# Designing And Analysis Using CST Studio

Name: SARVAGYA SANJAY

## Description of Antenna Array Design:

7 dipole antennas were made of length 143mm with feed gap of 20mm with gap between them being lambda/2 = 300mm.

Outer radius = 4mm

Centre frequency = 1GHz

Type of antenna: dipole

centre freq: 1Ghz

number of elements: 7

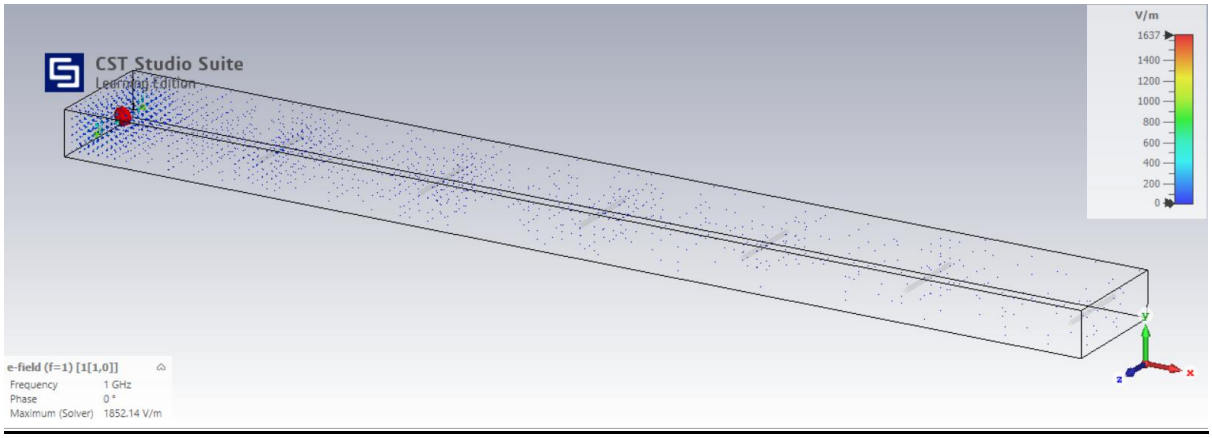
feed gap: 20mm

element spcing: lambda/2

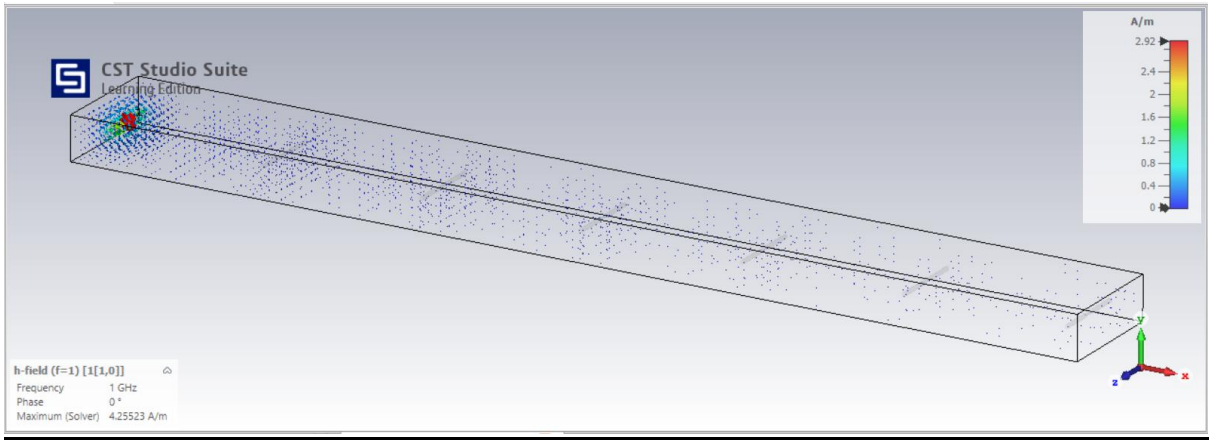
The dipole material was PEC and gap between the dipole is vacuum.

## Simulated radiation patterns (2D and 3D) and gain plots. E field, H field, far field, S parameter, VSWR:

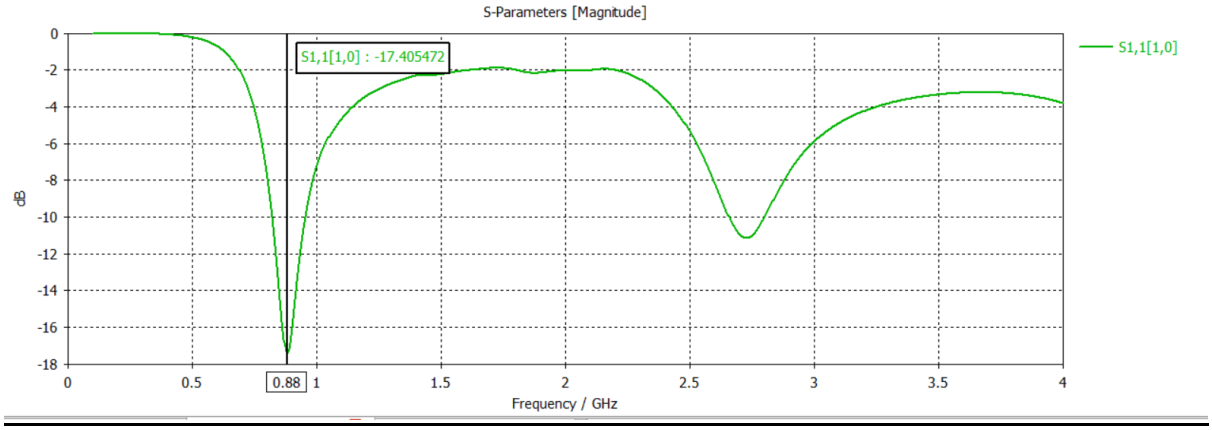
## E-field:



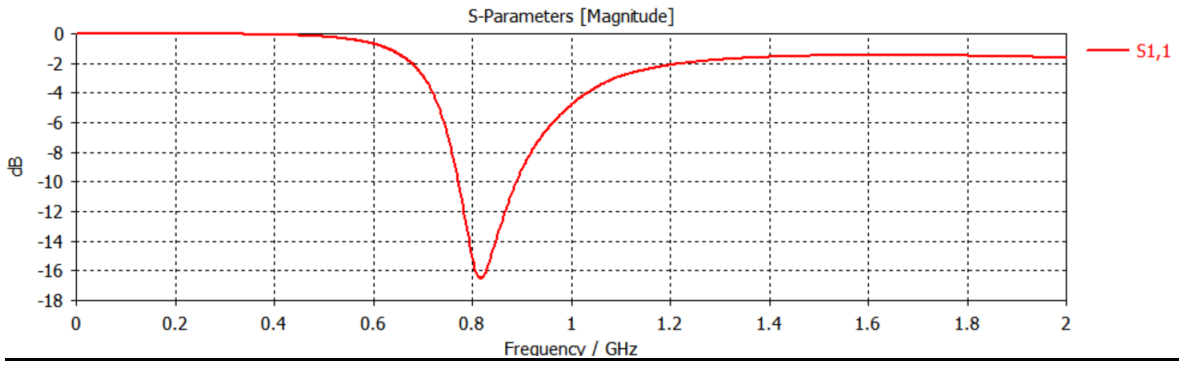

## H-field:



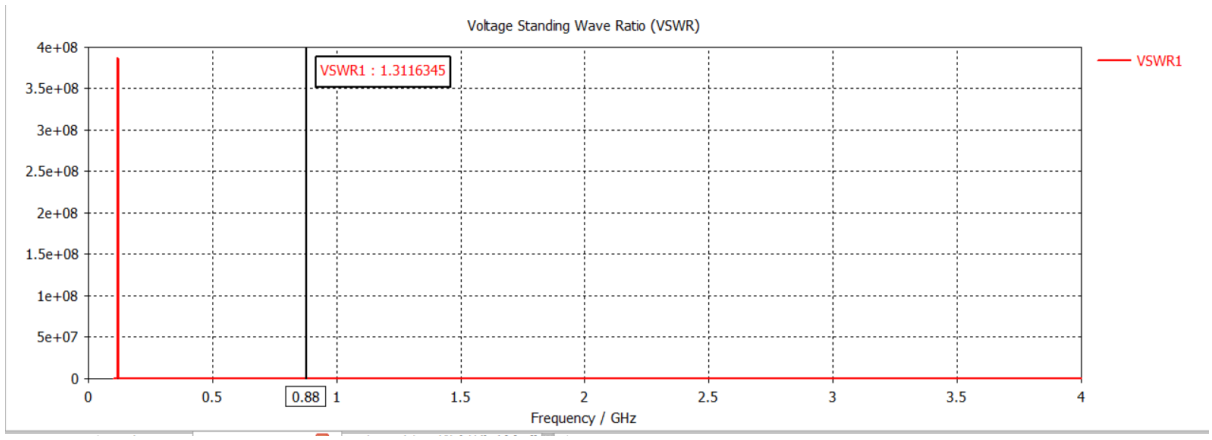

## S-parameter:



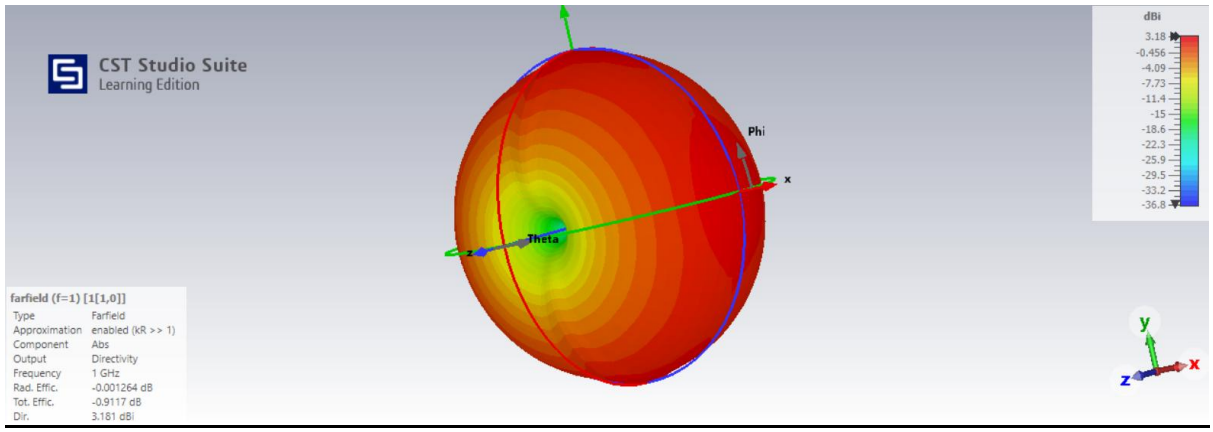
## S[1,1]




## VSWR:




## Farfield:



## Report the gain, directivity, and impedance matching characteristics.

**Gain:** The dipole array exhibits higher gain than a single dipole, typically ranging from 6 dBi (2-element) to 12+ dBi (8-element).

**Directivity**: Increases with more elements; a 4-element dipole array can achieve around **10 dBi**, meaning most energy is radiated in a specific direction.

Directivity for this 7 element array is 3.181 dBi.

**Impedance Matching**: The array's input impedance is tuned using techniques like:

- **Optimization of element spacing and feedline impedance**

  Element spacing is lambda/2 = 300mm and feedline impedance is 50 ohm.

- **S-parameter analysis**, where **S11 < -10 dB** indicates good reflection loss.

## Discuss the optimization process and any one design trade-off parameter and how?

We can increase the length of the dipole antenna to get better performance.

We can also increase the impedance of the discrete port of all the dipole antennas in the array.

Design Trade-Off – Beamwidth vs Gain

One key trade-off in dipole antenna arrays is **beamwidth vs gain**:

- **Higher gain** (more elements) results in a **narrower beamwidth**, reducing coverage area.
- **Wider beamwidth** (fewer elements) improves coverage but reduces gain, making it less effective for long-range applications.
- **Compromise**: Choosing the optimal number of elements based on application needs.