

第8章 用PyAEDT完成完整模擬過程

8.1 雙極子天線

雙極子天線 (Dipole Antenna) 確實是一個非常基本且重要的天線結構，在無線通信和電磁學領域中被廣泛研究和應用。它由兩根金屬導體組成，這兩根導體通常是相等的長度，並從它們的中心點以相對方向延伸。

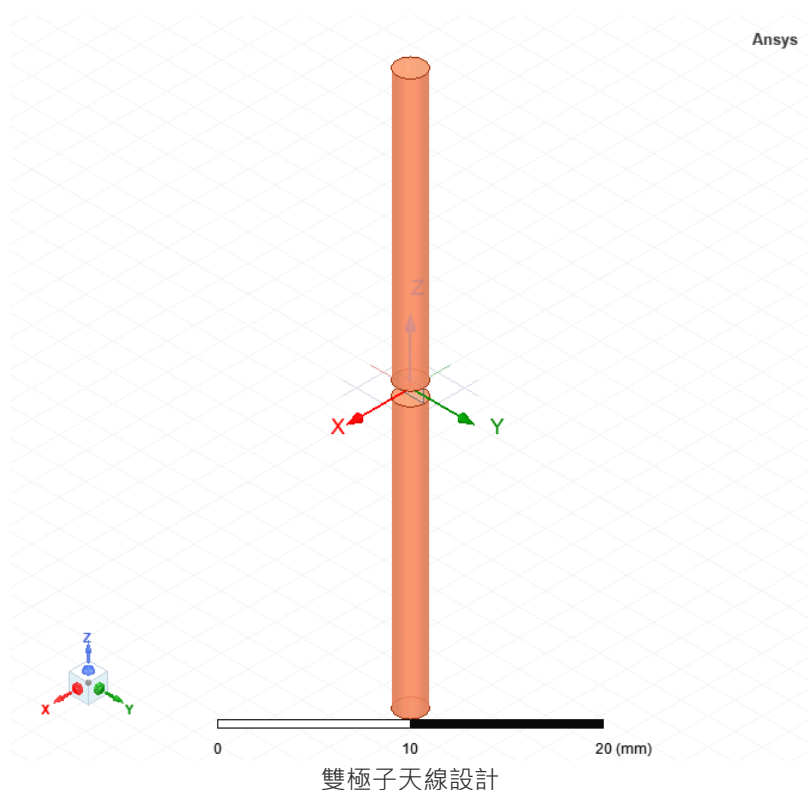
對於使用PyAEDT的初學者來說，雙極子天線是一個極好的學習範例。PyAEDT提供了一個強大的界面，可以用來在AEDT當中創建、分析和優化各種電磁組件和系統設計。

模擬流程

這段代碼是利用 Python 語言結合了 pyaedt 模塊來操作 HFSS 軟件進行電磁場模擬的一個完整流程。主要步驟包括：

1. **導入 HFSS 模塊和初始化**：首先導入 pyaedt 中的 Hfss 模塊，並初始化指定版本的 HFSS 環境。
2. **材料和參數設置**：修改材料設定，使用因果材料，並設置模擬所需的幾何參數。
3. **創建幾何體**：利用 HFSS 模型器 (modeler) 創建所需的幾何體，如銅製圓柱和矩形片等。
4. **端口和邊界設定**：在矩形片上創建端口，並設定開放區域，這些是模擬中的關鍵部分，用於定義模擬的邊界條件。
5. **模擬設定**：創建並配置模擬設定，包括設定模擬頻率、最大迭代次數等。
6. **頻率掃描設定**：設定一個線性步進頻率掃描，以獲得在不同頻率下的模擬結果。
7. **數據繪圖和分析**：使用 matplotlib 庫進行數據的可視化，如繪製反射損耗 (return loss) 與頻率的關係圖。
8. **執行模擬和結果處理**：對不同參數下的模型進行模擬，獲取並分析模擬結果，如計算最小 S11 值和相應的頻率。
9. **圖表保存和 HFSS 關閉**：最後將獲得的圖表顯示出來並保存，然後關閉 HFSS 環境。

總的來說，這段代碼展示了如何在 Python 環境中利用 HFSS 進行雙極子天線建模電磁場模擬，從幾何模型創建到模擬設置，再到數據分析和結果展示，形成了一個完整的模擬流程。這對於從事電磁學、射頻設計或相關領域的工程師和研究人員來說非常有用。



雙極子天線是一種基本的無線電天線，它由兩個相同的導體元素組成，如金屬線或棒。這種天線的輻射模式類似於基本的電偶極子，並支持一條線電流，這個電流在每端只有一個節點。二極天線通常用於無線電通信中，由於其結構簡單、應用廣泛，它成為了最早使用的一類天線。詳細資訊可以在Wikipedia的相關頁面找到：[Dipole Antenna on Wikipedia](#)。

對於一個二極天線來說，當天線的長度增加時，它的共振頻率會降低。這是因為天線的物理尺寸與其輻射的波長相關。理想情況下，一個二極天線的長度應該是其工作頻率波長的一半。因此，當天線的長度增加時，它能夠有效地工作的波長也會增加，這意味著其共振頻率會降低。這一原理適用於各種天線設計。

完整pyAEDT程式碼

```
# 導入 pyaedt 的 Hfss 模塊
from pyaedt import Hfss

# 初始化 HFSS 版本為 2022.2
hfss = Hfss(specified_version='2022.2')

# 修改材料設定
hfss.change_material_override()

# 自動使用因果材料
hfss.change_automatically_use_causal_materials()

# 設定 'length' 參數為 '10mm'
hfss['length'] = '10mm'

# 創建一個朝 Z 軸的銅製圓柱
hfss.modeler.create_cylinder(cs_axis='Z',
                             position=(0,0,0.5),
                             radius='0.5mm',
                             height='length',
                             matname='copper'
                             )

# 創建另一個朝 Z 軸的銅製圓柱
hfss.modeler.create_cylinder(cs_axis='Z',
                             position=(0,0,-0.5),
                             radius='0.5mm',
                             height='-length',
                             matname='copper'
                             )

# 創建一個矩形片
hfss.modeler.create_rectangle(csPlane=1,
                              position=(-0.5, 0, -0.5),
                              dimension_list=(1, 1),
                              name='sheet')

# 在矩形片上創建一個端口
hfss.create_lumped_port_to_sheet(sheet_name='sheet', axisdir=2)

# 創建一個開放區域
hfss.create_open_region(Frequency='1GHz', )

# 創建一個模擬設定
setup = hfss.create_setup('mysetup')

# 設定模擬參數
setup.props['Frequency'] = '2GHz'
setup.props['MaxDeltaS'] = 0.02
setup.props['MaximumPasses'] = 20

# 創建一個線性步進頻率掃描
hfss.create_linear_step_sweep(setupname='mysetup',
                              unit='GHz',
                              freqstart=0.1,
                              freqstop=2,
```

```

        step_size = 0.1,
        sweepname='mysweep',)

# 導入 matplotlib 並設定圖表參數
import matplotlib.pyplot as plt
plt.grid()
plt.title('dipole antenna return loss')
plt.xlabel('freq (GHz)')
plt.ylabel('dB(S11)')

# 針對不同的長度進行模擬
for l in [80, 90, 100]:
    hfss['length'] = '{}mm'.format(l)
    hfss.analyze_nominal(num_cores=4)

# 獲取模擬結果
result = hfss.post.get_solution_data('dB(S11)', 'mysetup:mysweep')
s11 = result.data_real('dB(S11)')
freq = result.primary_sweep_values

# 找出最小的 S11 和對應的頻率
mins11, f0 = min(zip(s11, freq))

# 繪制結果
plt.text(f0, mins11, '{}GHz, {:.2f}dB'.format(f0, mins11))
plt.plot(freq, s11, c='r')

# 顯示和保存圖表
plt.show()
plt.savefig('c:/s11_.png')

# 關閉 AEDT 桌面
hfss.close_desktop()

```

初始化HFSS設計

這段代碼的開頭部分是設置 HFSS 模擬環境的基礎步驟，主要包括以下幾個關鍵部分：

1. 導入 HFSS 模塊：

- `from pyaedt import Hfss`：這一行代碼從 `pyaedt` 庫中導入了 `Hfss` 類。`pyaedt` 是一個 Python 模塊，用於操作 Ansys Electronic Desktop (AEDT) 中的各種模擬工具，包括 HFSS。

2. 初始化 HFSS 環境：

- `hfss = Hfss(specified_version='2024.1')`：這行代碼創建了一個 HFSS 對象 `hfss`，並指定了使用的 HFSS 版本為 2024.1。這樣，就可以在 Python 環境中便可以連結並控制 HFSS 進行電磁場模擬了。

3. 修改材料設定：

- `hfss.change_material_override()`：此行代碼允許對 HFSS 中的材料屬性進行自定義修改。在電磁場模擬中，材料的電磁屬性（如介電常數、導電率等）對模擬結果影響很大，因此能夠根據需要調整這些屬性是非常重要的。

4. 自動使用因果材料：

- `hfss.change_automatically_use_causal_materials()`：這行代碼設定 HFSS 在模擬中自動使用因果材料。所謂“因果材料”（Causal Materials），是指其電磁響應遵循因果關係的材料，這是現實物理世界中的普遍規律。在電磁模擬中，正確地處理材料的因果關係對於獲得準確的模擬結果至關重要。

在使用腳本進行模擬工作時，若不進行初始化設定，將可能遇到一系列問題，特別是當每台電腦上的模擬預設定值有可能不相同。這會對數據的一致性和可靠性造成影響，特別是在需要精確計算或模擬的情況下。

建立3D模型

在以下的代碼片段中，使用了modeler物件底下的接口方法來創建幾何形狀。讓我們逐一解釋每個函數的功能和參數：

1. 設定參數 'length'：

```
hfss['length'] = '10mm'
```

這行代碼設置了一個名為 `length` 的參數，其值為 '10mm'。在 HFSS 中，這樣的參數常用於定義幾何尺寸或其他屬性，並可以在多處重複使用。

2. 創建朝 Z 軸的銅製圓柱：

```
hfss.modeler.create_cylinder(cs_axis='Z', position=(0,0,0.5), radius='0.5mm', height='length', matname='copper')
```

這個函數用於創建一個圓柱形幾何物體。函數的參數解釋如下：

- `cs_axis='Z'`：指定圓柱的對稱軸是 Z 軸。
- `position=(0,0,0.5)`：設置圓柱底面中心的位置，這裡是 (0, 0, 0.5)。
- `radius='0.5mm'`：圓柱的半徑設為 0.5 毫米。
- `height='length'`：圓柱的高度。這裡使用了之前定義的 `length` 參數，即 10 毫米。
- `matname='copper'`：指定圓柱的材質為銅。

3. 創建另一個朝 Z 軸的銅製圓柱：

```
hfss.modeler.create_cylinder(cs_axis='Z', position=(0,0,-0.5), radius='0.5mm', height='-length', matname='copper')
```

這個函數與上面的類似，但有些參數不同：

- `position=(0,0,-0.5)`：這裡圓柱底面中心的位置在 (0, 0, -0.5)。
- `height='-length'`：高度設為 '-length'，意味著圓柱的方向相反，但高度依然是 10 毫米。

4. 創建一個矩形片：

```
hfss.modeler.create_rectangle(csPlane=1, position=(-0.5, 0, -0.5), dimension_list=(1, 1), name='sheet')
```

此函數用於創建一個矩形片。參數詳解如下：

- `csPlane=1`：選擇創建矩形的平面。這裡 `1` 通常代表 XY 平面。
- `position=(-0.5, 0, -0.5)`：矩形左下角的位置。
- `dimension_list=(1, 1)`：矩形的尺寸，這裡表示長度和寬度都是 1 單位。
- `name='sheet'`：為這個矩形指定一個名稱，這裡是 'sheet'。

總之，這些函數和參數共同作用於 HFSS 中，用於創建和定義特定的幾何形狀和特性。這對於進行高頻模擬和分析是非常重要的。

建立端口與邊界條件

代碼片段創建端口和開放區域，這是模擬高頻電磁場時的常見步驟。讓我們逐一解釋這些函數的功能和參數：

1. 在矩形片上創建一個端口：

```
hfss.create_lumped_port_to_sheet(sheet_name='sheet', axisdir=2)
```

這個函數用於在指定的矩形片上創建一個稱為 "lumped port" 的端口。端口是用來模擬電磁波在結構中的進入點或離開點。函數的參數解釋如下：

- `sheet_name='sheet'`：這指定了要在其上創建端口的矩形片的名稱。在這個例子中，名為 'sheet'，即之前創建的矩形片。
- `axisdir=2`：這指定了端口的方向。在 HFSS 中，不同的數字代表不同的軸方向。通常，0、1、2 分別代表 X、Y、Z 軸。

2. 創建一個開放區域：

```
hfss.create_open_region(Frequency='1GHz')
```

這個函數用於創建一個開放區域，這對於高頻模擬來說非常重要，尤其是在涉及天線或其他輻射系統時。開放區域允許電磁波自由進出模擬區域，而不會被任何人為的邊界條件反射或吸收。函數的參數解釋如下：

- `Frequency='1GHz'`：這指定了開放區域的工作頻率。在這個例子中，設定為 1 GHz。根據這個頻率，HFSS 會計算出合適的區域尺寸，以確保電磁波可以正確地模擬。

建立模擬設定條件

這段代碼是在 HFSS 電磁場模擬過程中創建和設定模擬參數以及定義頻率掃描的步驟。具體來說：

1. 創建一個模擬設定：

- `setup = hfss.create_setup('mysetup')`：這行代碼使用 `create_setup` 方法創建了一個新的模擬設定，並將其命名為 'mysetup'。這個設定將包含所有必要的模擬參數和選項。

2. 設定模擬參數：

- `setup.props['Frequency'] = '2GHz'`：這裡設定了模擬的頻率為 2 GHz。這意味著模擬將在 2 GHz 頻率下進行。
- `setup.props['MaxDeltaS'] = 0.02`：設定了模擬的最大 S 參數變化 (MaxDeltaS) 為 0.02。這是一個收斂標準，用於確定模擬何時停止。
- `setup.props['MaximumPasses'] = 20`：這設定了模擬的最大迭代次數為 20。如果在達到這個迭代次數之前模擬沒有收斂，則模擬將停止。

3. 創建一個線性步進頻率掃描：

- 這部分代碼創建了一個線性步進頻率掃描，用於在指定的頻率範圍內分析模型的響應。
- `hfss.create_linear_step_sweep(setupname='mysetup', unit='GHz', freqstart=0.1, freqstop=2, step_size=0.1, sweepname=)`：這裡設定了掃描的起始頻率為 0.1 GHz，終止頻率為 2 GHz，步進大小為 0.1 GHz，並將這個掃描命名為 'mysweep'。這意味著模擬將在 0.1 GHz 到 2 GHz 的範圍內，每隔 0.1 GHz 頻率進行一次測量。

總結來說，這段代碼為 HFSS 模擬設定了基本的模擬參數和一個線性頻率掃描。這樣的設定對於分析組件在不同頻率下的響應非常重要，尤其是在天線設計、射頻元件分析等領域中。

抓取資料並繪圖

這段代碼主要涉及到使用 matplotlib 庫對 HFSS 模擬結果進行可視化分析，以及針對不同參數設置（在這裡是不同的長度）進行模擬和結果處理。具體步驟如下：

1. 導入 matplotlib 並設定圖表參數：

- `import matplotlib.pyplot as plt`：首先導入 matplotlib 的 pyplot 模塊，這是 Python 中常用的繪圖工具。
- `plt.grid()`：在圖表中添加網格線，方便閱讀數據。
- `plt.title('dipole antenna return loss')`：設定圖表的標題，這裡的標題是 "dipole antenna return loss"（偶極天線的回波損失）。

- `plt.xlabel('freq (GHz)')` 和 `plt.ylabel('dB(S11)')`：分別設定 x 軸和 y 軸的標籤，x 軸是頻率（單位：GHz），y 軸是 S11 參數的分貝值。

2. 針對不同的長度進行模擬：

- 代碼中使用了一個 for 循環 `for l in [80, 90, 100]:`，分別對長度為 80mm、90mm、100mm 的情況進行模擬。
- `hfss['length'] = '{}mm'.format(l)`：這行代碼設定了模擬的物體長度。
- `hfss.analyze_nominal(num_cores=4)`：執行模擬，使用 4 個核心進行計算。

3. 獲取模擬結果並處理數據：

- `result = hfss.post.get_solution_data('dB(S11)', 'mysetup:mysweep')`：從模擬中獲取 S11 參數的數據。
- `s11 = result.data_real('dB(S11)')` 和 `freq = result.primary_sweep_values`：提取 S11 參數的真實部分和對應的頻率值。

4. 找出最小的 S11 和對應的頻率，並繪製結果：

- 使用 `min(zip(s11, freq))` 找出最小的 S11 值及其對應的頻率。
- 使用 `plt.text` 和 `plt.plot` 將這些信息添加到圖表中。

5. 顯示和保存圖表：

- `plt.show()`：顯示圖表。
- `plt.savefig('c:/s11_.png')`：將圖表保存到指定的路徑。

總之，這段代碼展示了如何對 HFSS 模擬結果進行數據處理和可視化分析，特別是在評估不同幾何參數對模擬結果的影響方面。這在天線設計和射頻元件分析等領域非常有用。

關閉AEDT

`hfss.close_desktop()` 是一個在 HFSS (High Frequency Structure Simulator) 腳本中使用的函數，其主要目的是關閉 HFSS 應用程序及其桌面環境。當你在使用腳本或自動化流程操作 HFSS 並完成所有模擬、分析或其他任務後，這個函數可以被調用來正確地關閉應用程序。