# HITECH

RF & Microwave solutions

# SCRIPTING THE CST STUDIO SUITE USING PYTHON


- **Kamal Mustafa**
  - CST Studio Suite Instructor
  - SIMULIA CST Technical Support


# ABOUT HI-TECH

- 33 years representing RF & Microwave suppliers
- 20 years EDA Software
- Also: Components, T&M, Radar
- 18 years Electromagnetic Simulations
- Based in Zeist, Netherlands


# ABOUT THE SPEAKER

## Kamal Mustafa (Application Engineer)

- 2013-15 Master of Science @ Politecnico di Torino (Turin)
  - Telecommunications Engineering
- 2015-15 Research Assistant @ Chalmers University (Gothenburg)
- 2016-19 RF Engineer @ Rosenberger Asia Pacific (Shanghai)
- Joined HI-Tech in 2019 (Zeist)

# OUTLINE

- PYTHON Libraries Installation
- Open CST & Create A Project
- Parameters, Units, Frequency Setup
- Modeling and Simulation
- Other Possibilities
- Demo

# VALUE OF SCRIPTING

- Helps to perform repetitive or predictable tasks without direct human inputs.
- Provides a faster time to value, when business processes must change.
- Consistent, on-time output
- Fewer costly errors
- Growth & Scalability

# CST PYTHON LIBRARIES

- The standard CST installation package comes with Python 3.6 (64-bit).
- To work with CST Python Libraries with an external python distribution, include the directory in Python's system path.
- This workflow uses Python distribution **Anaconda** with the provided Python Libraries.

**Supported Python versions:**

- Python 3.9.x (64-bit)
- Python 3.8.x (64-bit)
- Python 3.7.x (64-bit)
- Python 3.6.x (64-bit)

32-bit versions of Python are not supported.

# CST PYTHON LIBRARIES

- The workflow is written in the **Jupyter® Notebook**.
- After a Python® environment is properly setup in the Jupyter® Notebook, CST Python Libraries can be imported with the following commands:

```
import sys
sys.path.append("C:\Program Files (x86)\CST Studio Suite2022\AMD64\python_cst_libraries")
import cst
import cst.results
import cst.interface
print(cst.__file__) # should print ' C:\Program Files (x86)\CST Studio Suite 2022\AMD64\python_cst_libraries\cst\__init__.py'
```

```
C:\Program Files (x86)\CST Studio Suite 2022\AMD64\python_cst_libraries\cst\__init__.py
```

# OPEN CST Studio Suite® & CREATE MICROWAVE STUDIO PROJECT

- Use the following command to open CST Studio Suite environment.

```
project = cst.interface.DesignEnvironment() #Open CST Design Environment
```

- Create a new Microwave Studio Project and save it.


```
my_CST = project.new_mws() #Create a new Microwave Studio project
```

```
folder_path = "C:\\Users\\Kamal\\jupyter-notebooks\\"my_CST.save(folder_path + "My_dipole.cst") #Save the CST project
```

# PARAMETERS HANDLING

- The code quoted as a string is a VBA code to store the parameter in Parameter List.
- The VBA code is assigned to the variable "`ParameterDefineString`" as a multiline string.

```
# 3. Add a parameter, define units, and define frequency range
# 3.1 Parameter handling
ParameterDefineString = """
Sub Main
StoreParameter("D","0.05")
StoreParameter("gap","L/200")
StoreParameter("L","150")
End Sub"""
my_CST.schematic.execute_vba_code(ParameterDefineString)
```

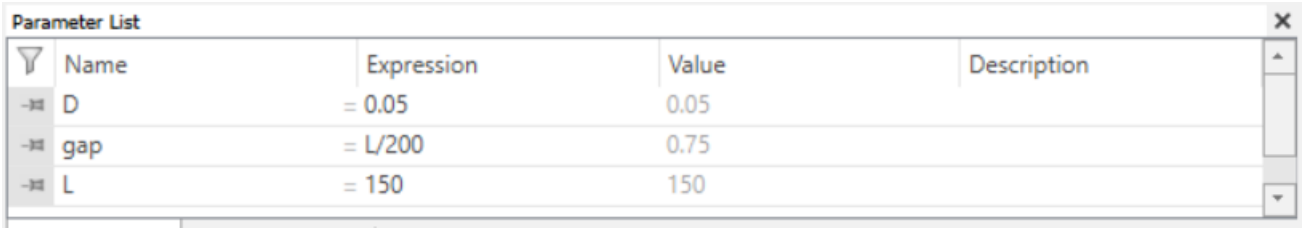
# PARAMETERS HANDLING

- The VBA code is executed by using a command "`execute_vba_code()`" defined in CST Python Libraries.

```
# 3. Add a parameter, define units, and define frequency range
# 3.1 Parameter handling
ParameterDefineString = """
Sub Main
StoreParameter("D","0.05")
StoreParameter("gap","L/200")
StoreParameter("L","150")
End Sub"""
my_CST.schematic.execute_vba_code(ParameterDefineString)
```

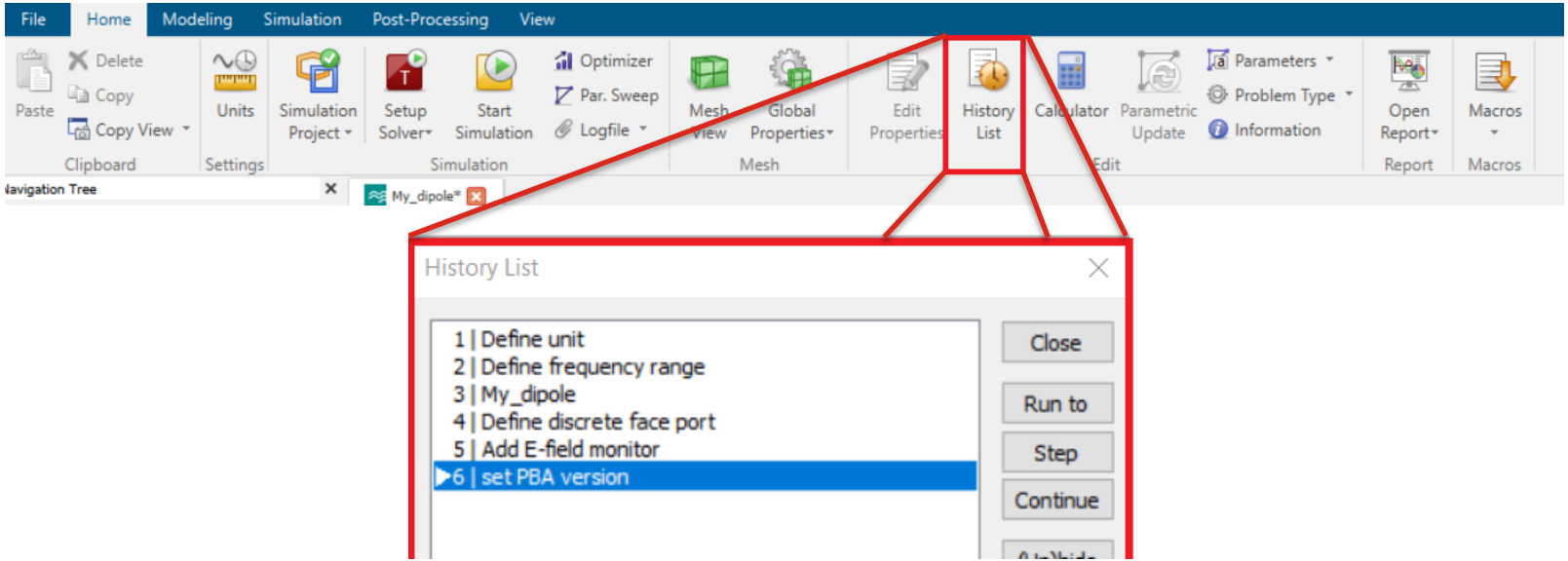
# PARAMETERS HANDLING

- After "`my_CST.schematic.execute_vba_code(ParameterDefineString)`" is executed, 3 new parameters with predefined values can be found in Parameter List in the CST Studio Suite®



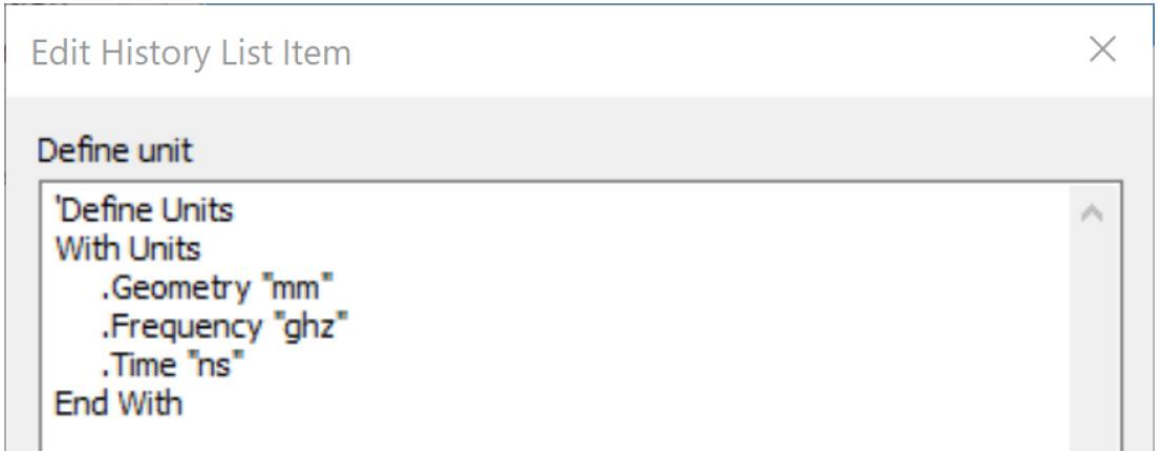
# DEFINE UNITS & FREQUENCY RANGE

- Important actions (i.e., modeling, solver setup, excitation, etc.) are recorded as VBA-Commands and their names are listed in History List.



# DEFINE UNITS & FREQUENCY RANGE

- Double clicking an item in the history list opens the VBA code.



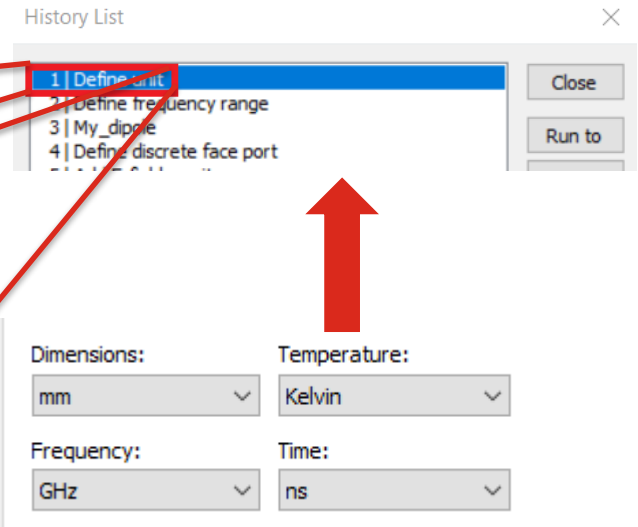
# DEFINE UNITS & FREQUENCY RANGE

- A multiline string of VBA commands is assigned to a variable, and it will be added to the History List by executing

  "`add_to_history ("Define unit", Full_History)`"

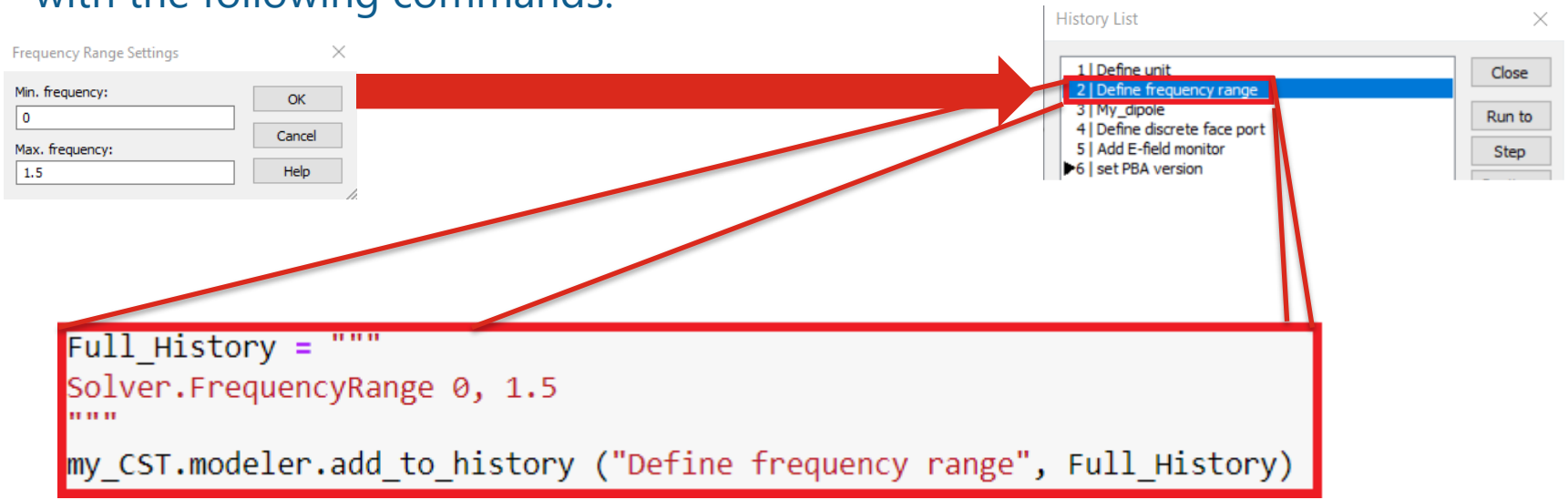
- The name of the action is "Define unit".

```
# 3.2 Define units and frequency range
Full_History = """
With Units
     .Geometry "mm"
     .Frequency "ghz"
     .Time "ns"
End With
"""
my_CST.modeler.add_to_history ("Define unit", Full_History)
```



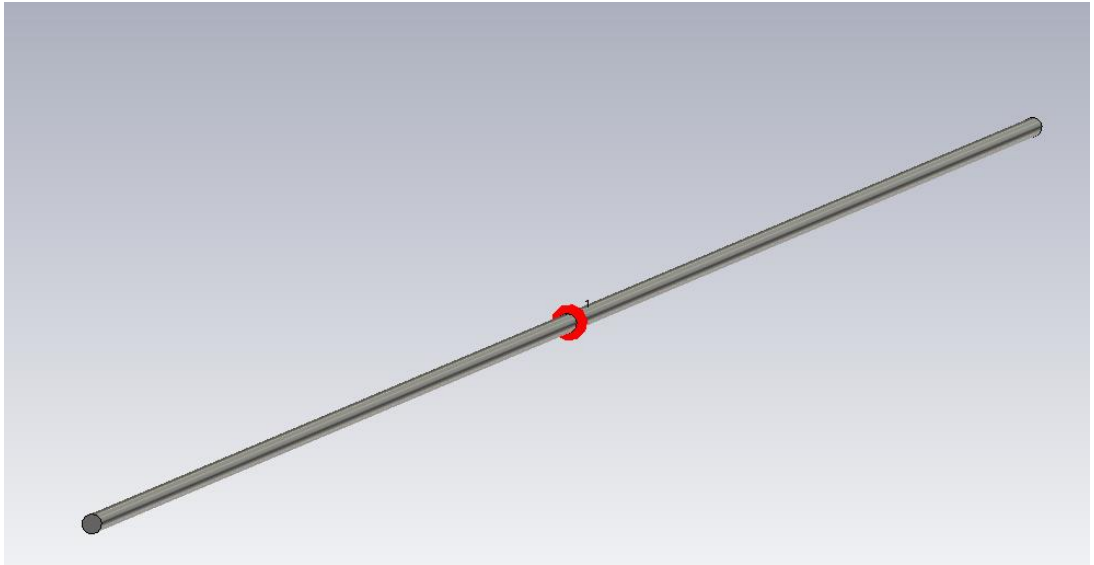
# DEFINE UNITS & FREQUENCY RANGE

- Similarly, frequency range can be defined by adding a History List Item with the following commands.

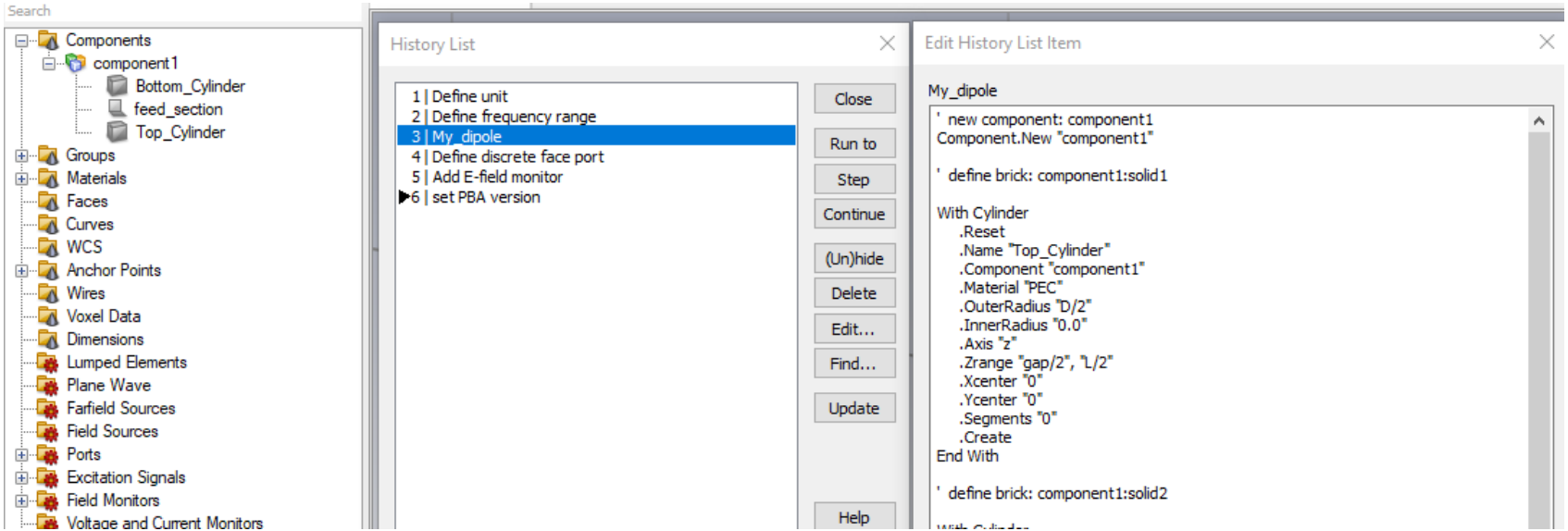


# BUILD MODEL

- Python script can be used to build the model and to add the item “My_dipole” to the History List of the CST Studio Suite®



# BUILD MODEL



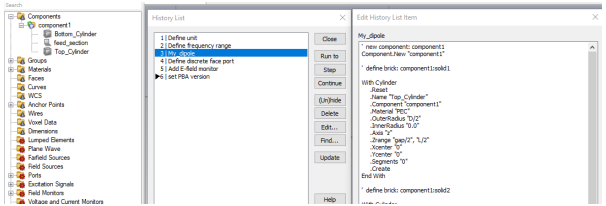





HITECH RF & Microwave solutions

DASSAULT SYSTEMES | The 3DEXPERIENCE® Company

# BUILD MODEL

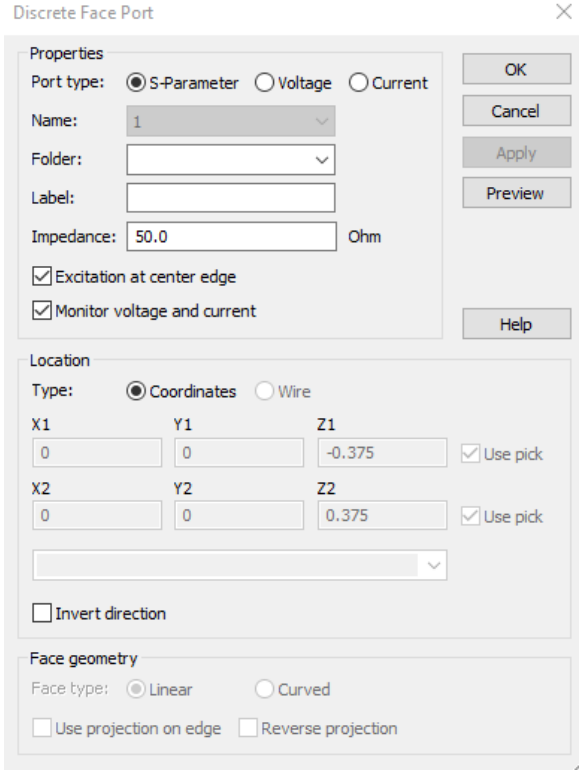


```
Full_History = """
'   new component: component1
Component.New "component1"

'   define brick: component1:solid1

With Cylinder
     .Reset
     .Name "Top_Cylinder"
     .Component "component1"
     .Material "PEC"
```

⋮

```
     .Yrange "0", "0"
     .Zrange "-gap/2", "gap/2"
     .Create
End With
"""

my_CST.modeler.add_to_history ("My_dipole", Full_History)
```

# EXCITATION




```
Full_History = """

Pick.PickEdgeFromId "component1:feed_section", "4", "4"
Pick.PickEdgeFromId "component1:feed_section", "2", "2"

'  define port: 1

With DiscretePort
     .Reset
     .PortNumber "1"
     .Type "SParameter"
     .Label ""
     .Folder ""
     .Impedance "50.0"
     .VoltagePortImpedance "0.0"
     .VoltageAmplitude "1.0"
     .CurrentAmplitude "1.0"
     .Monitor "True"
     .CenterEdge "True"
     .SetP1 "True", "0", "0", "-0.375"
     .SetP2 "True", "0", "0", "0.375"
     .LocalCoordinates "False"
     .InvertDirection "False"
     .UseProjection "False"
     .ReverseProjection "False"
     .FaceType "Linear"
     .Create
End With
"""
my_CST.modeler.add_to_history ("Define discrete face port", Full_History)
```





HITECH RF & Microwave solutions

DASSAULT SYSTEMES | The 3DEXPERIENCE® Company

# AND MORE..!!

- Field Monitors
- Mesh Properties
- Solver Selection & Simulation Run
- Access Results & Plot (0D/1D)
- Access Results & Plot (2D/3D)
- Post Processing
- Parametric Sweep
- Optimization

# AND MORE..!!

- Field Monitors
- Mesh Properties
- Solver Selection & Simulation Run
- Access Results & Plot (0D/1D)
- Access Results & Plot (2D/3D)
- Post Processing
- Parametric Sweep
- Optimization

# SOLVER & SIMULATION

- Solver Setup added in the history list.
- To check the selected solver, execute the command below and it will print the selected solver.

```
my_CST.modeler.get_active_solver_name()
```

```
'HF Time Domain'
```

- Start Simulation:

```
my_CST.modeler.run_solver()
```

# ACCESS & PLOT RESULTS

- The package `cst.result` in CST Python Libraries provides access to 0D/1D results of CST files.
- It is not necessary to open the CST Studio Suite® while using functions defined in `cst.result`
- If CST Project is already open, we can work in interactive mode to access 0D/1D results.

```
#allow_interactive=True
My_dipole = cst.results.ProjectFile(r"C:\\Users\\Kamal\\jupyter-notebooks\\My_dipole.cst", allow_interactive=True)
```

```
You are working in interactive mode.
```

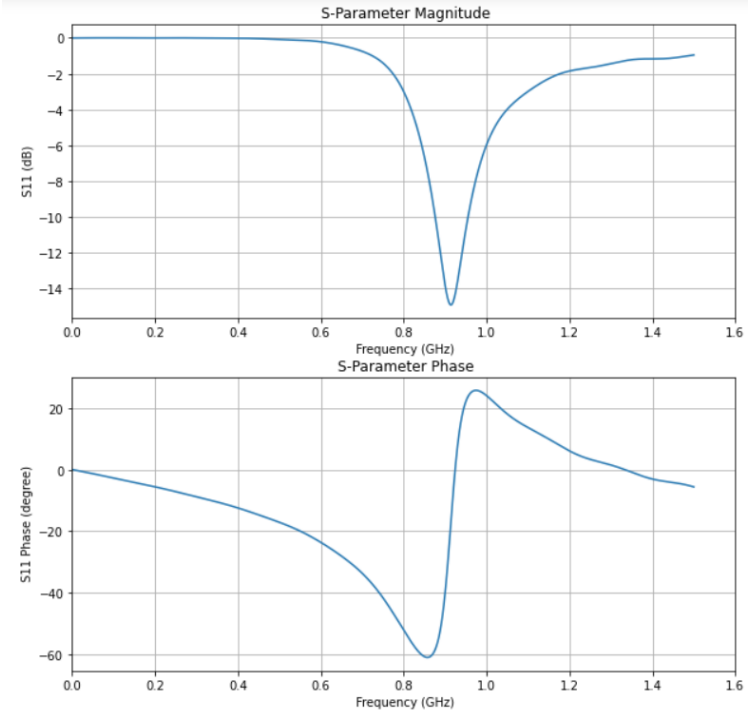
# ACCESS & PLOT RESULTS

- The code below access S11 from CST & plots it.

```
s11 = My_dipole.get_3d().get_result_item("1D Results\S-Parameters\S1,1")
ss = np.asarray([s11.get_xdata() , s11.get_ydata()])
fig1 = plt.figure(figsize=(10,10))
plt.subplot(2,1,1)
plt.plot(s11.get_xdata(),20*np.log10(np.absolute(np.asarray(s11.get_ydata()))))
plt.title(' S-Parameter Magnitude')
plt.ylabel(' S11 (dB)')
plt.xlabel(' Frequency (GHz)')
plt.grid(True)
plt.xlim((0,1.6))
plt.subplot(2, 1, 2)
plt.plot(s11.get_xdata(),np.angle(np.asarray(s11.get_ydata()),deg=True))
plt.title(' S-Parameter Phase')
plt.ylabel(' S11 Phase (degree)')
plt.xlabel(' Frequency (GHz)')
plt.grid(True)
plt.xlim((0,1.6))
```

# ACCESS & PLOT RESULTS

- S11 (dB) and phase





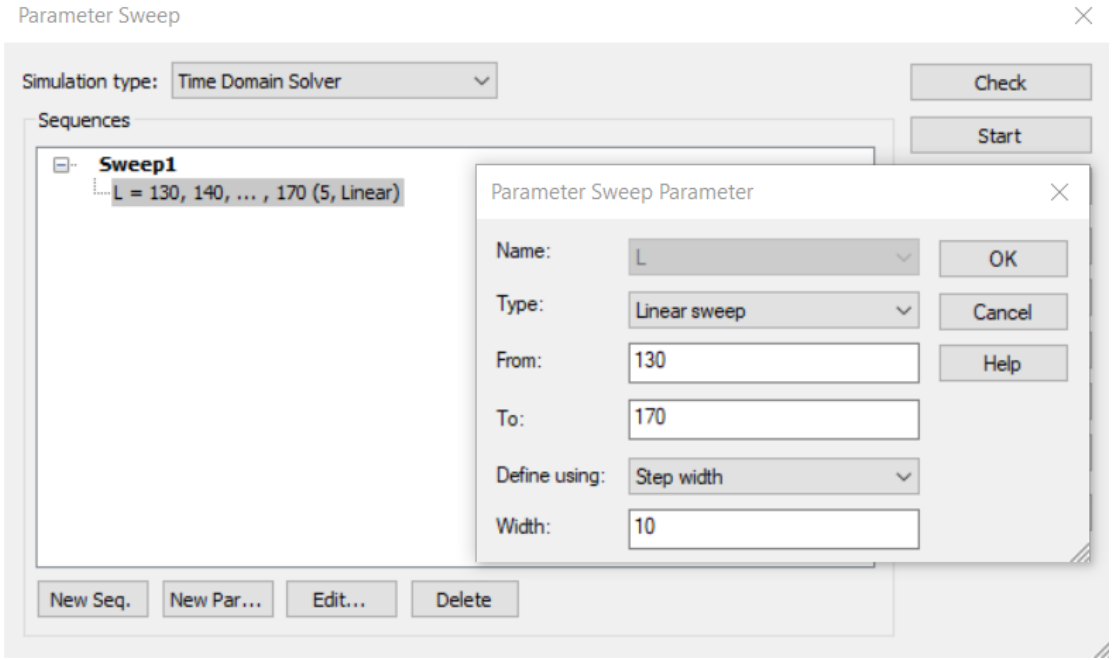



HITECH RF & Microwave solutions

DASSAULT SYSTEMES | The 3DEXPERIENCE® Company

# PARAMETRIC SWEEP

- A linear parametric sweep was performed on “L” from 130 to 170 with step width of 10








HITECH RF & Microwave solutions

DASSAULT SYSTEMES | The 3DEXPERIENCE® Company

# PARAMETRIC SWEEP

- A linear parametric sweep was performed on “L” from 130 to 170 with step width of 10

```
#Run Parametric sweep on parameter L from 130 to 170 with step width of 10
par_sweep = """
Sub Main
With ParameterSweep
    .DeleteAllSequences
    .SetSimulationType ("Transient")
    .AddSequence ("Sweep1")
    .AddParameter_Stepwidth ("Sweep1", "L", 130, 170, 10)
    .Start
End With
End Sub """
my_CST.schematic.execute_vba_code(par_sweep)
```

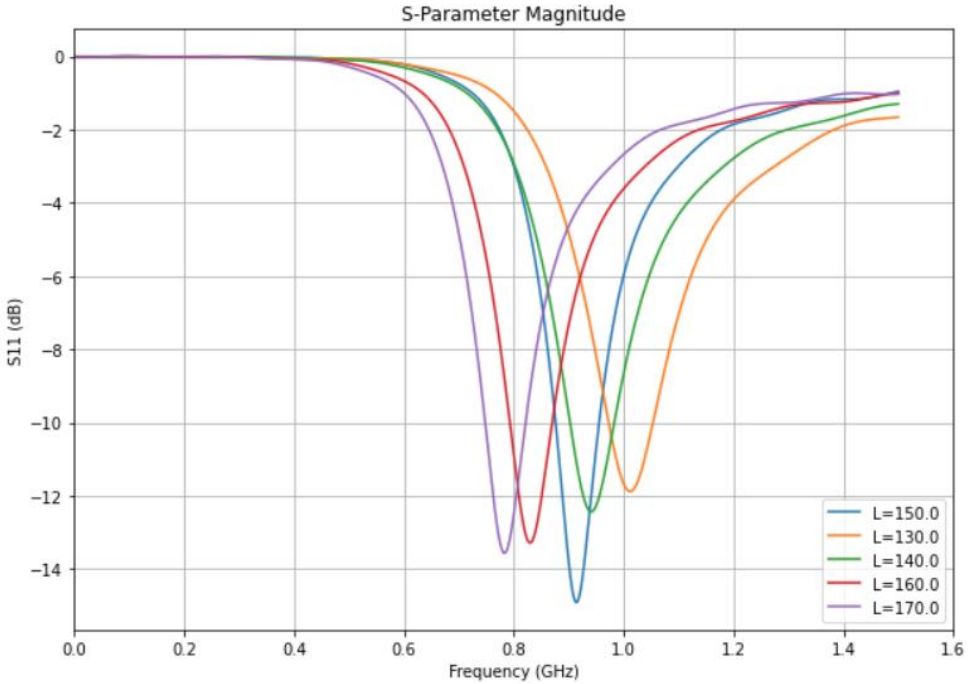
# PARAMETRIC SWEEP

- Result of different L values is plotted.

```
#Plot different values of Parameter L
run_id = My_dipole.get_3d().get_all_run_ids()
fig2 = plt.figure(figsize=(10,7))
L = [];
for id in run_id:
    s11 = My_dipole.get_3d().get_result_item("1D Results\S-Parameters\S1,1",id)
    ss = np.asarray([s11.get_xdata() , s11.get_ydata()])
    par = s11.get_parameter_combination()
    S11_mag_dB = 20*np.log10(np.absolute(np.asarray(s11.get_ydata())))
    if id!=0:
        plt.plot(s11.get_xdata(),S11_mag_dB,label="L="+str(par['L']))
plt.legend(loc='lower right')
plt.
title('S-Parameter Magnitude')
plt.ylabel('S11 (dB)')
plt.xlabel('Frequency (GHz)')
plt.grid(True)
plt.xlim((0,1.6))
```

# PARAMETRIC SWEEP

- Result of different L values is plotted.








HITECH RF & Microwave solutions

DASSAULT SYSTEMES | The 3DEXPERIENCE® Company

# FURTHER WORK

- Access Results & Plot (2D/3D) – hdf5 file (.h5)
- Optimization

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Input In [120], in <cell line: 67>()
      1 #set up the optimization settings and start the optimizer
      3 StartOptimizer = """
      4 Sub Main
      5
   (...)
     65 End With
     66 End Sub"""
---> 67 my_CST.schematic.execute_vba_code(StartOptimizer)

RuntimeError: An error occurred while trying to execute execute_vba_code:
Error in VBA code:
```

# FOR HELP

- More commands about parameter handling can be found in CST Studio Suite 2022 Help → Automation and Scripting → Visual Basic (VBA) → 3D Simulation VBA → VBA Objects

# DEMO!

# QUESTIONS?