**Experiment-3**

**Aim:-** To implement concept code of the Cluster, Frequency Reuse and Co-Channel Scheme for the cellular Communication.

**Activities:**

1. To implement and execute a MATLAB program to divide the given area into equal hexagons and create clusters.
2. To locate co-channel cells using Activity a clustering method.

**Theory:**

A Cell is the basic geographical unit of a cellular system commonly represented as a hexagon. The term cellular comes from the hexagonal or honeycomb shape of the coverage area. Each cell has a BS transmitting over a cell. Because of constraints imposed by natural terrain and manmade structure. The true shapes of cell are not hexagonal. The coverage area of cell is called the footprint. The BS simultaneously communicates with many mobile using one channel (pair of frequencies) per mobile. One frequency is for the forward link (Base station to the mobile) and other frequency is for the reverse link (Mobile to the Base station). Each cell size varies depending on landscape, subscriber density and demand within particular region. Cells can be added to accommodate growth e.g.; creating new cells by overlaying, splitting, or sectoring existing cells. This technique increases the capacity of the system. Sectoring existing cells and then using directional antenna can also increase capacity.

**Cluster:**

A cluster is a group of cells. No frequencies are reused within a cluster. Frequencies used in one cell cluster can be reused in another cluster of cells. A large number of cells per cluster arrangement reduces interference to the system.

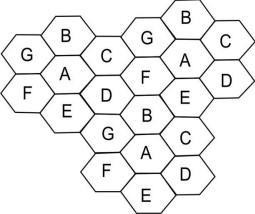
**Frequency Reuse:**

Frequency reuse is a technique of allocating channels to the cellular system. Because of the unavailability of the spectrum at the cellular band, channels frequencies must be reused. Cells are assigned group of channels that are completely different from those of neighboring cells. Cells with the same number have the same set of frequencies, if the number of available frequency is 7, the frequency reuse factor is 1/7, which implies that each cell is using 1/7 of available frequencies. Frequency reuse introduces interference into the system.

In the cellular concept, frequencies allocated to the service are re-used in a regular pattern of areas, called 'cells', each covered by one base station. In mobile-telephone nets these cells are usually hexagonal. In radio broadcasting, a similar concept has been developed based on rhombic cells.

To ensure that the mutual interference between users remains below a harmful level, adjacent cells use different frequencies. In fact, a set of C different frequencies {f1, ..., fC} are used for each cluster of C adjacent cells. Cluster patterns and the corresponding frequencies are re-used in a regular pattern over the entire service area. The total bandwidth for the system is C times the bandwidth occupied by a single cell.

In the practice of cell planning, cells are not hexagonal as in the theoretical studies. Computer methods are being used for optimised planning of base station location and cell frequencies. Pathloss and link budgets are computed from the terrain features and antenna data. This determines to coverage of each base station and interference to other cells.



**Program:**

import numpy as np

import matplotlib.pyplot as plt

# Define hexagon parameters r = 2 # Radius

angles = np.linspace(0, 2 \* np.pi, 7) labels = ['A', 'B', 'C', 'D', 'E', 'F', 'G']

# Create a figure and axis fig, ax = plt.subplots(figsize=(8, 8))

# Iterate to create and label hexagons for i in range(6):

cx = r \* np.sqrt(3) \* np.cos(np.pi/2 + i \* np.pi/3) cy = r \* np.sqrt(3) \* np.sin(np.pi/2 + i \* np.pi/3)

ax.plot(cx + r \* np.cos(angles), cy + r \* np.sin(angles)) ax.plot(-9 + cx + r \* np.cos(angles), cy - 1.8 + r \* np.sin(angles)) ax.plot(9 + cx + r \* np.cos(angles), cy + 1.8 + r \* np.sin(angles)) ax.plot(6 - cx + r \* np.cos(angles), cy - 6.9 + r \* np.sin(angles))

# Label hexagons

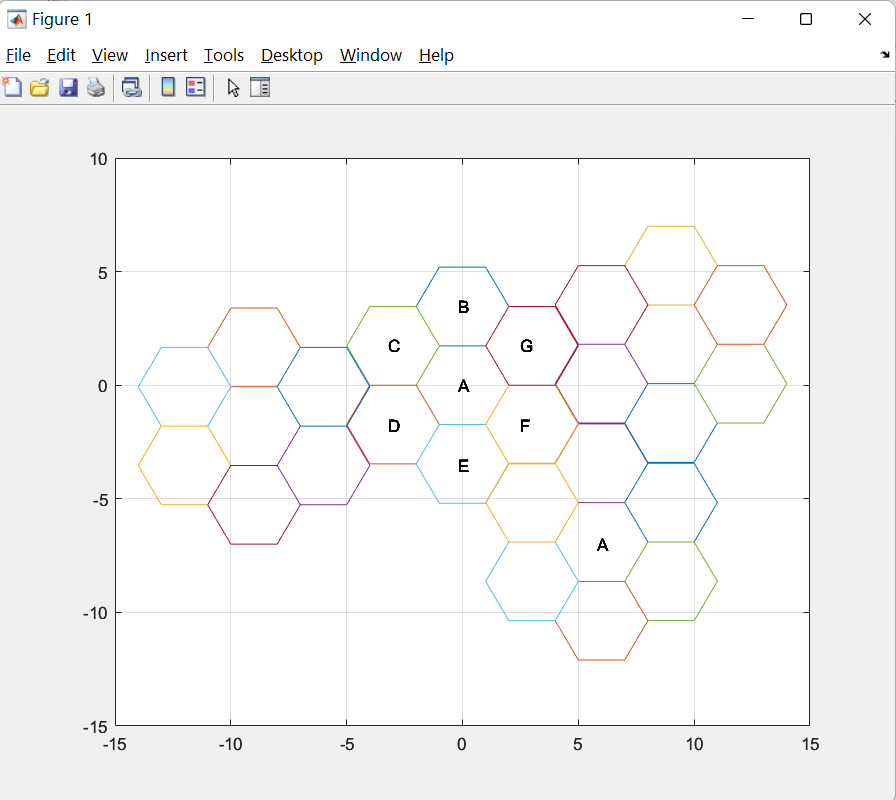
ax.text(cx - 0.2, cy, labels[i])

# Add the label for the last hexagon ax.text(5.8, -3.4, 'A')

# Set axis limits and display the grid ax.set\_xlim(-15, 15) ax.set\_ylim(-15, 10) ax.grid(True)

# Show the plot

plt.show()

**OUTPUT:-**

**Conclusion:**

Using the method of complex numbers, we can specify six points of hexagon by inter-relating them with trigonometry and, hence, plotting them together on the graph and joining them will create a hexagon. Hence, using Python, we can define several such equations which can help us to create hexagonal clusters. Similarly, we can follow this method for creating pentagonal, heptagonal, and quadrilateral clusters.