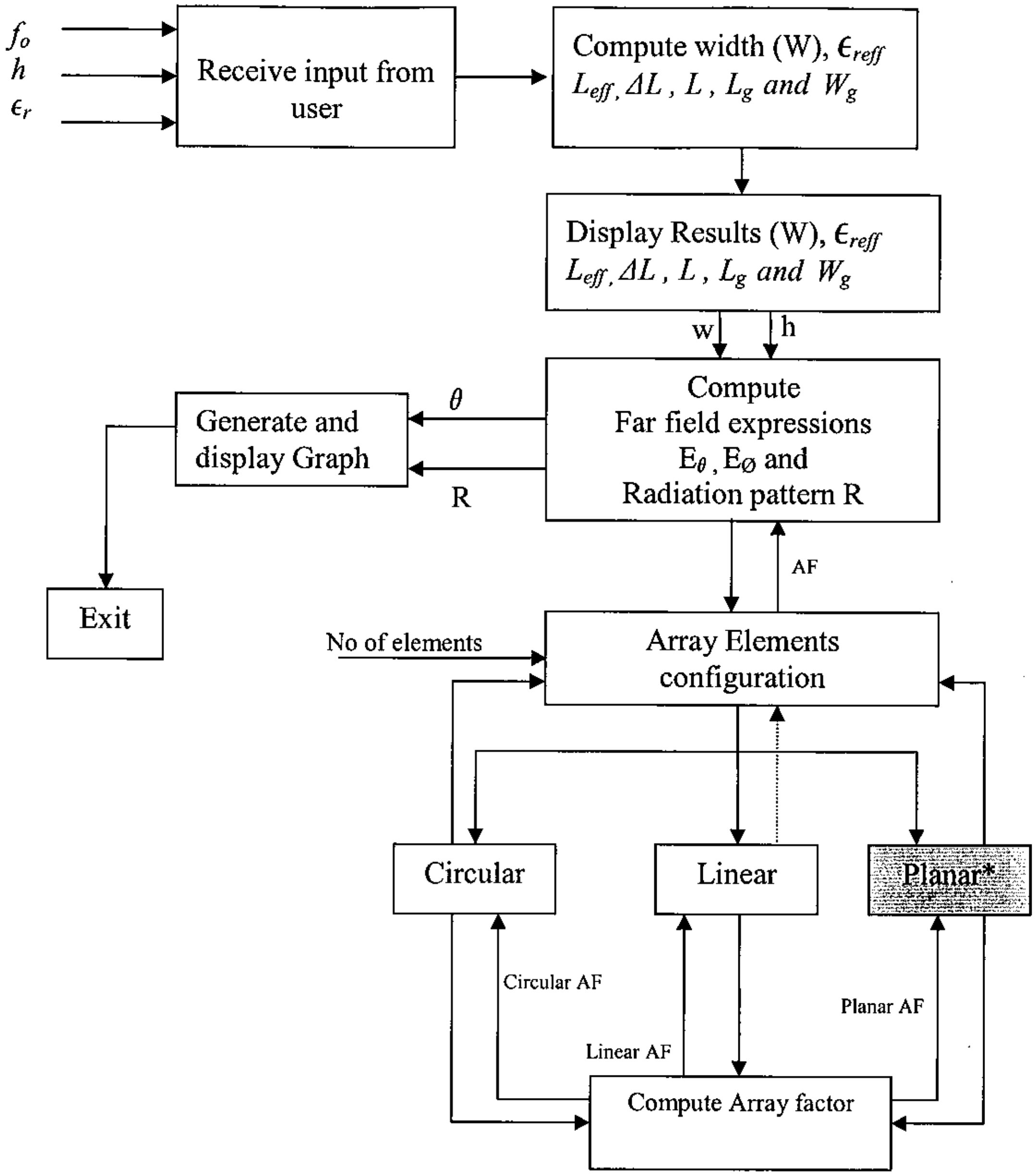
$$AF\_circular = \sum_{m=1}^{N} \exp^{j\{\beta_0 \rho \sin\theta \cos(\phi - \phi m) + \beta 1\}} \tag{1.13}$$

2.5 SOFTWARE DESIGN

A C/C++ software is designed for the simulation of a Rectangular Microstrip Patch Antenna. In this software, only the basic input parameters are given as an input. The first step is to design the antenna which requires calculating the physical dimensions of the antenna. The second step involves the computation of far field expressions. The graphs for the far-field expressions are also plotted by the software. The final step involves computation of parameters for arrays. First, the user selects the type of array desired. The software calculates the parameters for Linear and Circular Arrays. The user is prompted to input two parameters, the number of microstrip patches and the distance between consecutive elements. The software then calculates the configuration for the array.

The steps involved in the designing of the software are self explained by the block diagram and the flow chart of the software that follow.

The block diagram for the simulator software is given as follows..



* not incorporated in present work

Fig. 5: Block Diagram for the Simulator Software

The control flow chart for the simulator program is as follows.

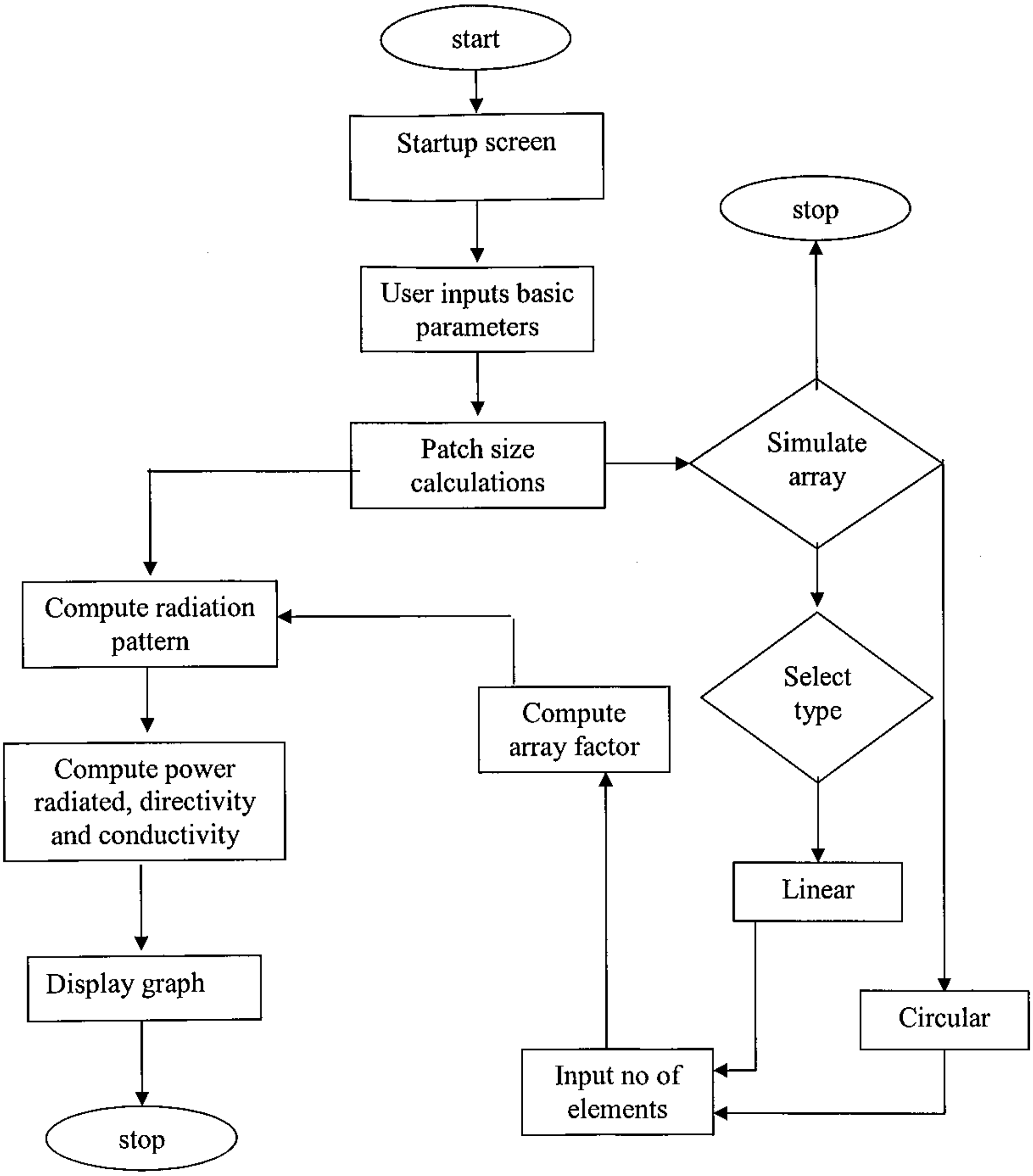


Fig. 6: Flow Chart for the Simulator Software

RESULTS AND CONCLUSION

Chapter 3

3.1 RESULTS

The simulator software results are obtained for different operating frequencies and materials used as the substrate. In order to verify our results we have first presented the standard numerical results for Rectangular Microstrip patch antenna.

A table presenting both the results is given below.

Considering the following values for parameters are supplied by the user, the results are shown.

$f_o = 1.9\ GHz$

$h = 1.5\ mm$

$\epsilon_r = 11.9$

We have the following theoretical results for the simulation of a single patch.

Parameter	Numerically Computed Value	Software Computed Value	Units	Error %
W	31.101	31.578	mm	-1.52
ϵ_{reff}	10.7871	10.439	-	+3.22
L_{eff}	24	23.288	mm	+2.9
ΔL	6.3455 e -4	63582 e -4	mm	-0.21
L	22.4	22	mm	+1.8
L_g	31.8	31.008	mm	+2.55
W_g	40.1	40.578	mm	-1.19

Fig. 7: Theoretical Results for Microstrip Patch Antenna

From the above mentioned results, it is very evident that the software results are in agreement with the numeric results. The error percentages are also listed in the table. These errors occur due to the fact that all numeric calculations are done by keeping the minimum number of significant figures.

The results from the simulator software are shown below.

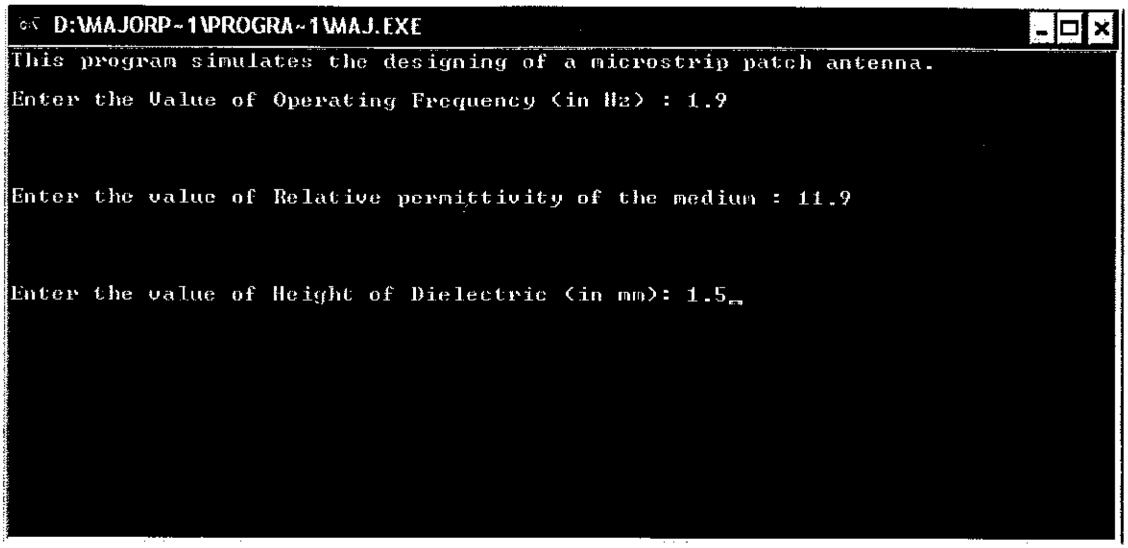


Fig. 8: Input Parameters for a Microstrip Patch Antenna

Once the above mentioned parameters are provided, the simulator software computes the vital parameter and displays the results. These are shown below.

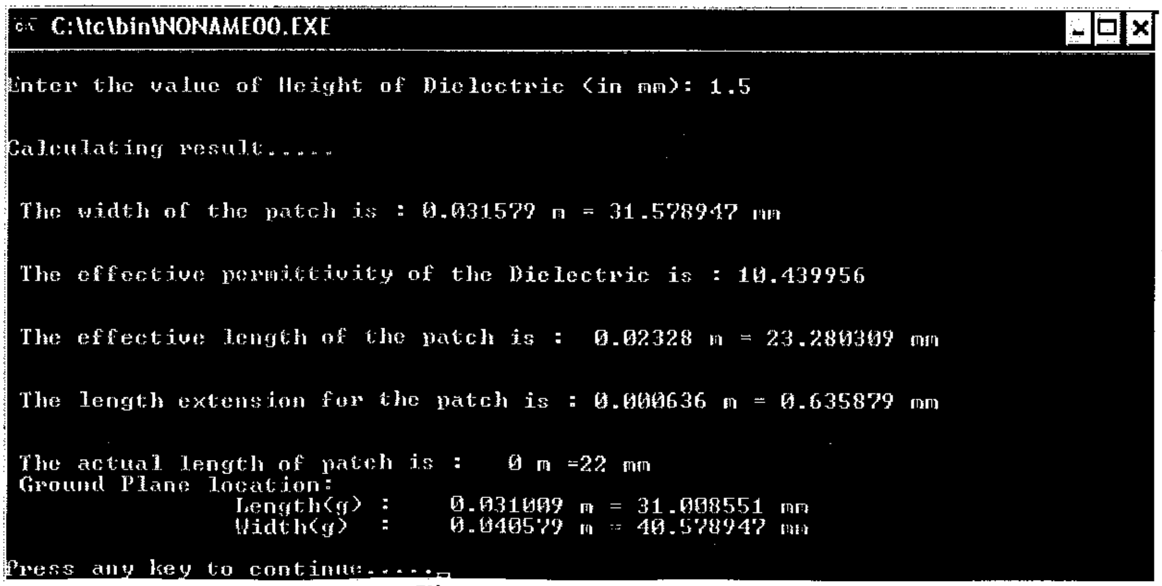


Fig. 9: Computed Results

Once the simulation is done for the designing of a single Rectangular Microstrip Patch Antenna, The software prompts the user to configure the array of Microstrip Patch Antenna.The snapshot is given below.

```
C:\TC\BIN\NONAME00.EXE

Switching to arraying of microstrip antenna.....

*******MENU*******

1. Linear Array

2. Circular Array

Enter your choice : 1

Enter No. of array_elements : -6
        No of elements can't be negative.Please enter again...
Enter No. of array_elements : 4
Enter the distance b/n consecutive elements (in mm): 1.5
```

Fig. 10: Arraying of Microstrip Patch Antenna

The Following snapshot displays the computation of Radiation Field components based on user provided inputs.

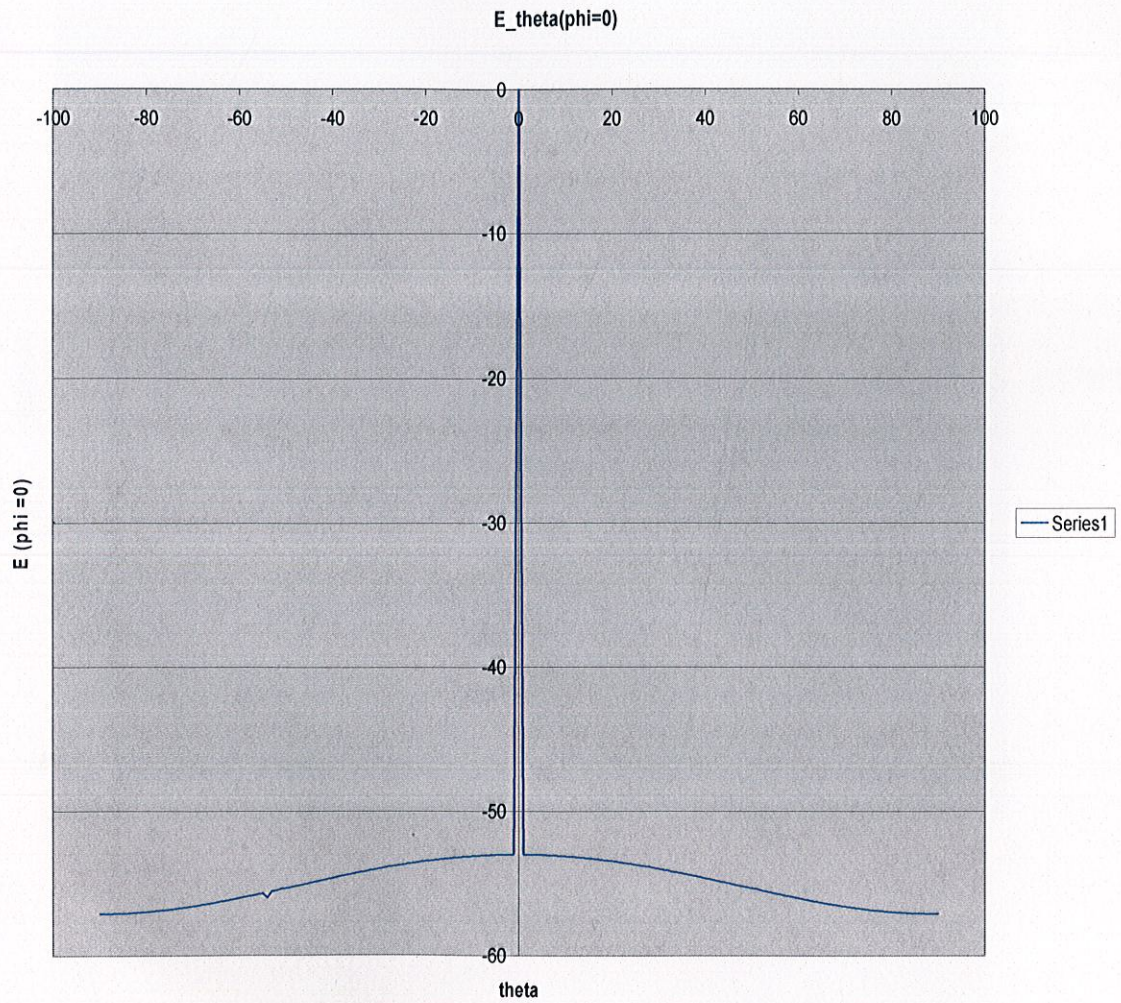
```
C:\tc\bin\NONAME00.EXE

Generating Data for phi = 0 degrees

-90      6.708546e-11
-89      6.709527e-11
-88      6.712304e-11
-87      6.716875e-11
-86      6.723237e-11
-85      6.731386e-11
-84      6.741316e-11
-83      6.75302e-11
-82      6.76649e-11
-81      6.781716e-11
-80      6.798687e-11
-79      6.81739e-11
-78      6.837813e-11
-77      6.85994e-11
-76      6.883754e-11
-75      6.909238e-11
-74      6.936372e-11
-73      6.965136e-11
-72      6.995506e-11
-71      7.027459e-11
-70      7.06097e-11
```

Fig. 11: Computation of Radiation Field components

The graphs for the radiation pattern of a single Rectangular Microstrip Patch Antenna are plotted below.

The graphs have the horizontal and the vertical components of the radiation pattern are plotted on the Y-axis and the angle theta is represented on the X-axis.

The graph for the horizontal component of the radiation field pattern is plotted. The following graph shows the radiation field component E_θ with φ= 0.



F

Fig.12: Graph for phi = 0 plane

The next figure represents the graph for the vertical component of the radiation pattern for a single patch. The graph for E_θ, $\varphi = 90$, is plotted as shown below. The peak power handling occurs at theta = 0.

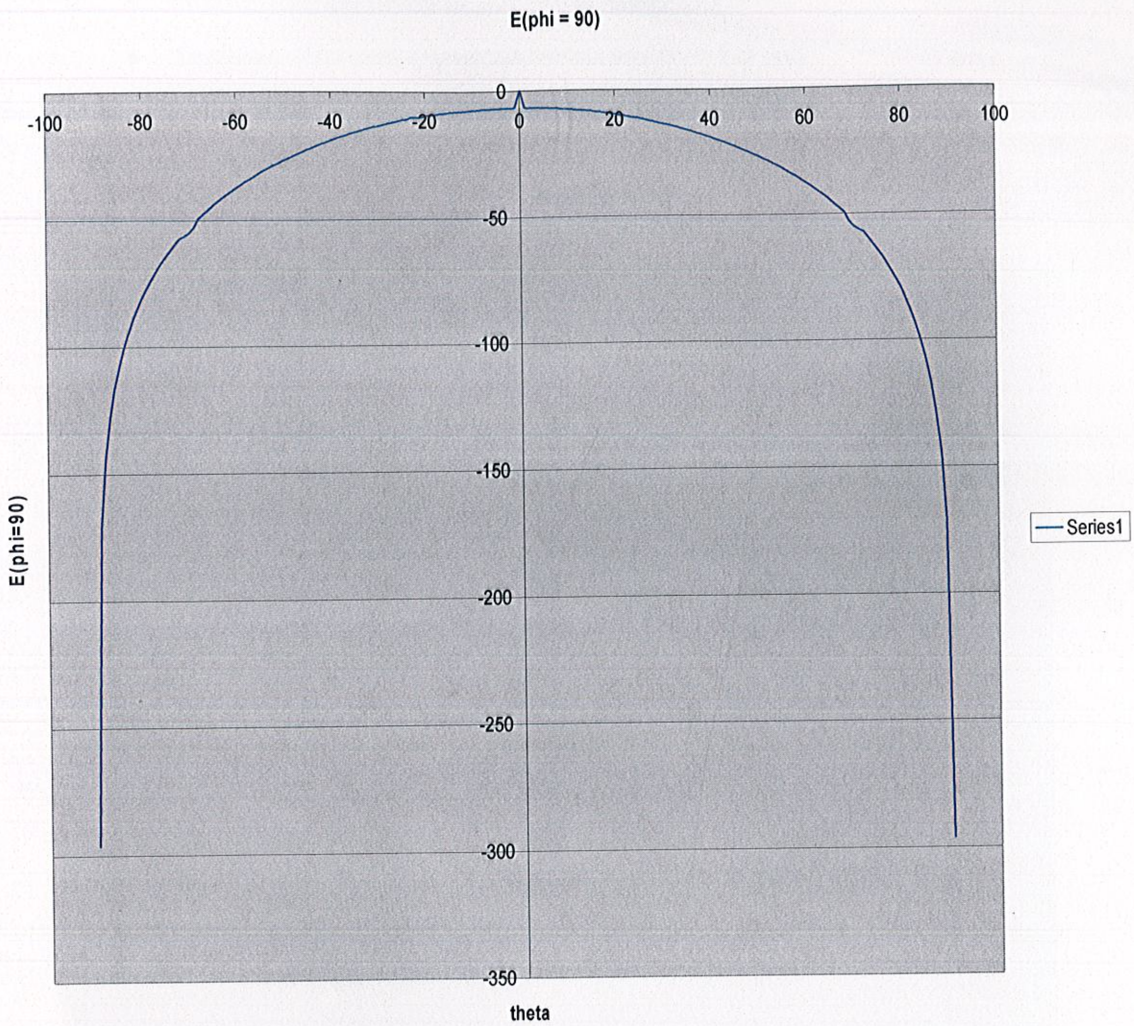


Fig. 13: Graph for phi = 90 plane

The following graph represents the Radiation Pattern obtained for arrays of elements.

The various individual elements of the array pattern work constructively and lead to better efficiency, higher bandwidth and greater directivity.

The graph is plotted for the following input values:

- Number of elements in the linear array = 8
- Distance between consecutive elements = 1.5 mm

The plot for the Radiation Pattern of a linear array is given below.

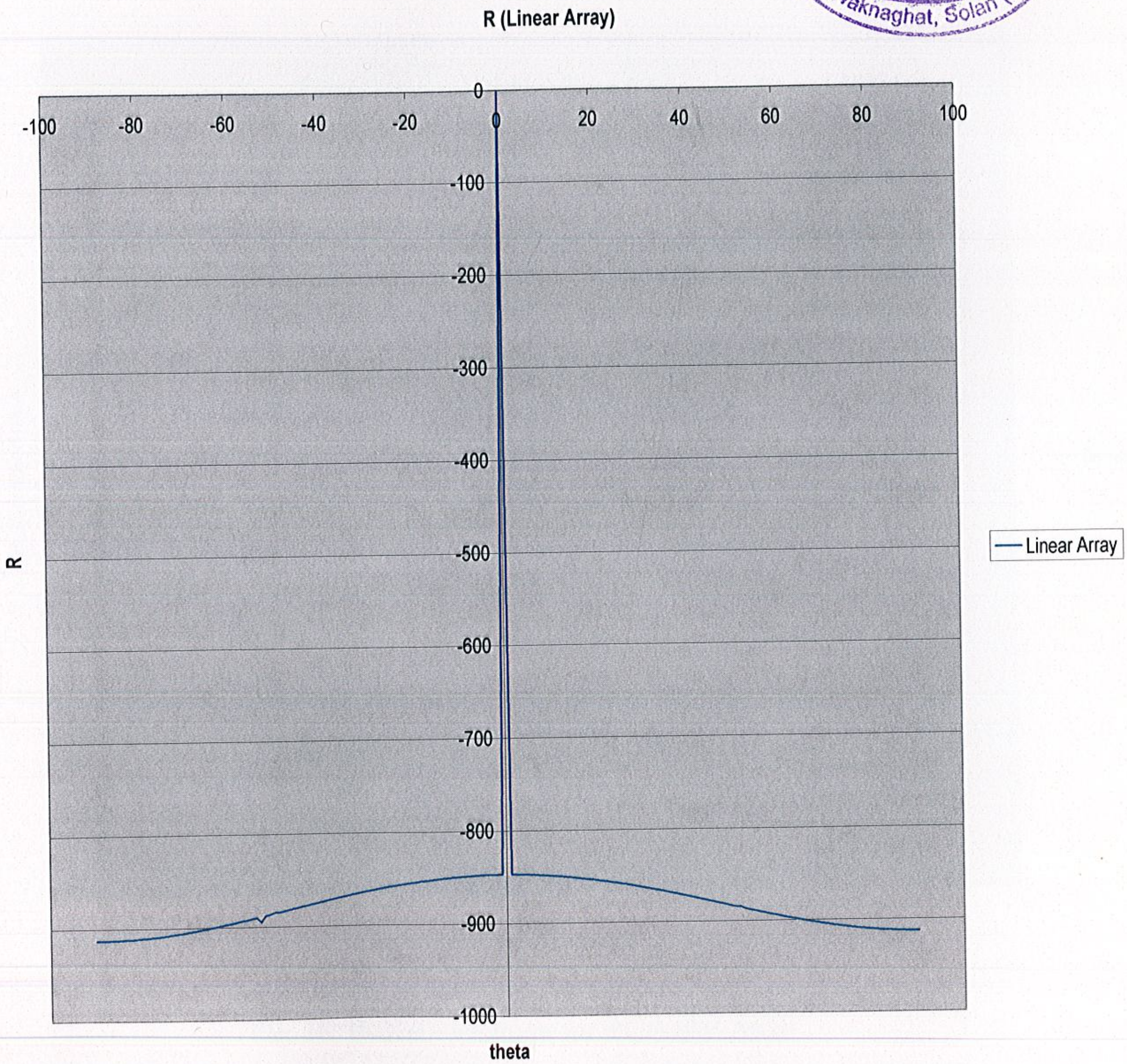


Fig. 14: Radiation Pattern for a Linear Array

The Circular Array is the second type of configuration that can be simulated by the software presented. The following graph displays the results for the simulation of a circular array with the following values input by the user.

- Number of elements in the circular array = 8
- Radius of the circle of array = 15 mm

The graph for the Radiation Pattern of a Circular Array is given below.

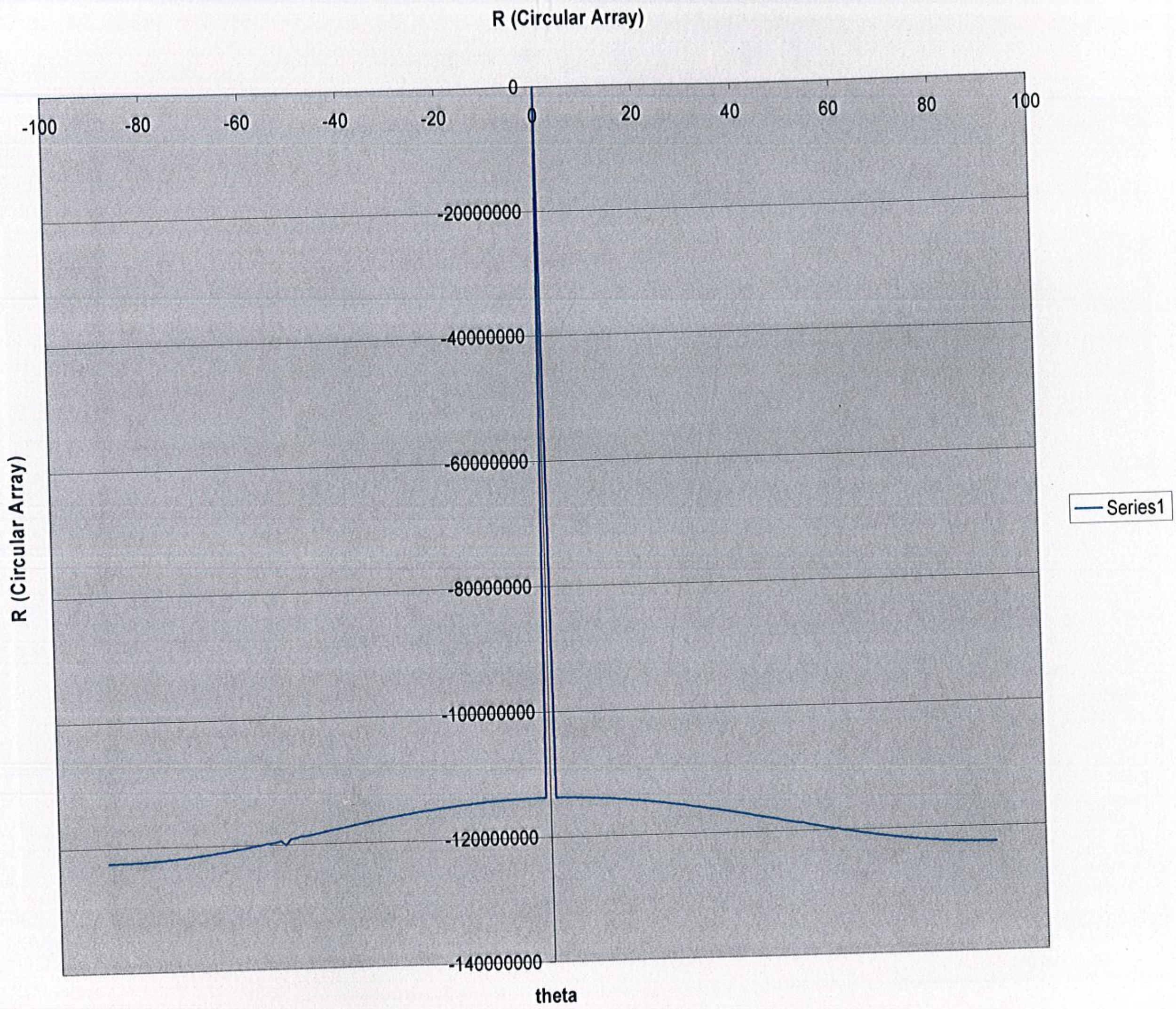


Fig. 15: Radiation Pattern for a Circular Array

Snapshots from the web page are given below:

The following figure is the snapshot of the main index page with a brief introduction of Microstrip Patch Antenna on the right hand side.

The left side lists all links accessible from this page.

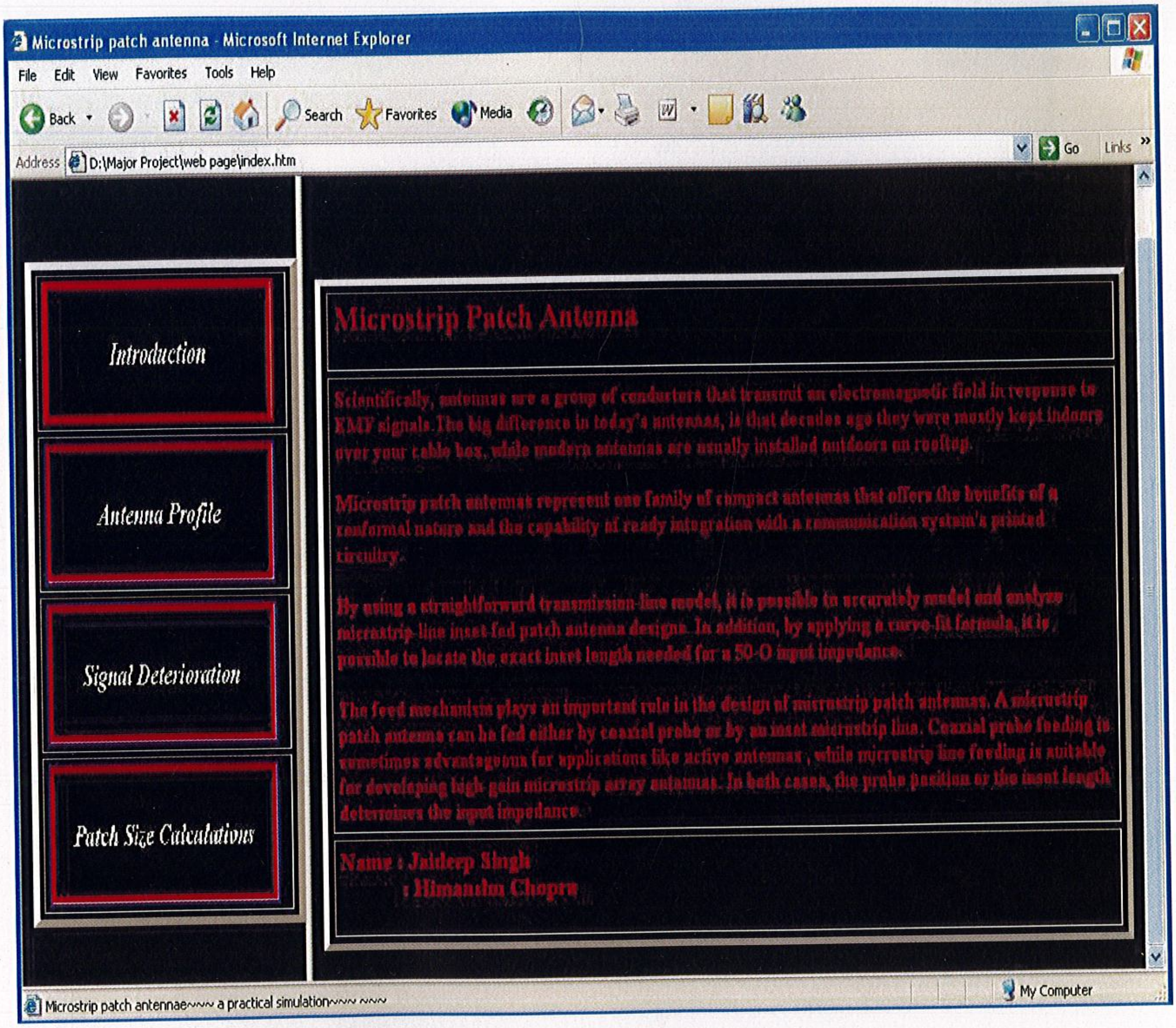


Fig. 16: Main page for the Web Portal

The figure below shows the complete antenna profile of a microstrip patch antenna and discusses the various parameters that deteriorate antenna performance and thus must be carefully considered.

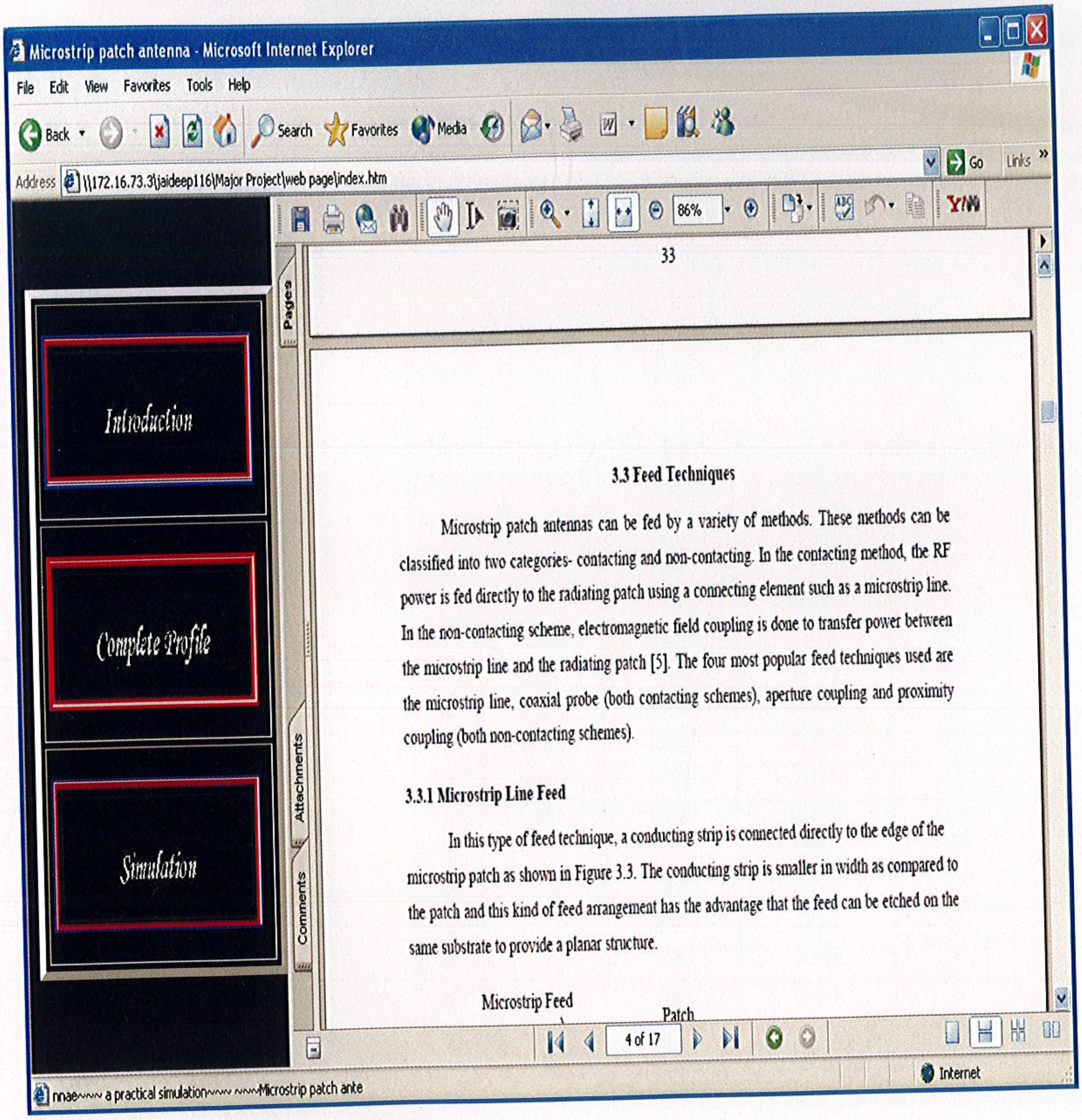


Fig. 17: Profile of a Microstrip Patch Antenna

The Following figure is the snapshot of the web page displaying the theoretical results for a Rectangular Microstrip Patch Antenna.

For ease for downloading and greater security, an Adobe PDF file format is employed.

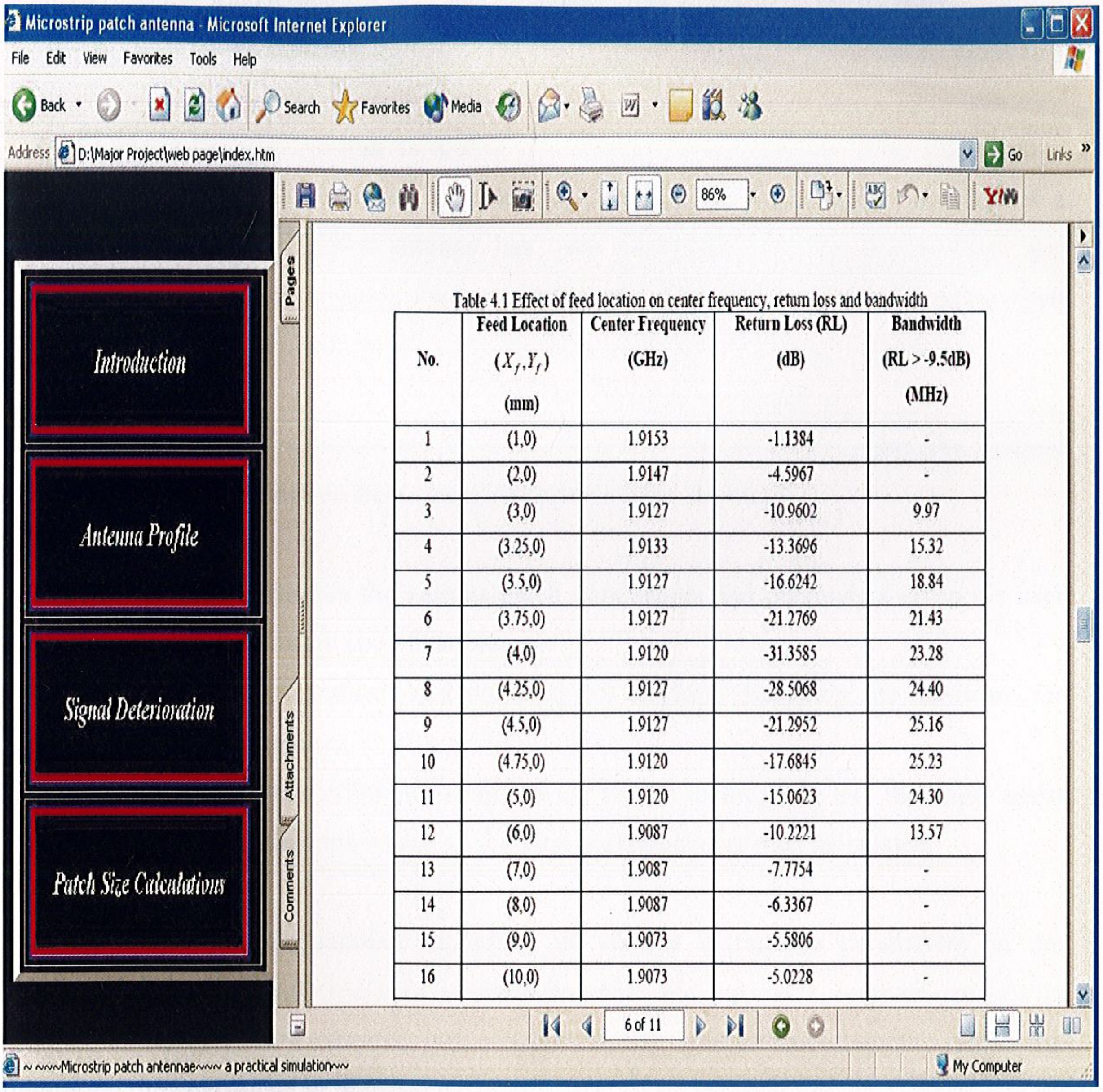


Table 4.1 Effect of feed location on center frequency, return loss and bandwidth

No.	Feed Location (X_f, Y_f) (mm)	Center Frequency (GHz)	Return Loss (RL) (dB)	Bandwidth (RL > -9.5dB) (MHz)
1	(1,0)	1.9153	-1.1384	-
2	(2,0)	1.9147	-4.5967	-
3	(3,0)	1.9127	-10.9602	9.97
4	(3.25,0)	1.9133	-13.3696	15.32
5	(3.5,0)	1.9127	-16.6242	18.84
6	(3.75,0)	1.9127	-21.2769	21.43
7	(4,0)	1.9120	-31.3585	23.28
8	(4.25,0)	1.9127	-28.5068	24.40
9	(4.5,0)	1.9127	-21.2952	25.16
10	(4.75,0)	1.9120	-17.6845	25.23
11	(5,0)	1.9120	-15.0623	24.30
12	(6,0)	1.9087	-10.2221	13.57
13	(7,0)	1.9087	-7.7754	-
14	(8,0)	1.9087	-6.3367	-
15	(9,0)	1.9073	-5.5806	-
16	(10,0)	1.9073	-5.0228	-

Fig. 18: Results for Microstrip Patch Antenna

3.2 CONCLUSION

The technology employing miniature and light weight antenna operating at microwave and millimeter wavelengths has made tremendous growth and progress recently.
This has propelled excellent developments in the area of printed circuit antenna circuits.

These advances have ushered in a new era in system applications fields like satellite communication, missiles, radars, and satellite radiometric imaging. A simulation of a rectangular microstrip patch antenna has been presented. The vital parameters and performance parameters have been investigated. The results obtained have been compared to the theoretical results.

The radiation pattern is plotted and shown in figures 11 and 12 above.. Also, the far-field EM and EA mode radiation field components have been derived.

The present work identifies the various patch dimensions and parameters given the user constraints and user design specifications.
This is very useful for educational purposes for students designing the hardware for microstrip patch antennae.
The presented simulator efficiently displays the results to the user, and thus may assist the fabrication of an antenna where all the vital parameters are well calculated.

It also helps in understanding the effect of various extraneous parameters on the performance of a single rectangular patch antenna. For e.g., if a compact antenna is required, one may use a substrate with a high dielectric constant and attempt to reduce the physical dimensions of an antenna.
If, however, an antenna with a high performance is required, one may opt for a thick dielectric substrate with a low dielectric constant.

3.3 FUTURE SCOPE

Microstrip Patch antennas are used extensively in mobile as well as satellite communications. These antennas can be mounted on spacecraft. These antennas encounter ionized medium in space. In short, Microstrip Patch Antennas work in diverse atmospheres. The present study is carried out on Rectangular Patches only. This work could be extended to all regular as well as irregular shapes. The simulation work can also be done for radiations in ionized medium.

This extension of work is possible only by development of computer software for mathematical simplifications using super fast digital computers.

Method of moments can also be used for non standard shapes of antennas or other irregular contours. All these software developments are a step towards the designing of highly efficient directive antennas.

The scope for further refinement and improvement of the presented work exists both in the theoretical as well as the practical domain.

From the point of view of this simulation, file handling may be incorporated into the code. This would help in storing the results for future references. Also, hard copies may be printed out and supplied as data sheets with manufacturer's product.

For greater user interactivity, the graphics presented may be more refined.

BIBLIOGRAPHY

[1] Chatterjee, Rajeswari *Antenna theory and Practice*, Second Edition, New Age International (P) Limited 1994

[2] Kraus, John D. *Antennas*, Second Edition, McGraw Hill Book Company 1998

[3] Khah, Sunil K. *Study of radiation characteristics of Microstrip Patch Antenna* , a dissertation for Master of Philosophy 1994

[4] Bahl I J, Stuchly S S and Bhartia P, *Design of Microstrip Antennas Covered with Dielectric Layer*, IEEE paper, Vol. AP-30, No. 2 1982

[5] Gupta K C and Benella A, *Microstrip Antenna Design*, Reprint Volume, Artech House, Norwood M A 1988

[6] Bhall J and Bhartia P *Microstrip Antennas*, Artech House, Dedham (USA) 1980

[7] www.taconic-add.com/pdf/technicalarticles--microstrip_patch_antenna_for_gps.pdf

APPENDICES

Appendix A

The source code for the designing of a rectangular Microstrip Patch Antenna is given here. The various parameters for the physical dimensions are calculated here.

```
int ph1,th1;
double t1,t2,t3,t4;
double f0,er,h,w,ereff,leff,del_l,l,lg,wg;
double R_fin[181]={0.0};
double del_l_temp1=0,del_l_temp2=0;
const double c = 300000000.00;
char *introduction ="This program simulates the designing of a microstrip patch
antenna";
double E_theta[181],E_phi[181],k2=0,k0=0,lamda0=0,R[181];
double theta[181]={0.0};
double AF_linear,AF_circular;
int flag=0,phi=0,phi_count=0,z0 = 378;
float temp;
int count=0;


void input_param()
{
 clrscr();
 do
  {
```

```
cout<<introduction[count];
count++;
delay(rand()%10);
}while(introduction[count]);
cout<<endl<<endl;
//operating frequency
cout<<"Enter the Value of Operating Frequency (in GHz) : ";
cin>>f0;
f0=f0*pow(10,9); //frequency of operation now in Hz
cout<<endl;


//relative permittivity of the medium
cout<<"\n\nEnter the value of Relative permittivity of the medium : ";
cin>>er;
cout<<endl;


//height of the dielectric
cout<<"\n\nEnter the value of Height of Dielectric (in mm): ";
cin>>h;
cout<<endl;


//width of the patch, c defined const
w=(c/(2*f0))*sqrt((2/(er+1)));
w=w*1000; //required to convert w into mm (as above result in m)


//result calculation interactivity with user
cout<<endl<<"Calculating result";
```

```
for(count=0;count<5;count++)
 {
   cout<<".";
   delay(rand()%500);
 }
cout<<endl<<endl;

//displaying results
 cout<<"\n The width of the patch is : "<<w/1000<<" m = "<<w<<" mm";

//effective permittivity of the dielectric  ereff
 ereff=((er+1)/2)+((er-1)/2)*(1/sqrt(1+((12*h)/w)));  // w and h in mm
 cout<<endl<<endl;
 cout<<"\n The effective permittivity of the Dielectric is : "<<ereff;

//effective length of the patch leff
 leff = c/(2*f0*sqrt(er));    //er placed here replacing er_eff
 leff = leff * 1000;    //leff in mm here
 cout<<endl<<endl;
 cout<<"\n The effective length of the patch is :  "<<leff/1000<<" m = "<<leff<<" mm";


//length extension change del_l
 del_l_temp1 = (ereff +.3) * ( (w/h) + .264);
 del_l_temp2 = (ereff - .258) * ( (w/h) + .8);
 del_l=0.412*h*(del_l_temp1/del_l_temp2);// del_l in mm
```

```
cout<<endl<<endl;
cout<<"\n The length extension for the patch is : "<<del_l/1000<<" m = "<<del_l<<"
mm";


//actual length of l
 l=leff-(2*del_l);  // l , lg, wg all in mm
 lg=( 6*h) + l;
 wg=( 6*h) + w;
 cout<<endl<<endl;
 cout<<"\n The actual length of patch is :   "<<abs(l/1000)<<" m ="<<abs(l)<<" mm";
 cout<<"\n Ground Plane location: "<<endl<<"\t\t";
 cout<<"Length(g) :    "<<lg/1000<<" m"<<" = "<<lg<<" mm";
 cout<<"\n\t\tWidth(g)  :    "<<wg/1000<<" m"<<" = "<<wg<<" mm";
 cout<<endl<<endl<<"Press any key to continue";
 for(count=0;count<5;count++)
  {
    cout<<".";
    delay(rand()%500);
  }
 getch();
 clrscr();

 }
```

APPENDIX B

The following code illustrates the implementation of the far field expressions for a Rectangular Microstrip Patch Antenna.

The radiation pattern components other basic parameters are calculated.

```
int ph1,th1;
double t1,t2,t3,t4;
double f0,er,h,w,ereff,leff,del_l,l,lg,wg;
double R_fin[181]={0.0};
double del_l_temp1=0,del_l_temp2=0;
const double c = 300000000.00;
char *introduction ="This program simulates the designing of a microstrip patch
antenna";
double E_theta[181],E_phi[181],k2=0,k0=0,lamda0=0,R[181];
double theta[181]={0.0};
double AF_linear,AF_circular;
int flag=0,phi=0,phi_count=0,z0 = 378;
float temp;
int count=0;

int far_field_exp()
{
 //far field expressions
 lamda0 = c/f0; // lambda in m thus converting to mm in next line
 lamda0 = lamda0 * 1000;// lamda now in mm
 k0 = (2*3.1412/lamda0);//k0 in hz
 k2 = ((4*h)/(lamda0 * z0));//h in mm as i/p by user
 temp = (k0*w)/2;
```

```
//initialising theta
 double j= -90;
 for(count=0;count<181;count++)
   theta[count] = j++;
//theta[181] has elements -90 -89 ...upto +90

 for(phi=0;phi<=90;phi=(phi+90) )
   {
    cout<<endl<<"Generating Data for phi = "<<phi<<"degrees"<<endl;


    for(int counter=0;counter<181;counter++)
        {
        if(theta[counter] == 0)
          {
            E_theta[counter] = k2;
            E_phi[counter] = k2;
            R[counter] = ( pow(k2,2) + pow(k2,2));
          }
        else
          {


            if(phi == 0)
                {
                  E_phi[counter] = 0;
                  t1 = cos(temp * sin((3.14/180)*theta[counter]));//converting angle into
radians
                  t2 = cos((3.14/180)*temp * sin(theta[counter]*(3.14/180)));
                  E_theta[counter] = k2 * t1 * t2;
```

```
                R[counter] = (pow(E_theta[counter],2) + pow(E_phi[counter],2));
                R[counter] /= 100;
            }
        if(phi == 90 )
            {
             E_theta[counter] = 0;
             t1 = cos(temp * sin((3.14/180)*theta[counter]));
             t3 = sin(k0 * h * sin((3.14/180)*theta[counter]))/2; // this eqn
             t4 = (k0*h*sin((3.14/180)*theta[counter]))/2;
             E_phi[counter] = (k2 * t1 * t3 * c
    cos((3.14/180)*theta[counter])*cos((3.14/180)*theta[counter])) / t4;
             R[counter] = (pow(E_theta[counter],2) + pow(E_phi[counter],2));
            }
        }

}

t1 = R[0];
for(int p=1;p<181;p++)
        if(R[p] >t1)
              t1 = R[p];

cout<<endl<<"max value of R is : "<<t1;

for(p=0;p<181;p++)
 {
      R_fin[p] = 10 * log( R[p]/t1);
 }

draw_graph(R_fin,phi);
```

```
}

 int ch;
  ch=1;
  while(ch==1)
   {
    cout<<"\nEnter the value of gap of phi for which power shud be computed :";
    cin>>ph1;
    if((ph1<=90)&&(ph1>0))
     {
     ch=0;
     }
    else
     {
          cout<<"\nEntered Value is incorrect .....";
          cout<<"\nPlz Enter value less than equal to 90 and greater than 0";
     }
    }

  ch=1;

  while(ch==1)
   {
    cout<<"\nEnter the value of gap of theta for which power shud be computed :";
    cin>>th1;
    if((th1<=180)&&(th1>0))
     {
     ch=0;
     }
    else
     {
```

```
            cout<<"\nEntered Value is incorrect .....";
            cout<<"\nPlz Enter value less than equal to 180 and greater than 0";
        }
       }


  power();
  restorecrtmode();

  return 0;
}

int draw_graph(double R_fin[181], int phi)
{

 int gd=DETECT,gm;
 int counter11=0;
 int org_x=20,org_y=200;
 double max_R_fin=0,min_R_fin=0;


 max_R_fin = R_fin[0];
 min_R_fin = R_fin[0];
 for(counter11=0;counter11<180;counter11++)
  {
   //computing max value of R_fin
   if(R_fin[counter11] > max_R_fin)
        max_R_fin = R_fin[counter11];

   //computing min value of R_fin
   if(R_fin[counter11] <min_R_fin)
```

```
        min_R_fin = R_fin[counter11];
    }
cout<<endl<<"Max Value of R_fin is "<<max_R_fin;
cout<<endl<<"Min value of R_fin is "<<min_R_fin;

initgraph(&gd,&gm,"C:\\tc\\bgi");
//drawing X axis
line(org_x,org_y+200 , org_x+600,org_y+200);
//drawing arrowheads on X axis
line(org_x+600,org_y+200,org_x+600-3,org_y-2+200);
line(org_x+600,org_y+200,org_x+600-3,org_y+2+200);

// writing the values on axis

outtextxy(org_x,org_y+202,"-90");
outtextxy(org_x+30,org_y+202,"-80");
outtextxy(org_x+60,org_y+202,"-70");
outtextxy(org_x+90,org_y+202,"-60");
outtextxy(org_x+120,org_y+202,"-50");
outtextxy(org_x+150,org_y+202,"-40");
outtextxy(org_x+180,org_y+202,"-30");
outtextxy(org_x+210,org_y+202,"-20");
outtextxy(org_x+240,org_y+202,"-10");

outtextxy(org_x+272,org_y+202,"0");

outtextxy(org_x+300,org_y+202,"10");
outtextxy(org_x+330,org_y+202,"20");
outtextxy(org_x+360,org_y+202,"30");
outtextxy(org_x+390,org_y+202,"40");
outtextxy(org_x+420,org_y+202,"50");
```

```
outtextxy(org_x+450,org_y+202,"60");
outtextxy(org_x+480,org_y+202,"70");
outtextxy(org_x+510,org_y+202,"80");
outtextxy(org_x+540,org_y+202,"90");



//drawing Y axis
line(org_x+270,org_y-160,org_x+270,org_y+260);
//drawing arrowheads on Y axis
line(org_x,org_y-460,org_x-2,org_y+3-460);
line(org_x,org_y-460,org_x+2,org_y+3-460);

outtextxy(org_x+270,org_y-160,"0");
outtextxy(org_x+270,org_y-150,"-50");
outtextxy(org_x+270,org_y-100,"-100");
outtextxy(org_x+270,org_y-50,"-150");
outtextxy(org_x+270,org_y,"-200");
outtextxy(org_x+270,org_y+50,"-250");
outtextxy(org_x+270,org_y+100,"-300");
outtextxy(org_x+270,org_y+150,"-350");
outtextxy(org_x+272,org_y+202,"-360");



// graph plot code goes here

int a,b,c;


if(phi==0)
```

```
 {
  a = org_x;
  for(int i=0;i<=180;i++)
 {
  b=0;
  b=int (R_fin[i]);
  b=(org_y-160)-b;

  putpixel(a,b,RED);

  a=a+3;
  delay(25);

  }
   }
else
 {
a=org_x;
for(int i=0;i<=180;i++)
 {
  b=0;
  b=int (R_fin[i]);
  b=(org_y-160)-b;
  putpixel(a,b,RED);
  a=a+3;
  delay(25);
 }
getch();
}
// restorecrtmode();
return 0;
```

```
}
```

APPENDIX C

The following code snippet illustrates the calculation of array factors for the following type of arrays:

- Linear Arrays
- Circular Arrays

The radiation pattern for the array configuration, the power radiated and the directivity of the antenna are computed. The variables used are the same as defined in appendix A and appendix B.

```
int array_elements()
 {
 int flag;
 int no_of_elements;
 double distance;
 int choice;
 double beta0 = (2 * 3.1412)/lamda0;
 double beta = (beta0/sqrt(ereff) );

 int gd=DETECT,gm;
 cout<<endl<<"Switching to arraying of microstrip antenna";
 for(count=0;count<5;count++)
   {
    cout<<".";
    delay(rand()%500);
    sound(2000000);
    nosound();
    sound(2000000);
```

```
   }

clrscr();
cleardevice();

initgraph(&gd,&gm,"C:\\tc\\bgi");

int x,y;
int counter;
clrscr();
setbkcolor(BLACK);
x=getmaxx();
y=getmaxy();

// outermost rectangle
setcolor(RED);
for(counter=0;counter<10;counter++)
 {
  rectangle(0+counter,0+counter,(x-counter),(y-counter));
  delay(100);
 }

flag = 0;
 // button 1 for Linear Array
setcolor(RED);
x=90;
y=200;
for(counter=0;counter<16;counter++)
 {
  rectangle(x+counter,y+counter,x+140-counter ,y+30-counter);
 }
```

```
setcolor(BLUE);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,2);
outtextxy(95,200,"Linear Array");


//button 2 for Circular Array
x=90;
y=250;
setcolor(RED);
for(counter=0;counter<16;counter++)
 {
  rectangle(x+counter,y+counter,x+140-counter,y+30- counter);
 }

setcolor(BLUE);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,2);
outtextxy(95,250,"Circular Array");


  union REGS i,o;
   int button;
   i.x.ax=0;
   int86(0x33,&i,&o);
   if(o.x.ax==0)
    {
    cout<<endl<<"No Mouse Device Detected.....";
     exit(7);
    }
   i.x.ax=1;
   int86(0x33,&i,&o);
```

```
  gotoxy(24,23);

for(;;)
{
char ch1;
if(kbhit())
 {
  ch1=getch();
  if(ch1==32)
   {
    clrscr();
    cleardevice();
   }
  if(ch1==27)
   {
        exit(0);
   }
 }


i.x.ax=3;
int86(0x33,&i,&o);
button=o.x.bx&7;
gotoxy(23,11);
i.x.ax=3;

// for simulation
if((90<o.x.cx)&&(230>o.x.cx)) // FOR SETTING X-AXIS OF MOUSE POINTER
 {
   if((200<o.x.dx)&&(230>o.x.dx)) // FOR SETTING Y-AXIS OF MOUSE POINTER
```

```
{
 if(button==1) // LEFT CLICK OF MOUSE
  {
   clrscr();  // CHANGING THE SCREEN
   cleardevice();
   restorecrtmode();
    do
        {
         cout<<endl<<"Enter No. of array_elements : ";
         cin>>no_of_elements;
         if(no_of_elements < 0)
           cout<<endl<<"\tNo of elements can't be negative.Please enter again...";
        }while(no_of_elements<0);
    do
        {
         cout<<endl<<"Enter the distance b/n consecutive elements (in mm): ";
         cin>>distance;
         if(distance< 0)
           cout<<endl<<"Distance can't be negative.Please enter again...";
        }while(distance<0);


    t1 = cos(beta0 * distance * cos(3.1412/2) + beta);
    t2 = cos(beta0 * beta * cos(3.1412/2) /2);
    AF_linear = no_of_elements * t1 * t2;

   flag = 1;

  }
 }
}
```

```
// for Circular
if((90<o.x.cx)&&(230>o.x.cx)) // FOR SETTING X-AXIS OF MOUSE POINTER
 {
 if((250<o.x.dx)&&(280>o.x.dx)) // FOR SETTING Y-AXIS OF MOUSE POINTER
  {
     if(button==1) // LEFT CLICK OF MOUSE
      {
      clrscr();  // CHANGING THE SCREEN
      cleardevice();
        AF_circular =0;

        do
           {
           cout<<endl<<"Enter No. of array_elements : ";
           cin>>no_of_elements;
           if(no_of_elements < 0)
            cout<<endl<<"\tNo. of elements can't be negative. Please enter again...";
           }while(no_of_elements<0);

        for(int counter=0;counter<181;counter++)
           {
            t1 = beta0 * sin( (3.1412/180) * (counter -90) );
            t2 = cos((3.1412 * 0)/180);
            t3 = cos( t1 * t2) + beta;
            AF_circular = AF_circular + t3;
            cout<<"\nAF_circular for theta = "<<counter-90<<" = "<<AF_circular;
          }
       flag = 1;
```

```
        }
      }
    }

  if(flag==1)
  {
  restorecrtmode();



  clrscr();
  cout<<endl<<"Displaying Radiation Pattern for Linear Array";
  for (int p=0;p<181;p++)
   {
    t1 = pow(E_theta[p],2) + pow(E_phi[p],2) ;
    t2 = 2 * AF_linear *(E_theta[p] + E_phi[p]);
    t3 = 2 * pow(AF_linear,2);
    R_fin[p] = t3+t2+t1;
    cout<<endl<<R[p];

   }
   getch();
   exit(0);

  }
 }
 return 0;
}
```