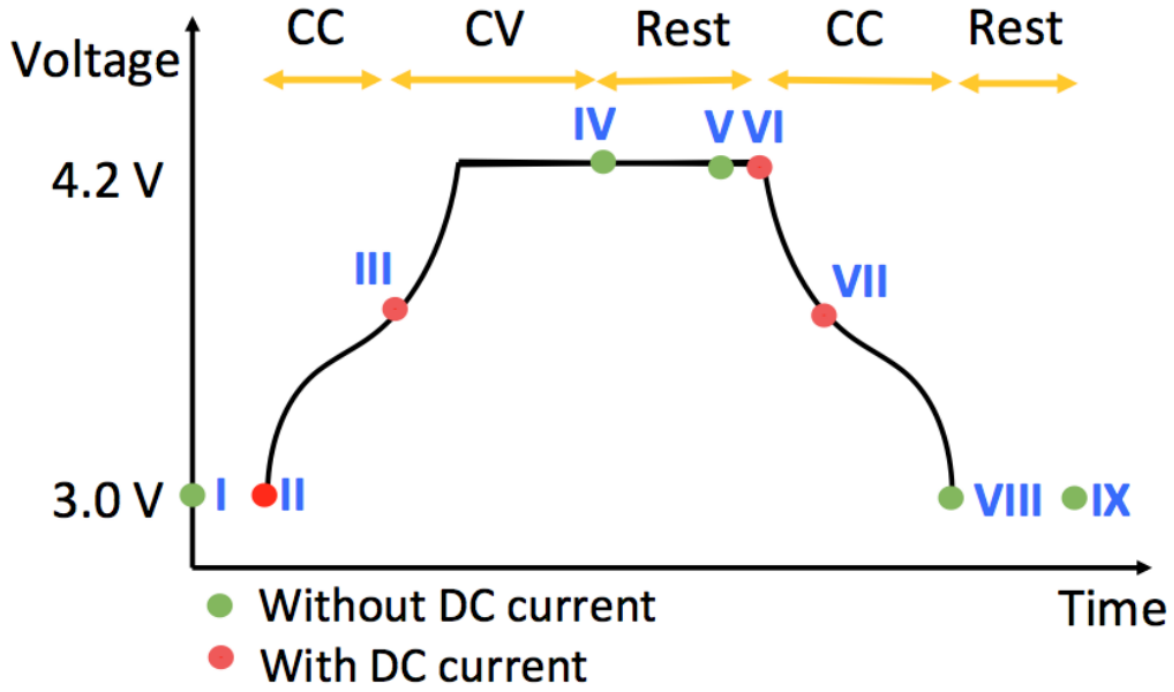
EIS Data Analysis

1. **Number of Cells Used**: The experiment was carried out on 12 commercially available cells.
2. **Type of Chemistry**: The cell chemistry is `LiCoO2/graphite`.
3. **Testing Conditions**:
   - The cells were cycled in three climate chambers set to temperatures of 25°C, 35°C, and 45°C.
   - Each cycle consisted of a 1C-rate (45mA) CCCV (constant current-constant voltage) charge up to 4.2 V and a 2C-rate (90 mA) CC (constant current) discharge down to 3 V.
   - Electrochemical Impedance Spectroscopy (EIS) was measured at nine different stages of charging/discharging during every even-numbered cycle in the frequency range.
   - All cells underwent 30 cycles at room temperature of 25°C before different temperatures were set.
   - The battery was cycled until its end of life (EoL), which is defined as when capacity drops below 80% of its initial value after undergoing these 30 cycles.
4. **Capacity of the Cells**: The cells were `45 mAh` Eunicell LR2032 Li-ion coin cells.

This information provides a detailed overview of the cells used in the study and the conditions under which they were tested.

- **EIS measurement** of cells are taken at differrent temperature and SOC state. Information about the state measurement is given in following figure.
-



Note: Dataset is available at link (https://zenodo.org/records/3633835)

In [1]:
```
# loading the data
import os
import pandas as pd

# Set the path to the folder containing the data files
data_folder = 'EIS Data/EIS data'
data_path = os.path.join(os.getcwd(), data_folder)

# Get the file paths for all .txt files in the folder
file_paths = [os.path.join(data_path, f) for f in os.listdir(data_path) if f.endswith('.txt')]

# Read the data from the files and store in a dictionary with the key as the name of the file
data_dict = {}
for file_path in file_paths:
    file_name = os.path.basename(file_path)
    df = pd.read_csv(file_path, sep='\t')
    data_dict[file_name] = df

```

In [2]:
```
# checking weather data is fully loaded or not
print(data_dict.keys())
```

```
dict_keys(['EIS_state_III_25C01.txt', 'EIS_state_III_25C02.txt', 'EIS_state_III_25C03.txt', 'EIS_state_III_25C04.txt', 'EIS_state_III_25C05.txt', 'EIS_state_III_25C06.txt', 'EIS_state_III_25C07.txt', 'EIS_state_III_25C08.txt', 'EIS_state_III_35C01.txt', 'EIS_state_III_35C02.txt', 'EIS_state_III_45C01.txt', 'EIS_state_III_45C02.txt', 'EIS_state_II_25C01.txt', 'EIS_state_II_25C02.txt', 'EIS_state_II_25C03.txt', 'EIS_state_II_25C04.txt', 'EIS_state_II_25C05.txt', 'EIS_state_II_25C06.txt', 'EIS_state_II_25C07.txt', 'EIS_state_II_25C08.txt', 'EIS_state_II_35C01.txt', 'EIS_state_II_35C02.txt', 'EIS_state_II_45C01.txt', 'EIS_state_II_45C02.txt', 'EIS_state_IV_25C01.txt', 'EIS_state_IV_25C02.txt', 'EIS_state_IV_25C03.txt', 'EIS_state_IV_25C04.txt', 'EIS_state_IV_25C05.txt', 'EIS_state_IV_25C06.txt', 'EIS_state_IV_25C07.txt', 'EIS_state_IV_25C08.txt', 'EIS_state_IV_35C01.txt', 'EIS_state_IV_35C02.txt', 'EIS_state_IV_45C01.txt', 'EIS_state_IV_45C02.txt', 'EIS_state_IX_25C01.txt', 'EIS_state_IX_25C02.txt', 'EIS_state_IX_25C03.txt', 'EIS_state_IX_25C04.txt', 'EIS_state_IX_25C05.txt', 'EIS_state_IX_25C06.txt', 'EIS_state_IX_25C07.txt', 'EIS_state_IX_25C08.txt', 'EIS_state_IX_35C01.txt', 'EIS_state_IX_35C02.txt', 'EIS_state_IX_45C01.txt', 'EIS_state_IX_45C02.txt', 'EIS_state_I_25C01.txt', 'EIS_state_I_25C02.txt', 'EIS_state_I_25C03.txt', 'EIS_state_I_25C04.txt', 'EIS_state_I_25C05.txt', 'EIS_state_I_25C06.txt', 'EIS_state_I_25C07.txt', 'EIS_state_I_25C08.txt', 'EIS_state_I_35C01.txt', 'EIS_state_I_35C02.txt', 'EIS_state_I_45C01.txt', 'EIS_state_I_45C02.txt', 'EIS_state_VIII_25C04.txt', 'EIS_state_VIII_25C06.txt', 'EIS_state_VIII_25C07.txt', 'EIS_state_VIII_25C08.txt', 'EIS_state_VII_25C04.txt', 'EIS_state_VII_25C06.txt', 'EIS_state_VII_25C07.txt', 'EIS_state_VII_25C08.txt', 'EIS_state_VI_25C01.txt', 'EIS_state_VI_25C02.txt', 'EIS_state_VI_25C03.txt', 'EIS_state_VI_25C05.txt', 'EIS_state_VI_25C06.txt', 'EIS_state_VI_25C07.txt', 'EIS_state_VI_25C08.txt', 'EIS_state_VI_25C42.txt', 'EIS_state_V_25C01.txt', 'EIS_state_V_25C02.txt', 'EIS_state_V_25C03.txt', 'EIS_state_V_25C04.txt', 'EIS_state_V_25C05.txt', 'EIS_state_V_25C06.txt', 'EIS_state_V_25C07.txt', 'EIS_state_V_25C08.txt', 'EIS_state_V_35C01.txt', 'EIS_state_V_35C02.txt', 'EIS_state_V_45C01.txt', 'EIS_state_V_45C02.txt'])
```

In [3]:
```
EIS_state_III_25C01 = data_dict['EIS_state_III_25C01.txt']
# converting the data into dataframe
EIS_state_III_25C01 = pd.DataFrame(EIS_state_III_25C01)
# shape of the data
print(EIS_state_III_25C01.shape)
# Print the first 5 rows of the data
EIS_state_III_25C01.head()
```

```
(15660, 7)
```

Out[3]:

	time/s	cycle number	freq/Hz	Re(Z)/Ohm	-Im(Z)/Ohm	|Z|/Ohm	Phase(Z)/deg
0	9651.00252	1.0	20004.4530	0.39182	-0.03055	0.39301	4.45806
1	9651.14552	1.0	15829.1260	0.39693	-0.01230	0.39712	1.77458
2	9651.28752	1.0	12516.7030	0.40190	0.00234	0.40191	-0.33358
3	9651.42952	1.0	9909.4424	0.40953	0.01629	0.40986	-2.27833
4	9651.59352	1.0	7835.4800	0.41789	0.02743	0.41879	-3.75550

In [4]:
```
# check the column names
EIS_state_III_25C01.columns
# Remove the spaces in the column names before and after the column names
EIS_state_III_25C01.columns = EIS_state_III_25C01.columns.str.strip()
# check the column names
EIS_state_III_25C01.columns
```

Out[4]:
```
Index(['time/s', 'cycle number', 'freq/Hz', 'Re(Z)/Ohm', '-Im(Z)/Ohm',
       '|Z|/Ohm', 'Phase(Z)/deg'],
      dtype='object')
```

Analysis

- From the graph it can be seen that the as the cell ages how resistance changes from cycle 1 to cycle 261.

EIS State_I_25C

In [5]:
```
EIS_state_I_25C01 = data_dict['EIS_state_I_25C01.txt']
# converting the data into dataframe
EIS_state_I_25C01 = pd.DataFrame(EIS_state_I_25C01)
# shape of the data
print(EIS_state_I_25C01.shape)
# Print the first 5 rows of the data
EIS_state_I_25C01.head()
```

```
(21000, 7)
```

Out[5]:

	time/s	cycle number	freq/Hz	Re(Z)/Ohm	-Im(Z)/Ohm	|Z|/Ohm	Phase(Z)/deg
0	7520.78391	1.0	20004.4530	0.40128	-0.02956	0.40237	4.21264
1	7520.92691	1.0	15829.1260	0.40688	-0.01046	0.40701	1.47198
2	7521.06891	1.0	12516.7030	0.41389	0.00415	0.41391	-0.57420
3	7521.21091	1.0	9909.4424	0.42152	0.01708	0.42187	-2.31990
4	7521.37491	1.0	7835.4800	0.42905	0.02692	0.42990	-3.59049

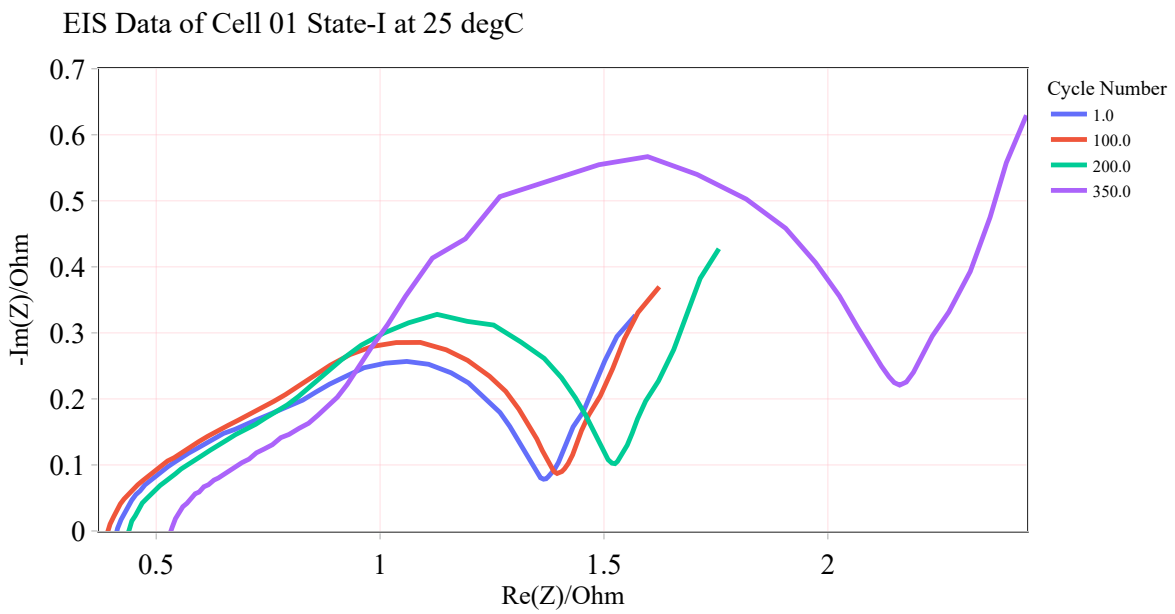
In [6]:
```
# check the column names
EIS_state_I_25C01.columns
# Remove the spaces in the column names before and after the column names
EIS_state_I_25C01.columns = EIS_state_I_25C01.columns.str.strip()
# check the column names
EIS_state_I_25C01.columns
```

Out[6]:
```
Index(['time/s', 'cycle number', 'freq/Hz', 'Re(Z)/Ohm', '-Im(Z)/Ohm',
       '|Z|/Ohm', 'Phase(Z)/deg'],
      dtype='object')
```

In [ ]:
```
%pip install plotly
```

In [9]:

```python
import plotly.express as px

# Select only the desired cycle numbers
cycle_numbers = [1, 100, 200, EIS_state_I_25C01['cycle number'].max()]

# Filter the data to include only the selected cycle numbers
EIS_filtered = EIS_state_I_25C01[EIS_state_I_25C01['cycle number'].isin(cycle_numbers)]

fig = px.line(EIS_filtered, x="Re(Z)/Ohm", y="-Im(Z)/Ohm", color='cycle number')

# Update the layout of the figure
fig.update_layout(
    title="EIS Data of Cell 01 State-I at 25 degC",
    xaxis_title="Re(Z)/Ohm",
    yaxis_title="-Im(Z)/Ohm",
    font=dict(
        family="Times New Roman",
        size=18,
        color="black"
    ),
    legend=dict(
        title="Cycle Number",
        font=dict(
            family="Times New Roman",
            size=14,
            color="black"
        )
    )
)

# Update the style of the lines
fig.update_traces(
    line=dict(
        width=4,
        dash="solid"
    )
)

# add axis line and ticks
fig.update_xaxes(showline=True, linewidth=2, linecolor='black', mirror=True, ticks="outside")
fig.update_yaxes(showline=True, linewidth=2, linecolor='black', mirror=True, ticks="outside")
# make background white
fig.update_layout(plot_bgcolor='white')
# Move x-label and y-label to the center
fig.update_layout(xaxis=dict(title=dict(standoff=5)), yaxis=dict(title=dict(standoff=5)))
# Icncrease x-axis and y-axis tick font size and font family
fig.update_xaxes(tickfont=dict(size=24, family='Times New Roman', color='black'))
fig.update_yaxes(tickfont=dict(size=24, family='Times New Roman', color='black'))
# Make y-axis to start it from 0
fig.update_yaxes(range=[0, 0.7])
# add grid
fig.update_layout(xaxis=dict(showgrid=True, gridwidth=0.5, gridcolor='LightPink'),
                  yaxis=dict(showgrid=True, gridwidth=1, gridcolor='LightPink'))

fig.show()

```

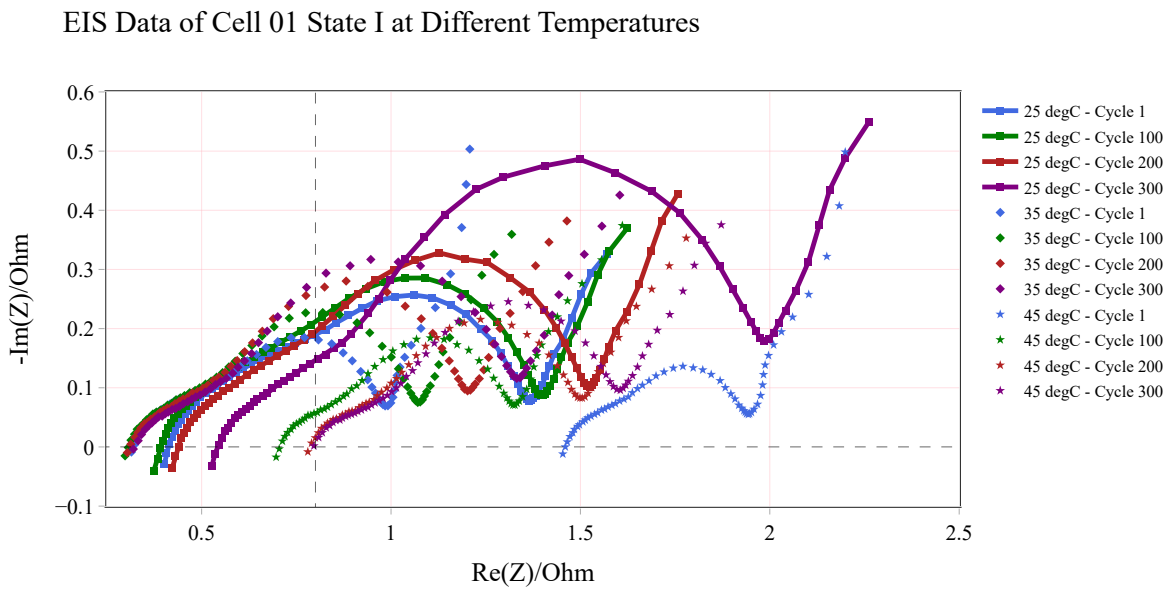


Analysis

- From the graph it can be seen that the as the cell ages how resistance changes from cycle 1 to cycle 350.
- The point where each curve intersect with the x-axis at far left side indicate the `ohmic resistance` . As you can see that the overall the resistance is increasing as the battery cell ages.
- Diameter of semi-circle represents the `charge transfer resistance` . As the cycle number increases, it appears that the charge transfer resistance increases, indicating a degradation in cell performance.
- The straight line in the low frequency region represents the warburg impedance associated with the `diffusion process` .

In [12]:

```
import plotly.graph_objects as go

# Function to add traces based on filtered data
def add_traces_to_fig(fig, data, temperature, mode='lines', dash='solid', marker_symbol=None):
    colors = ['royalblue', 'green', 'firebrick', 'purple']  # For different cycle numbers
    for idx, cycle_num in enumerate(cycle_numbers):
        filtered_data = data[data['cycle number'] == cycle_num]
        fig.add_trace(go.Scatter(x=filtered_data['Re(Z)/Ohm'], y=filtered_data['-Im(Z)/Ohm'],
                                 mode=mode,
                                 name=f'{temperature} - Cycle {cycle_num}',
                                 line=dict(
                                     color=colors[idx],
                                     width=4,
                                     dash=dash),
                                 marker=dict(
                                     symbol=marker_symbol
                                 )))

EIS_state_I_35C01 = data_dict['EIS_state_I_35C01.txt']
# converting the data into dataframe
EIS_state_I_35C01 = pd.DataFrame(EIS_state_I_35C01)
EIS_state_I_35C01.columns = ['time/s', 'cycle number', 'freq/Hz', 'Re(Z)/Ohm',
                             '-Im(Z)/Ohm', '|Z|/Ohm', 'Phase(Z)/deg']
EIS_state_I_45C01 = data_dict['EIS_state_I_45C01.txt']
# converting the data into dataframe
EIS_state_I_45C01 = pd.DataFrame(EIS_state_I_45C01)
EIS_state_I_45C01.columns = ['time/s', 'cycle number', 'freq/Hz', 'Re(Z)/Ohm',
                             '-Im(Z)/Ohm', '|Z|/Ohm', 'Phase(Z)/deg']

# Select and filter data for 25C
cycle_numbers = [1, 100, 200, 300]
EIS_filtered25 = EIS_state_I_25C01[EIS_state_I_25C01['cycle number'].isin(cycle_numbers)]

# Select and filter data for 35C
EIS_filtered35 = EIS_state_I_35C01[EIS_state_I_35C01['cycle number'].isin(cycle_numbers)]

# Select and filter data for 45C
EIS_filtered45 = EIS_state_I_45C01[EIS_state_I_45C01['cycle number'].isin(cycle_numbers)]

# Create a figure
fig = go.Figure()

# Add traces for each temperature
add_traces_to_fig(fig, EIS_filtered25, "25 degC", mode='lines+markers', marker_symbol='square')
add_traces_to_fig(fig, EIS_filtered35, "35 degC", mode='markers', marker_symbol='diamond')
add_traces_to_fig(fig, EIS_filtered45, "45 degC", mode='markers', marker_symbol='star')

# Rest of your figure updates
# add axis line and ticks
fig.update_xaxes(showline=True, linewidth=2, linecolor='black', mirror=True, ticks="outside") # , range=[0, 0.7]
fig.update_yaxes(showline=True, linewidth=2, linecolor='black', mirror=True, ticks="outside")
# make background white
fig.update_layout(plot_bgcolor='white')
# Move x-label and y-label to the center
fig.update_layout(xaxis=dict(title=dict(standoff=10)), yaxis=dict(title=dict(standoff=10)))
# Icncrease x-axis and y-axis tick font size and font family
fig.update_xaxes(tickfont=dict(size=18, family='Times New Roman', color='black'))
fig.update_yaxes(tickfont=dict(size=18, family='Times New Roman', color='black'))
# add grid
fig.update_layout(xaxis=dict(showgrid=True, gridwidth=1, gridcolor='LightPink'),
                  yaxis=dict(showgrid=True, gridwidth=1, gridcolor='LightPink'))
# fig size
#fig.update_layout(height=800, width=1000)
# make y-axis range to start from 0
fig.update_yaxes(range=[-.1, 0.6])
fig.update_xaxes(range=[0.25, 2.5])
# add a grid line at y=0
fig.add_shape(type="line", x0=0, y0=0, x1=2.5, y1=0, line=dict(color="black", width=0.5, dash="dash"))
fig.add_shape(type="line", x0=0.8, y0=-0.15, x1=0.8, y1=0.6, line=dict(color="black", width=0.5, dash="dash"))
#Making legend font size bigger and bold and Times New Roman font
fig.update_layout(legend=dict(font=dict(family='Times New Roman', size=14, color='black')))
# Update the layout of the figure
fig.update_layout(
    title="EIS Data of Cell 01 State I at Different Temperatures",
    xaxis_title="Re(Z)/Ohm",
    yaxis_title="-Im(Z)/Ohm",
    font=dict(
        family="Times New Roman",
        size=18,
        color="black"
    ))

fig.show()

```



Observations

- At 25°C, the impedance values appear to increase progressively with the cycle number, particularly evident in the high-frequency region. This increase suggests that the cell's internal resistance is growing with continued cycling, indicating degradation processes occurring within the cell.
- At 35°C, a similar pattern emerges. However, the semicircles appear slightly more pronounced and compact. The overall resistance seems to be slightly lower than that of 25°C initially (Cycle 1), but it does show increasing tendencies with cycling, suggesting that the temperature increment speeds up certain reactions and degradation.
- At 45°C, the impedance values seem to start off higher even at Cycle 1 when compared to the other two temperatures. The trend of increasing impedance with cycle number persists. This might imply that at this elevated temperature, the degradation processes are more pronounced right from the beginning for this type of perticular cell.

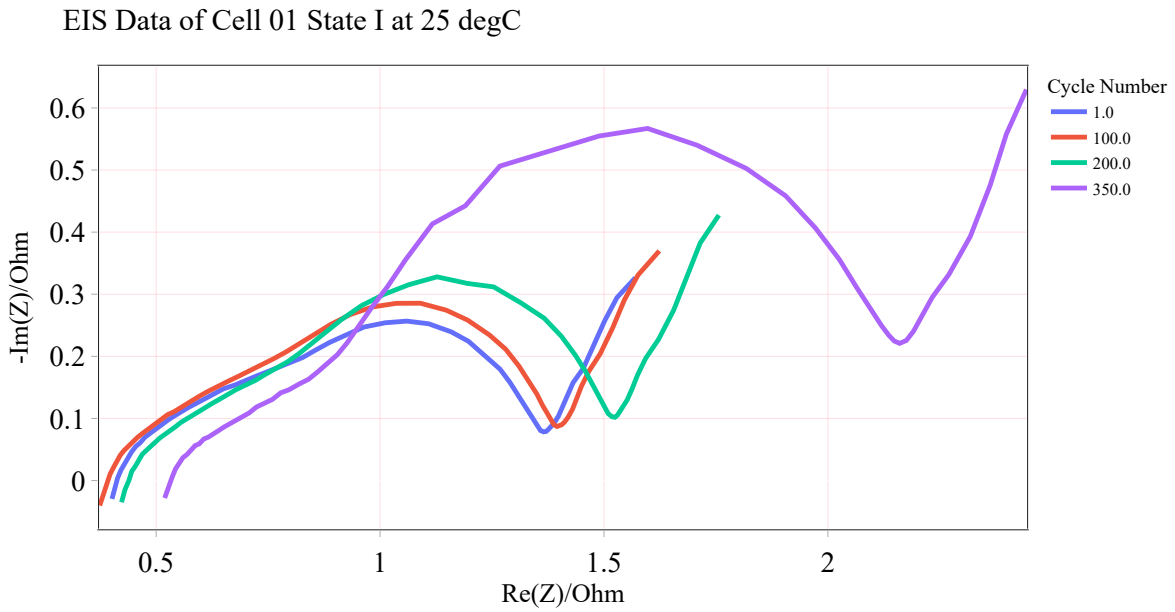
Detailed Data Visualization of cells at different temperatures and different stages

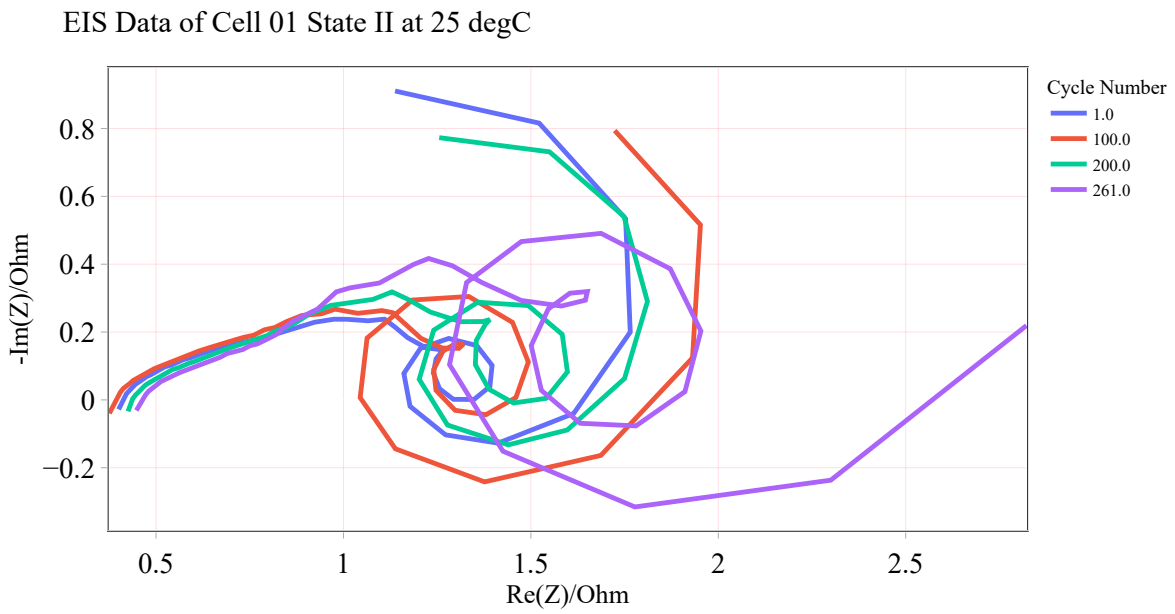
In [13]:

```python
import pandas as pd
import plotly.express as px

def plot_EIS_data(state, temp, cell_num, data_dict):
    # Construct the filename
    filename = f"EIS_state_{state}_{temp}C{cell_num:02d}.txt"

    # Fetch the data from the dictionary
    data = data_dict[filename]

    # Convert the data to a DataFrame
    df = pd.DataFrame(data)

    # Rename the columns
    df.columns = ['time/s', 'cycle number', 'freq/Hz', 'Re(Z)/Ohm', '-Im(Z)/Ohm', '|Z|/Ohm', 'Phase(Z)/deg']

    # Filter the data by cycle number
    cycle_numbers = [1, 100, 200, df['cycle number'].max()]
    df_filtered = df[df['cycle number'].isin(cycle_numbers)]

    # Plot
    fig = px.line(df_filtered, x="Re(Z)/Ohm", y="-Im(Z)/Ohm", color='cycle number')

    # Update layout and styling...
        # Update the layout of the figure
    fig.update_layout(
        title="EIS Data of Cell 01 State V at 45 degC",
        xaxis_title="Re(Z)/Ohm",
        yaxis_title="-Im(Z)/Ohm",
        font=dict(
            family="Times New Roman",
            size=18,
            color="black"
        ),
        legend=dict(
            title="Cycle Number",
            font=dict(
                family="Times New Roman",
                size=14,
                color="black"
            )
        )
    )

    # Update the style of the lines
    fig.update_traces(
        line=dict(
            width=4,
            dash="solid"
        )
    )

    # add axis line and ticks
    fig.update_xaxes(showline=True, linewidth=2, linecolor='black', mirror=True, ticks="outside")
    fig.update_yaxes(showline=True, linewidth=2, linecolor='black', mirror=True, ticks="outside")
    # make background white
    fig.update_layout(plot_bgcolor='white')
    # Move x-label and y-label to the center
    fig.update_layout(xaxis=dict(title=dict(standoff=5)), yaxis=dict(title=dict(standoff=5)))
    # Icncrease x-axis and y-axis tick font size and font family
    fig.update_xaxes(tickfont=dict(size=24, family='Times New Roman', color='black'))
    fig.update_yaxes(tickfont=dict(size=24, family='Times New Roman', color='black'))
    # Make y-axis to start it from 0
    #fig.update_yaxes(range=[0, 0.7])
    # add grid
    fig.update_layout(xaxis=dict(showgrid=True, gridwidth=0.5, gridcolor='LightPink'),
                      yaxis=dict(showgrid=True, gridwidth=1, gridcolor='LightPink'))


    # Update the title based on the parameters
    fig.update_layout(title=f"EIS Data of Cell {cell_num:02d} State {state} at {temp} degC")

    fig.show()

```

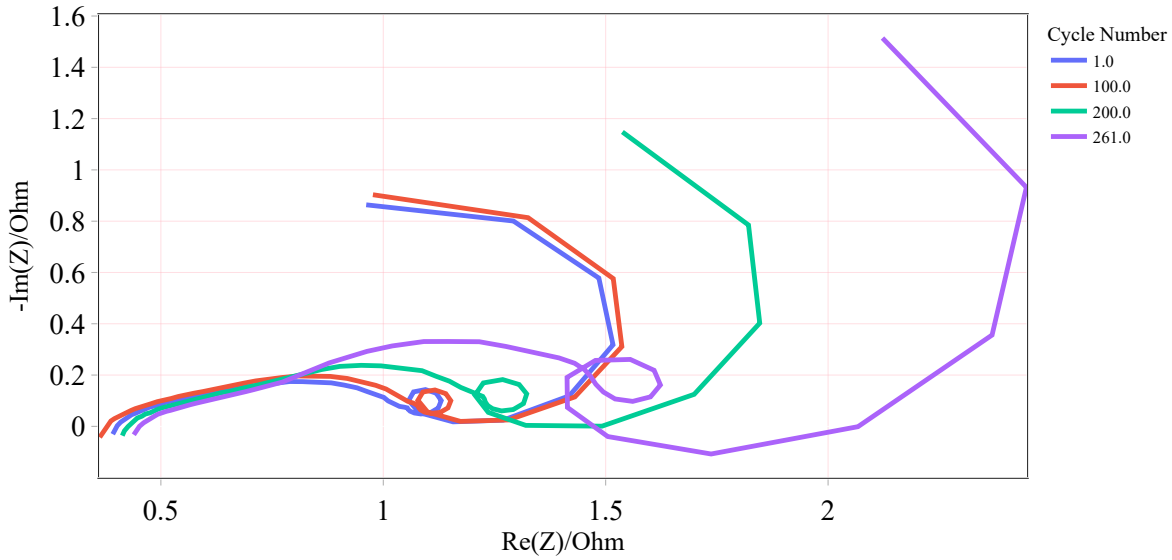
Plotting the Data of cell 01 at 25 degC

In [14]:
```
states = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX']

for state in states:
    filename = f"EIS_state_{state}_25C01.txt"
    if filename in data_dict:
        plot_EIS_data(state, 25, 1, data_dict)
    else:
        print(f"No data available for state {state}")


```

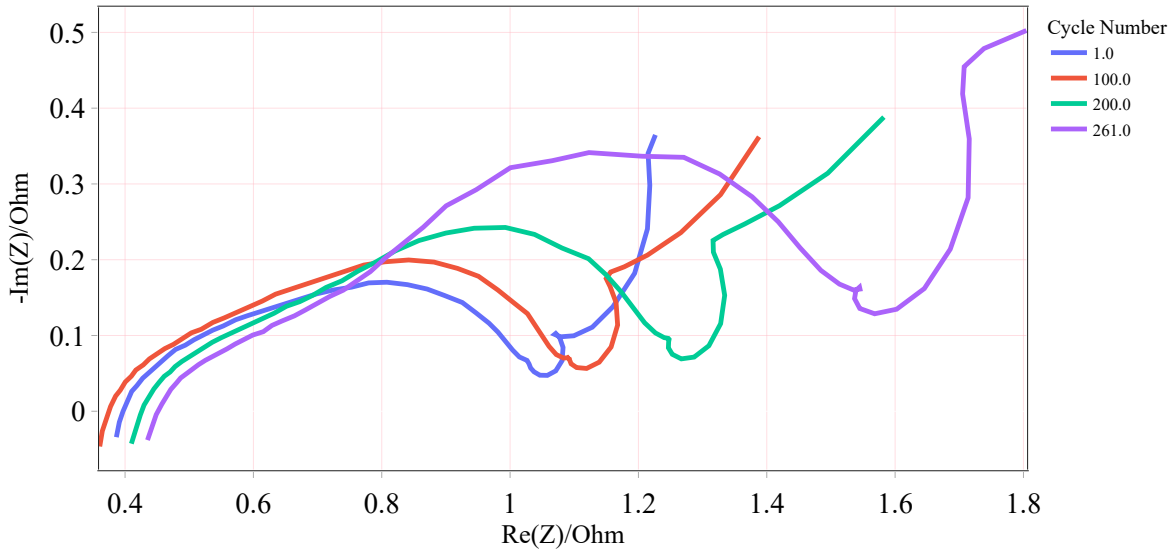




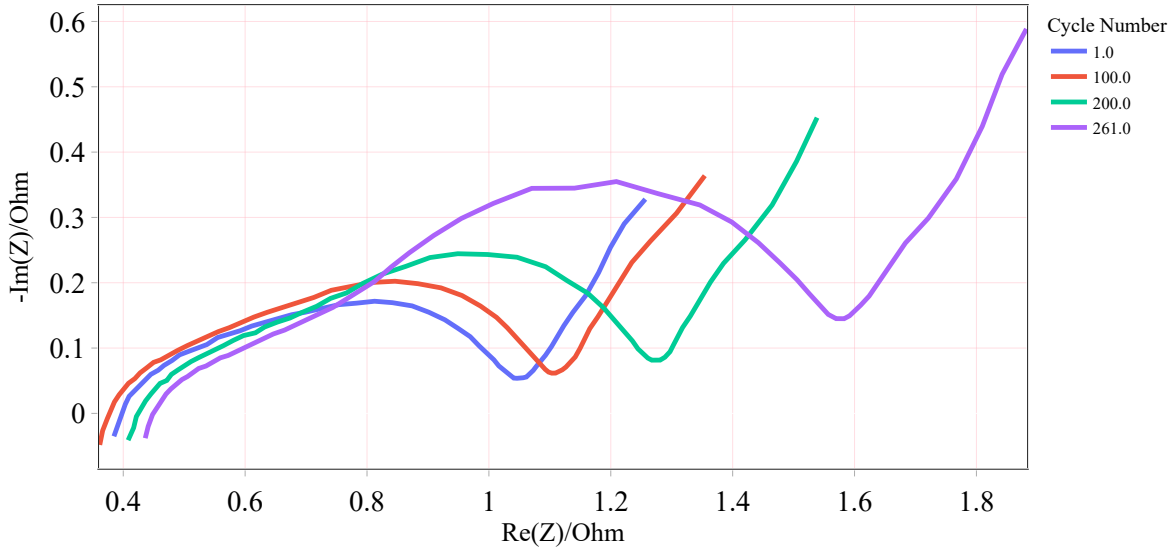
EIS Data of Cell 01 State III at 25 degC



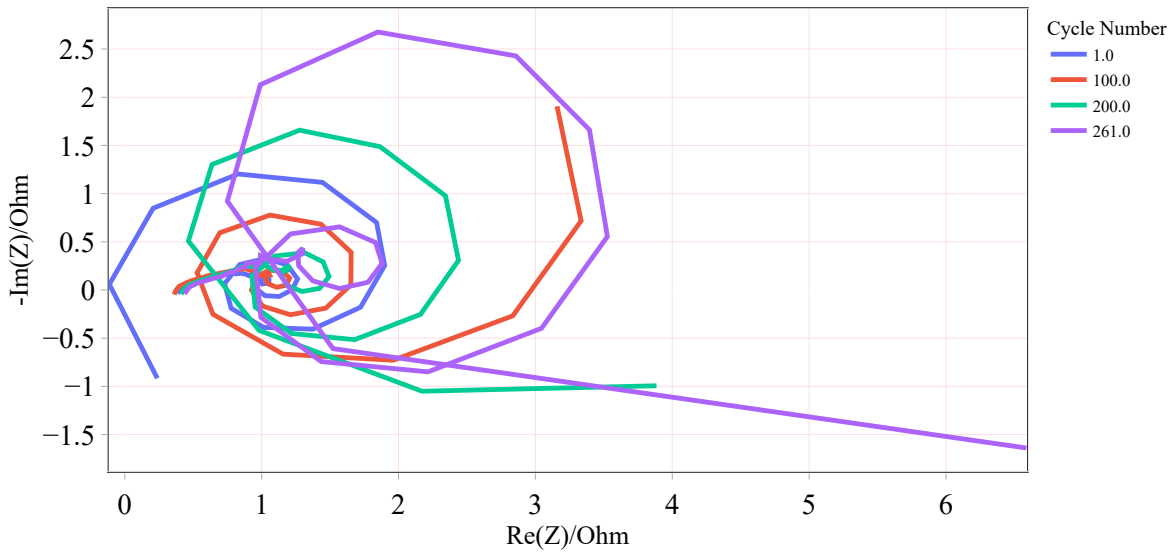
EIS Data of Cell 01 State IV at 25 degC



EIS Data of Cell 01 State V at 25 degC

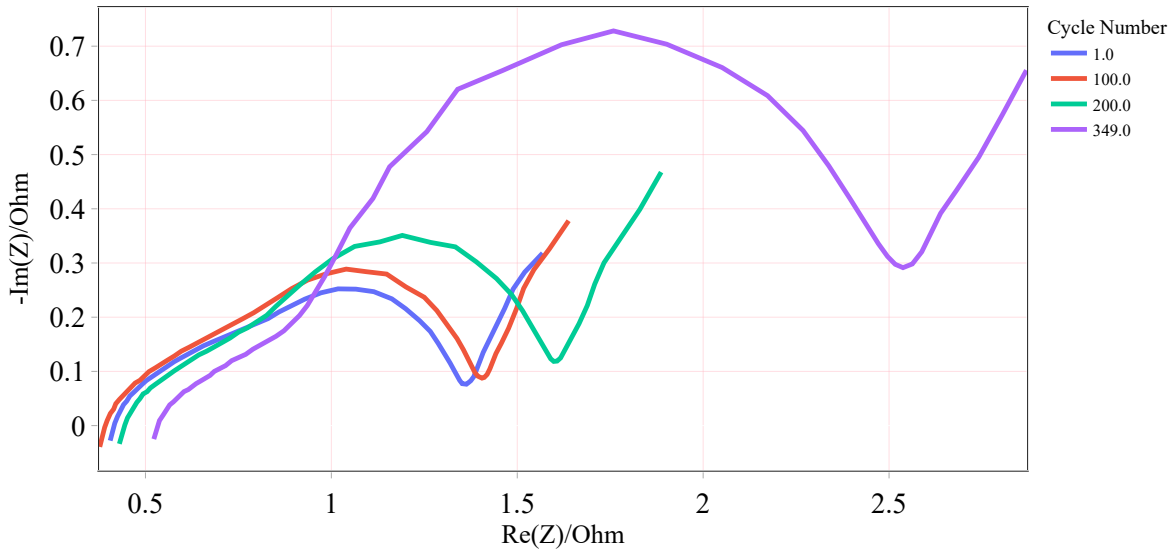


EIS Data of Cell 01 State VI at 25 degC



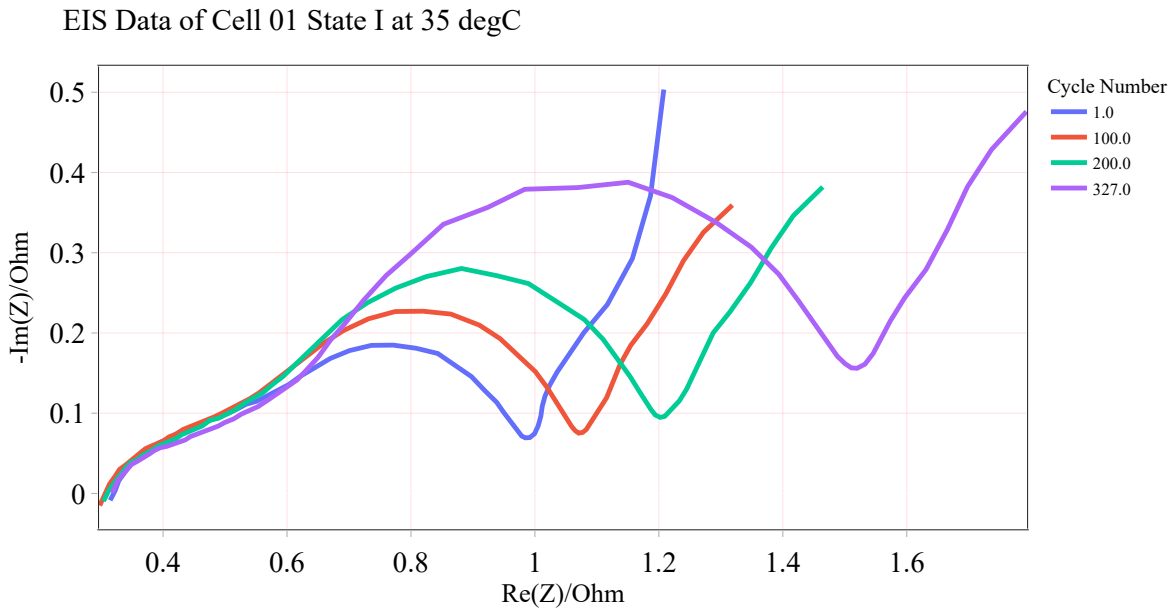
No data available for state VII
No data available for state VIII

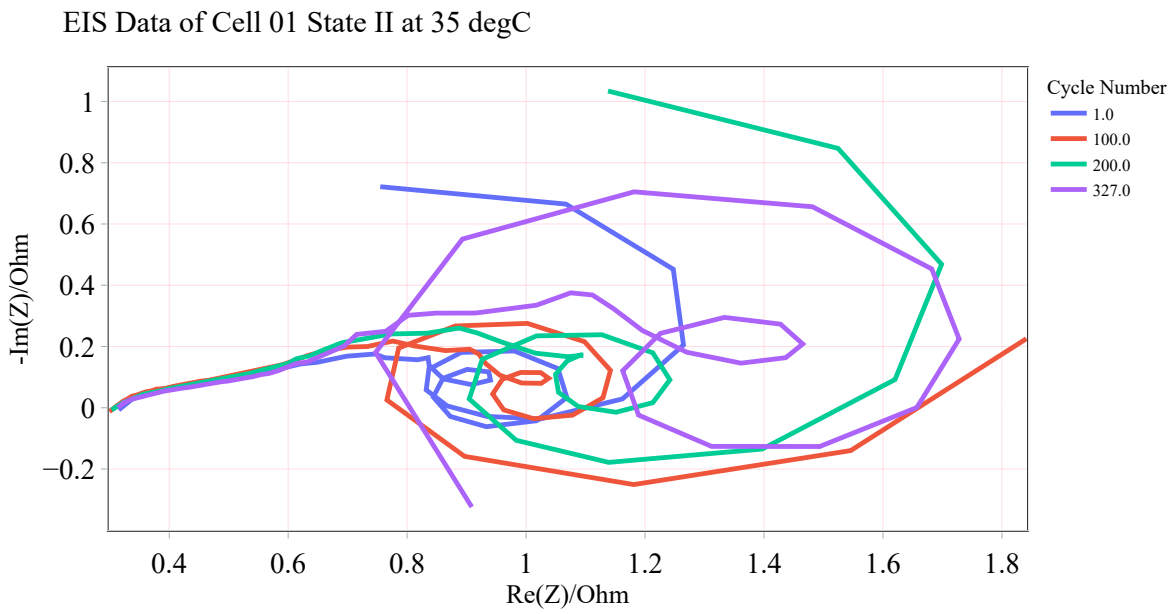
EIS Data of Cell 01 State IX at 25 degC



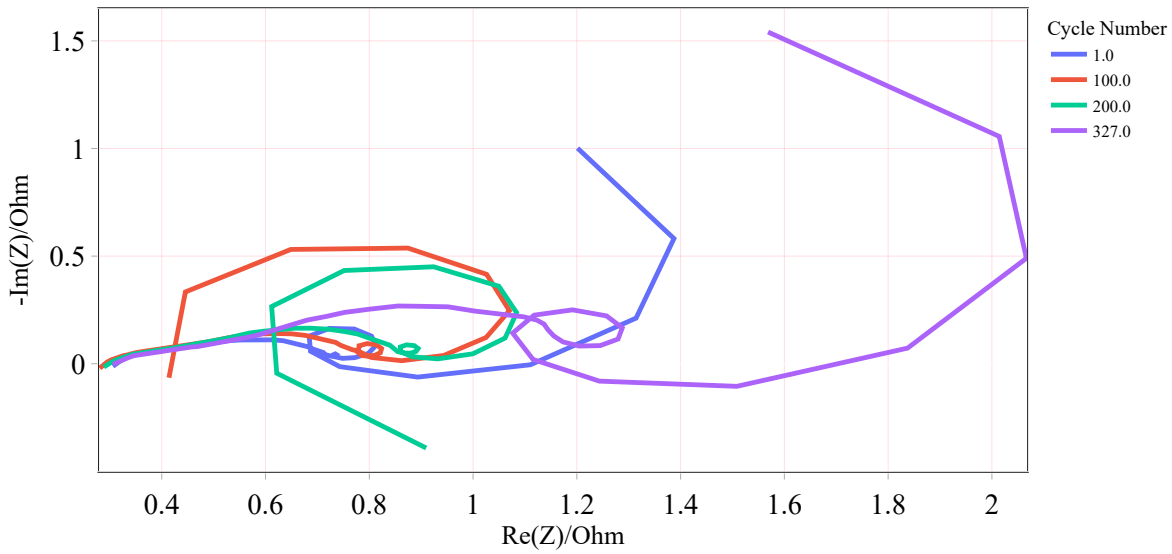
Plotting the Data of cell 01 at 35 degC

In [15]:
```
states = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX']

for state in states:
    filename = f"EIS_state_{state}_35C01.txt"
    if filename in data_dict:
        plot_EIS_data(state, 35, 1, data_dict)
    else:
        print(f"No data available for state {state}")


```

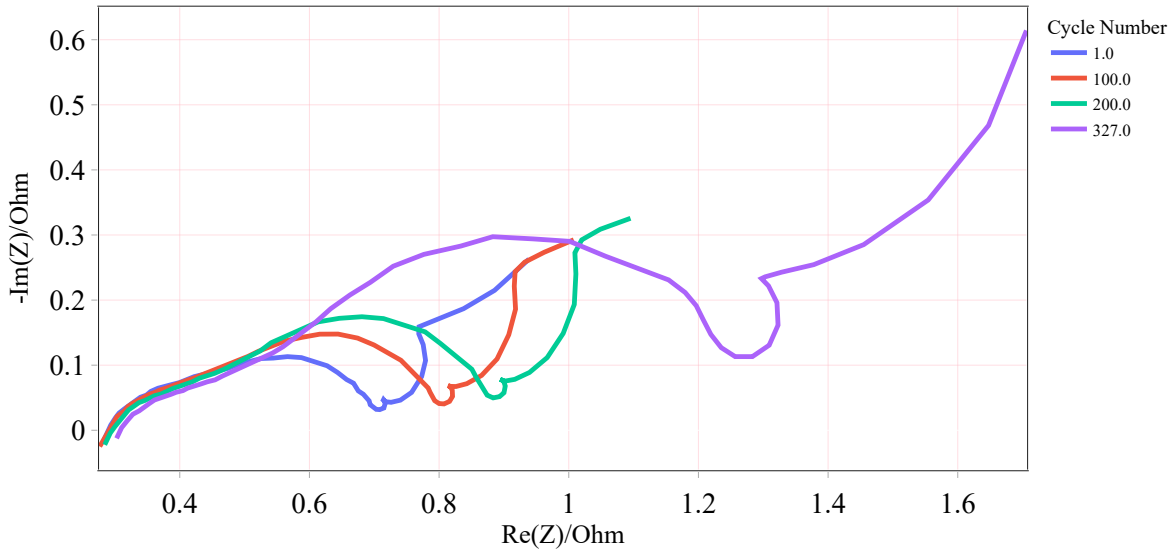




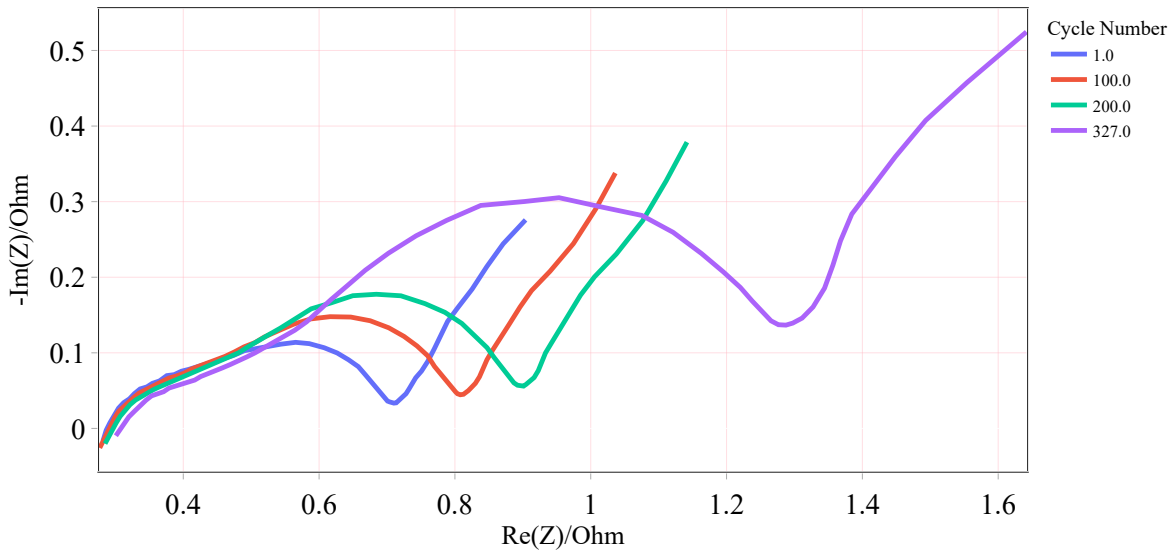
EIS Data of Cell 01 State III at 35 degC



EIS Data of Cell 01 State IV at 35 degC

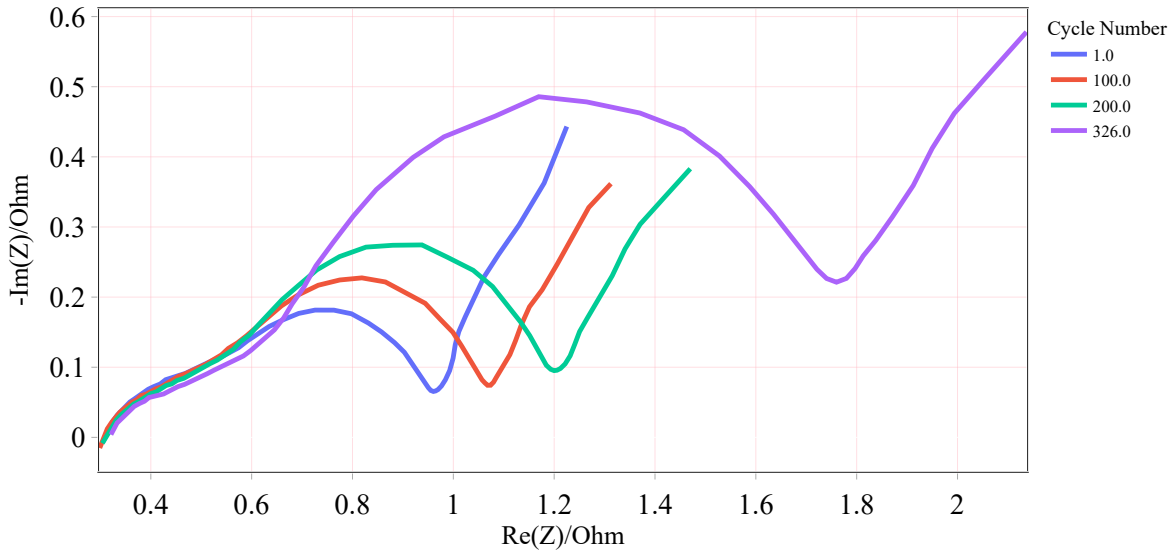


EIS Data of Cell 01 State V at 35 degC



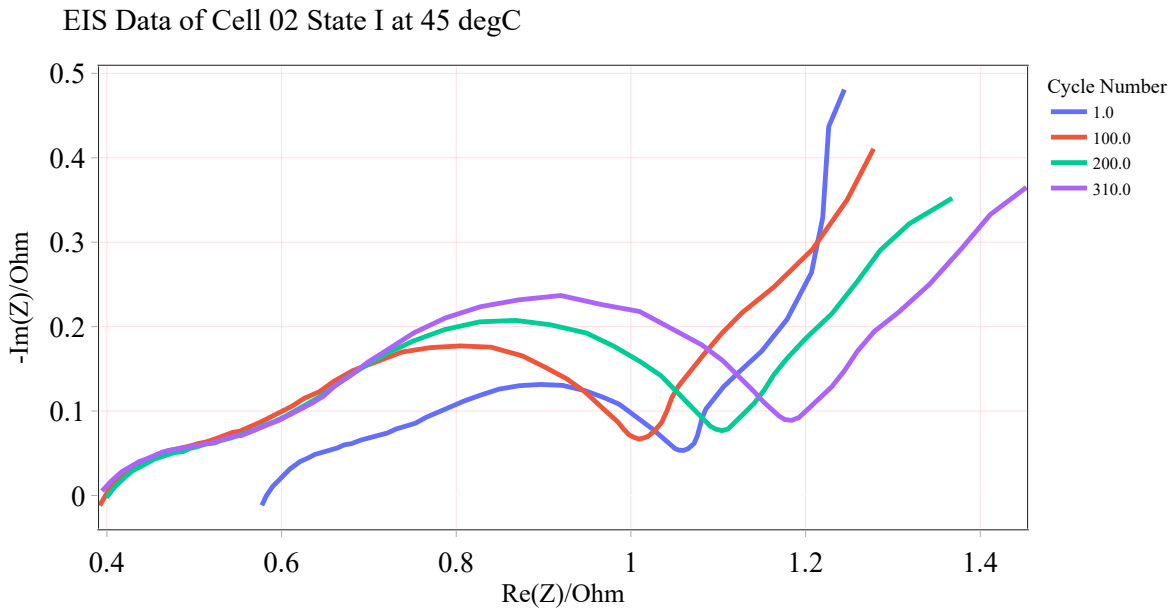
No data available for state VI
No data available for state VII
No data available for state VIII

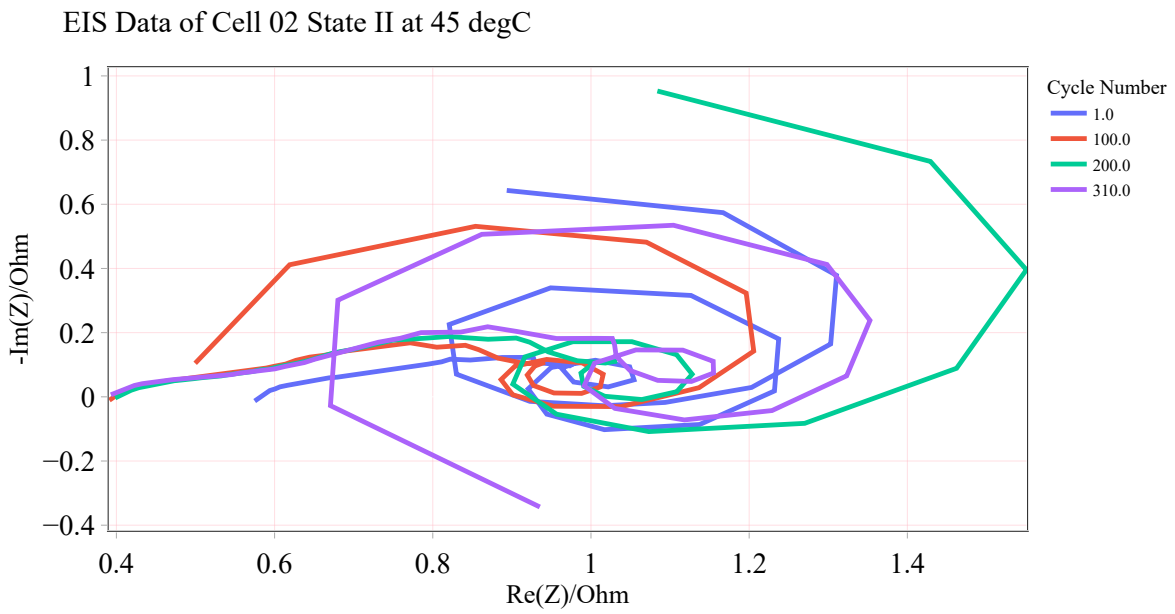
EIS Data of Cell 01 State IX at 35 degC



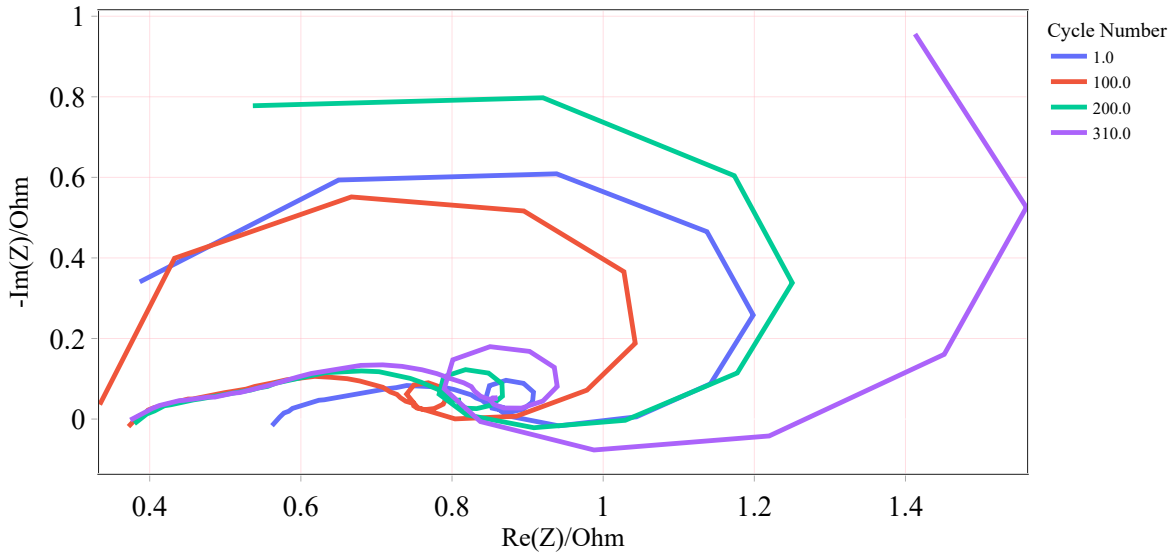
Plotting the Data of cell 01 at 45 degC

In [16]:
```
states = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX']

for state in states:
    filename = f"EIS_state_{state}_45C01.txt"
    if filename in data_dict:
        plot_EIS_data(state, 45, 2, data_dict)
    else:
        print(f"No data available for state {state}")


```

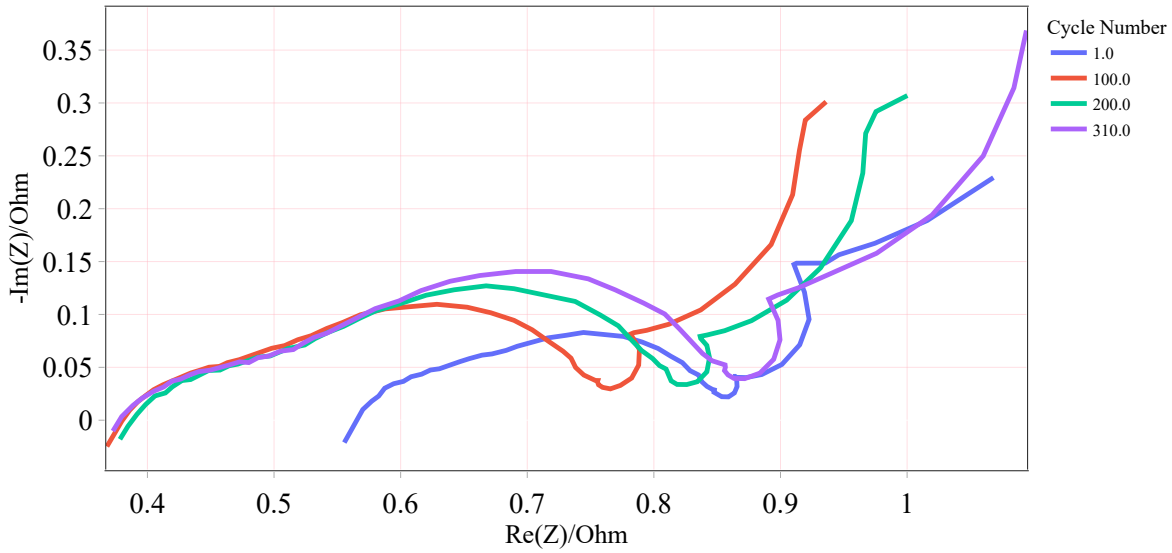




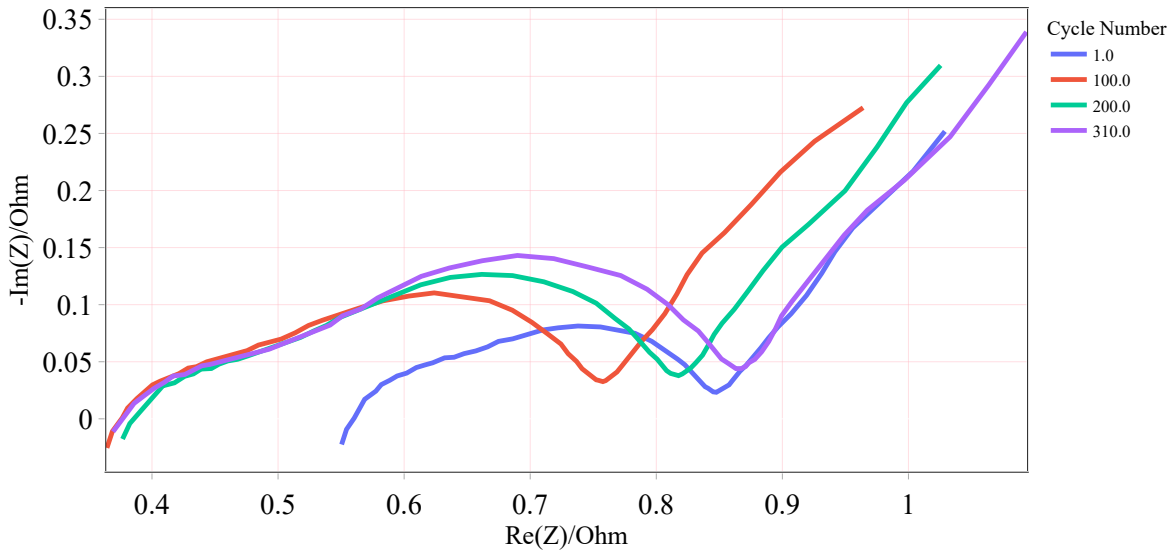
EIS Data of Cell 02 State III at 45 degC



EIS Data of Cell 02 State IV at 45 degC



EIS Data of Cell 02 State V at 45 degC



No data available for state VI
No data available for state VII
No data available for state VIII

EIS Data of Cell 02 State IX at 45 degC