

**DESIGN OF
RECTANGULAR PATCH ANTENNA ARRAY
FOR 5G WIRELESS COMMUNICATION**

Semester-V

Antennas and Wave Propagation

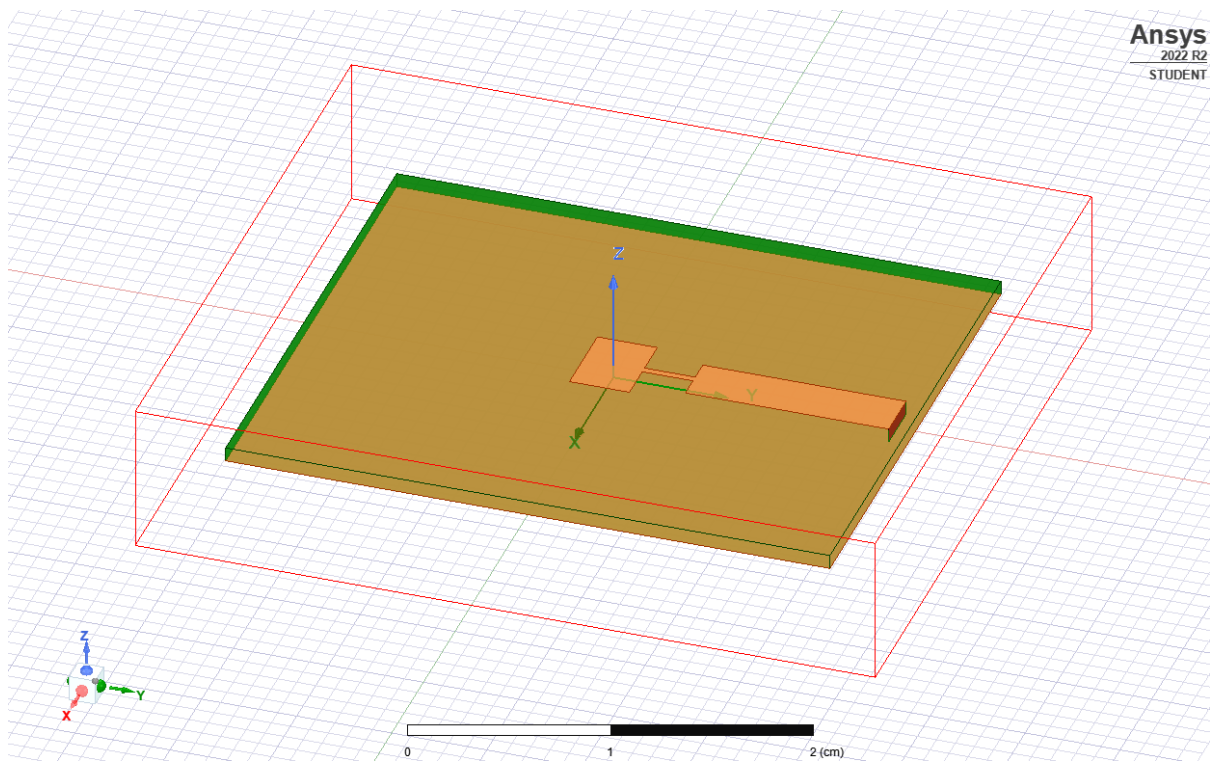
Poorvajha G

2020105562

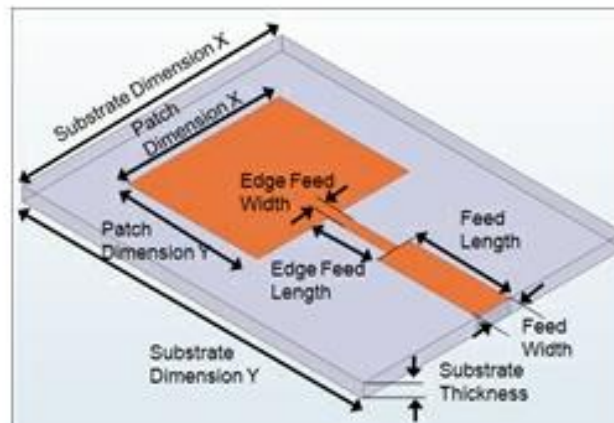
Aim:

We have planned to design a two-element array antenna for 5G wireless communication using HFSS software. The antenna operating frequency is chosen as 28GHz. Our final design must contain two rectangular patches on a Roger RT-5880 LZ, dielectric constant $\epsilon_r = 1.96$. Till now we have implemented a single patch edge-fed antenna with Koch snowflake fractal.

Design Image:



Dimensions:



Patch:

Patch width $X = 0.4403$ cm

Patch length $Y = 0.31535$ cm

Substrate:

Substrate material: Rogers RT-5880

Dielectric constant: 1.96

Substrate thickness: 0.0762 cm

Substrate width X : 2.7 cm

Substrate length Y : 3.18 cm

Feed:

Edge feed width: 0.0366 cm

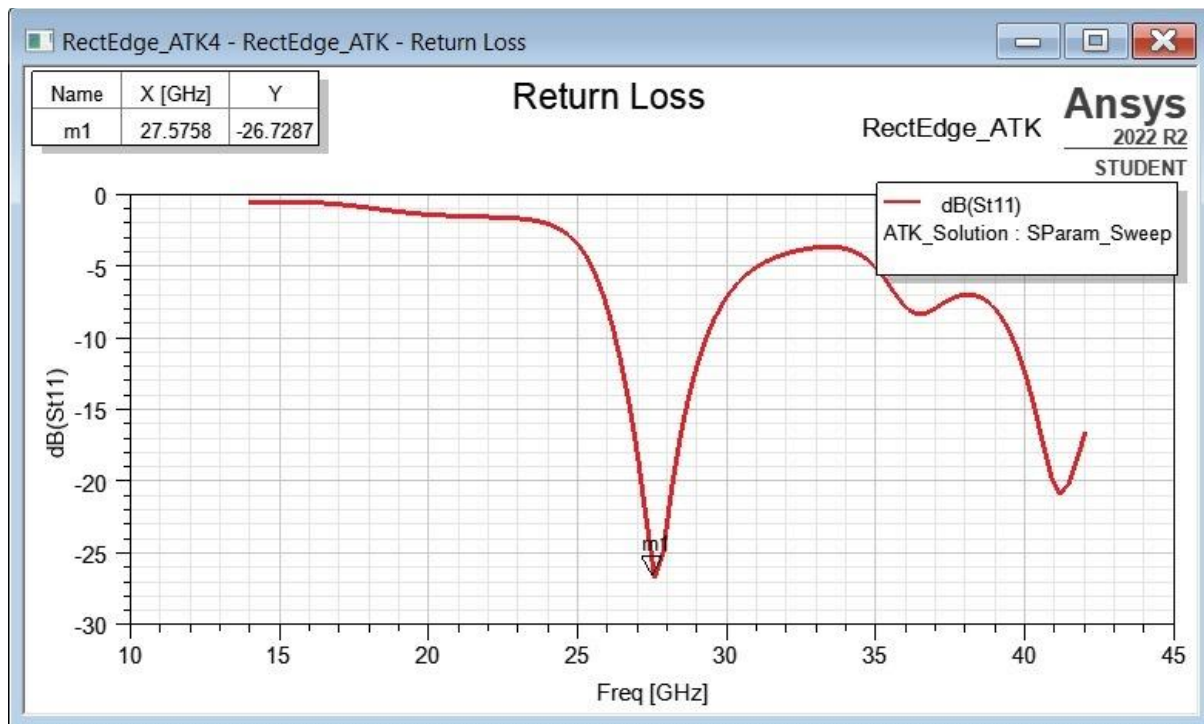
Edge feed length: 0.267 cm

Feed Width: 0.2738 cm

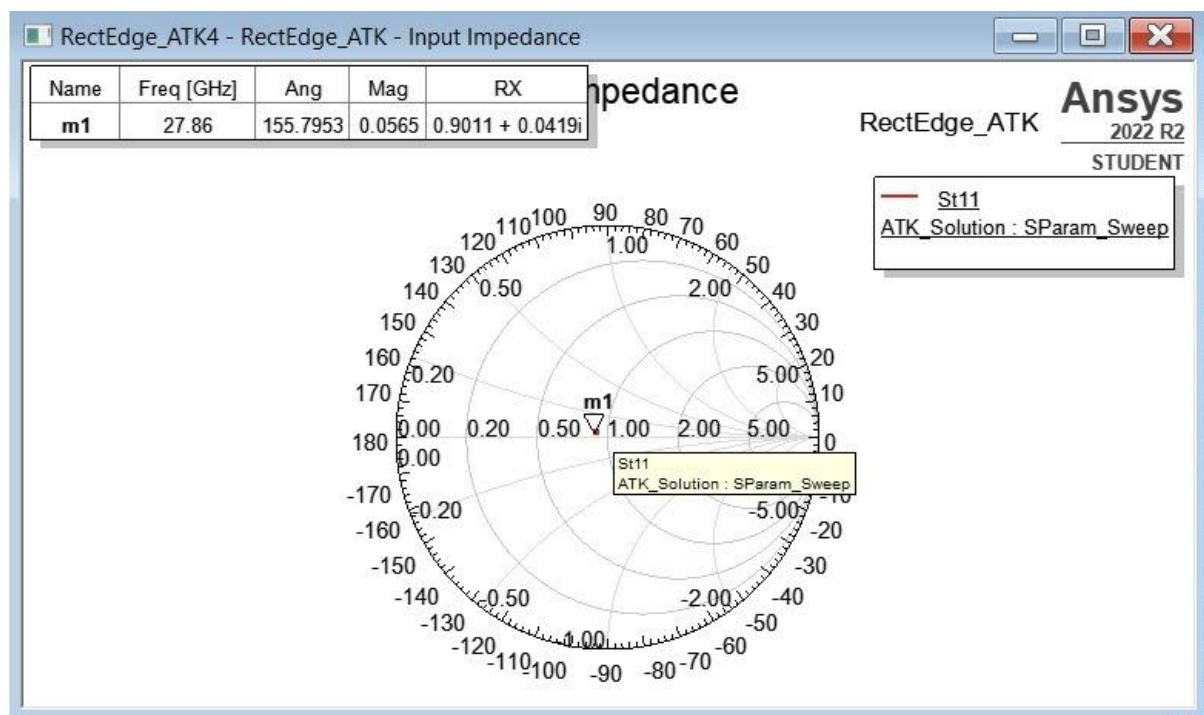
Feed length: 1.07 cm

Frequency: 28GHz

Results:



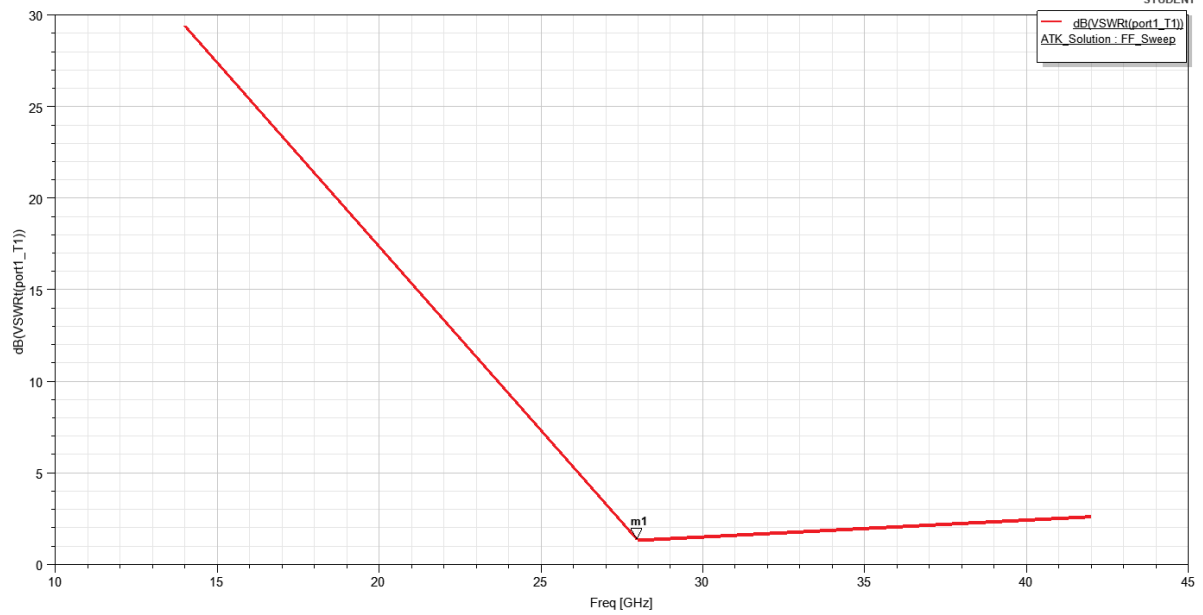
Input impedance:



Name	X [GHz]	Y
m1	28.0000	1.3411

Terminal VSWR Plot 1

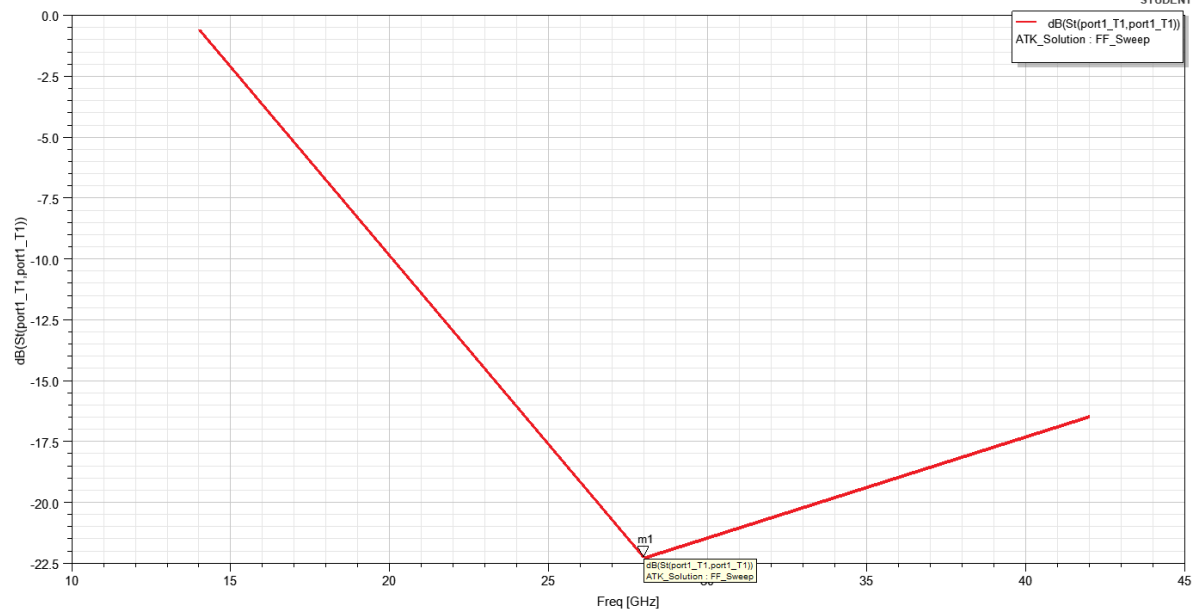
RectEdge_ATK
2022 R2
STUDENT



Name	X [GHz]	Y
m1	28.0000	-22.2649

Terminal S Parameter Plot 1

RectEdge_ATK
2022 R2
STUDENT



	Freq [GHz]	dB(PeakDirectivity) ATK_Solution : FF_Sweep
1	14.000000	3.484804
2	28.000000	9.596694
3	42.000000	8.757609

We are still working in Koch snowflake fractal.

The Koch snowflake is introduced in the patch by using a python code which is given as follows:

```
# -----
# Script Recorded by ANSYS Electronics Desktop Version 2020.1.0
# 19:45:19 Apr 23, 2020
# -----
import ScriptEnv
from math import sin, cos, pi
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.GetActiveProject()
oDesign = oProject.GetActiveDesign()
oEditor = oDesign.SetActiveEditor("3D Modeler")

r=1 #radius
unit='mm' #unit
n=4 #iteration

models=[]
def triangle(x,y,r):
    name=oEditor.CreateRegularPolygon(
        [
            "NAME:RegularPolygonParameters",
            "IsCovered:=", True,
            "XCenter:=", "{}{}".format(x, unit),
            "YCenter:=", "{}{}".format(y, unit),
            "ZCenter:=", "0mm",
            "XStart:=", "{}{}".format(x+r, unit),
            "YStart:=", "{}{}".format(y, unit),
            "ZStart:=", "0mm",
            "NumSides:=", "3",
            "WhichAxis:=", "Z"
        ],
        [
            "NAME:Attributes",
            "Name:=", "Polygon1",
            "Flags:=", "",
            "Color:=", "(255 128 0)",
            "Transparency:=", 0,
            "PartCoordinateSystem:=", "Global",
            "UDMId:=", "",
            "MaterialValue:=", "\"copper\"",
            "SurfaceMaterialValue:=", "\"\"",
            "SolveInside:=", True,
            "IsMaterialEditable:=", True,
            "UseMaterialAppearance:=", False,
            "IsLightweight:=", False
        ])
    return name

def fractal(n, x, y, r):
```

```

global models
if n == 0:
    models.append(triangle(x, y, r))
else:
    fractal(n-1, x-r*cos(0*pi/3), y-r*sin(0*pi/3), r/2)
    fractal(n-1, x-r*cos(2*pi/3), y-r*sin(2*pi/3), r/2)
    fractal(n-1, x-r*cos(4*pi/3), y-r*sin(4*pi/3), r/2)
    models.append(triangle(x, y, r))

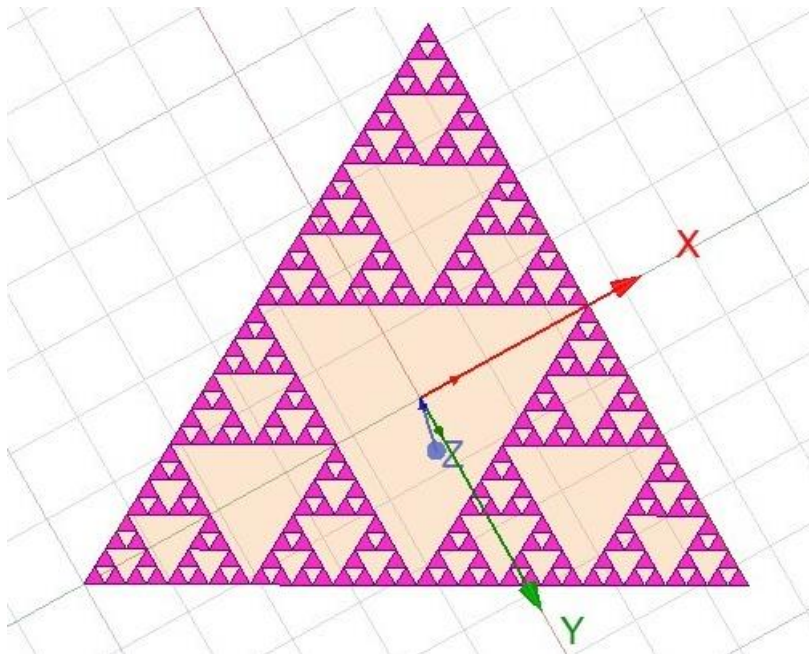
triangle(0,0,-r)
fractal(n,0,0,r/2)

tool = ','.join(models)
AddWarningMessage(tool)

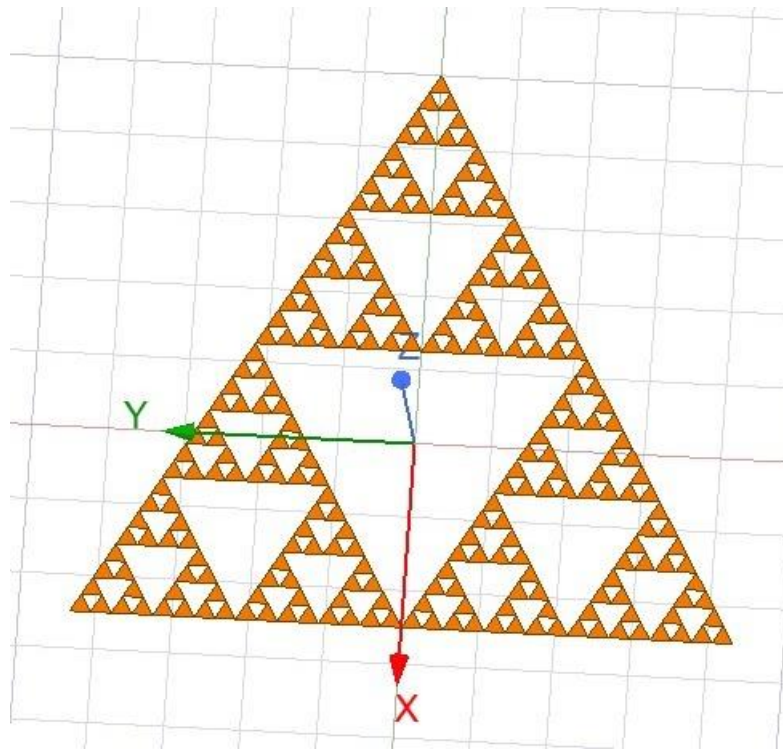
oEditor.Subtract(
    [
        "NAME:Selections",
        "Blank Parts:=", "Polygon1",
        "Tool Parts:=", tool
    ],
    [
        "NAME:SubtractParameters",
        "KeepOriginals:=", False
    ])

```

The Koch fractal in the patch is implemented as follows:



Iteration-3



Iteration-4

We are yet to add the substrate, ground and feed.