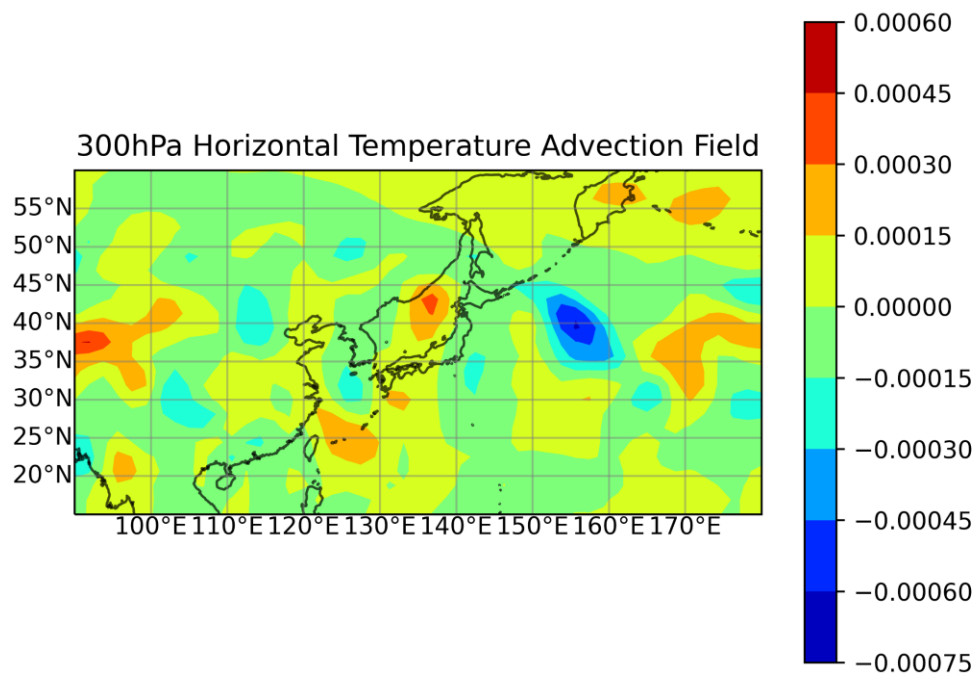


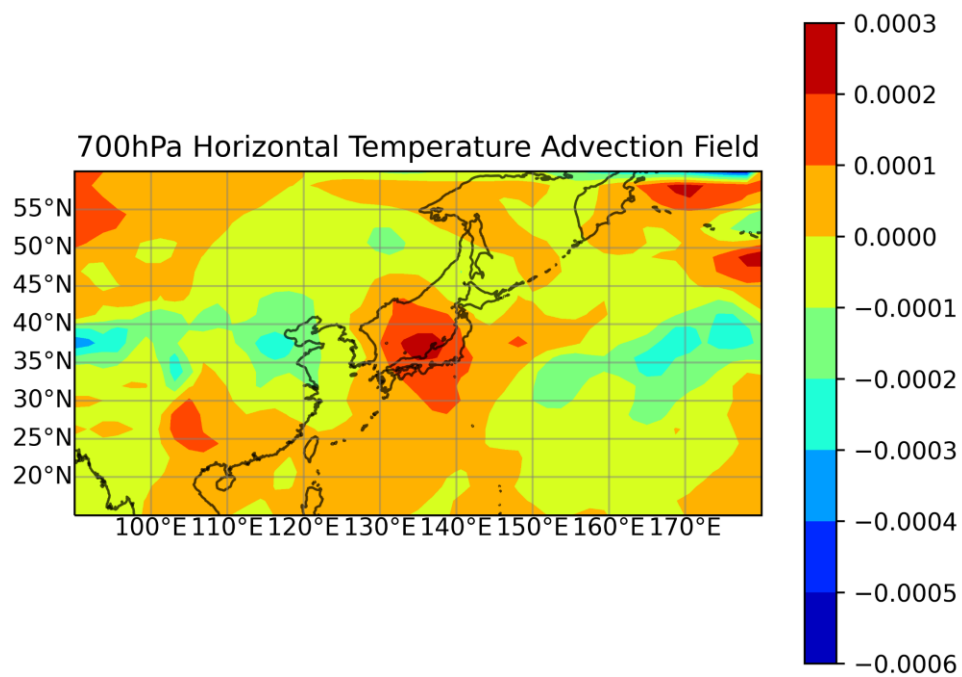
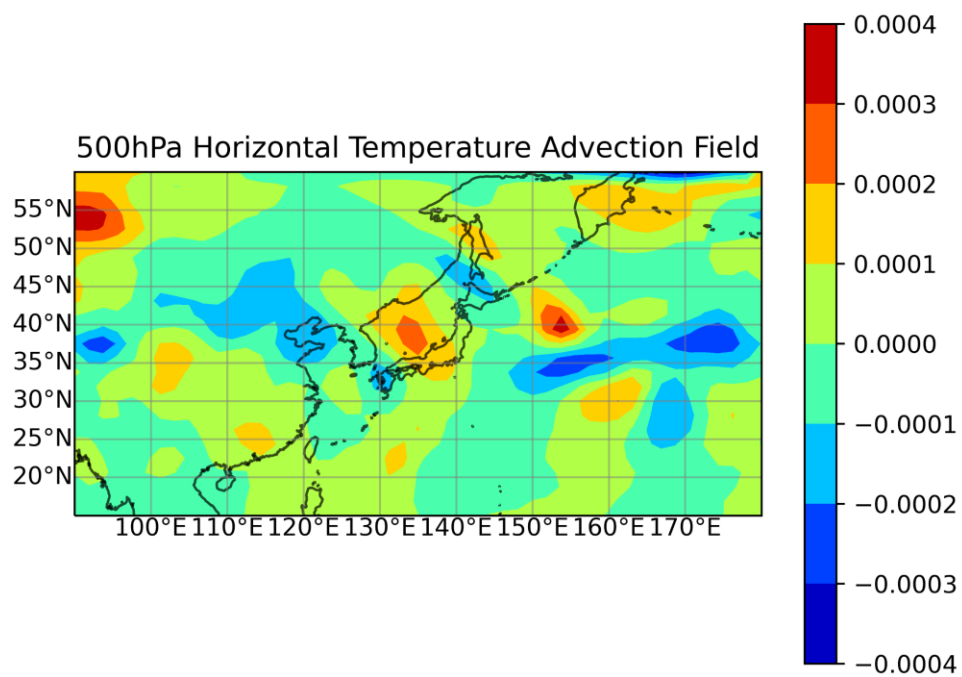
作業系統：x86 Windows10 · conda env

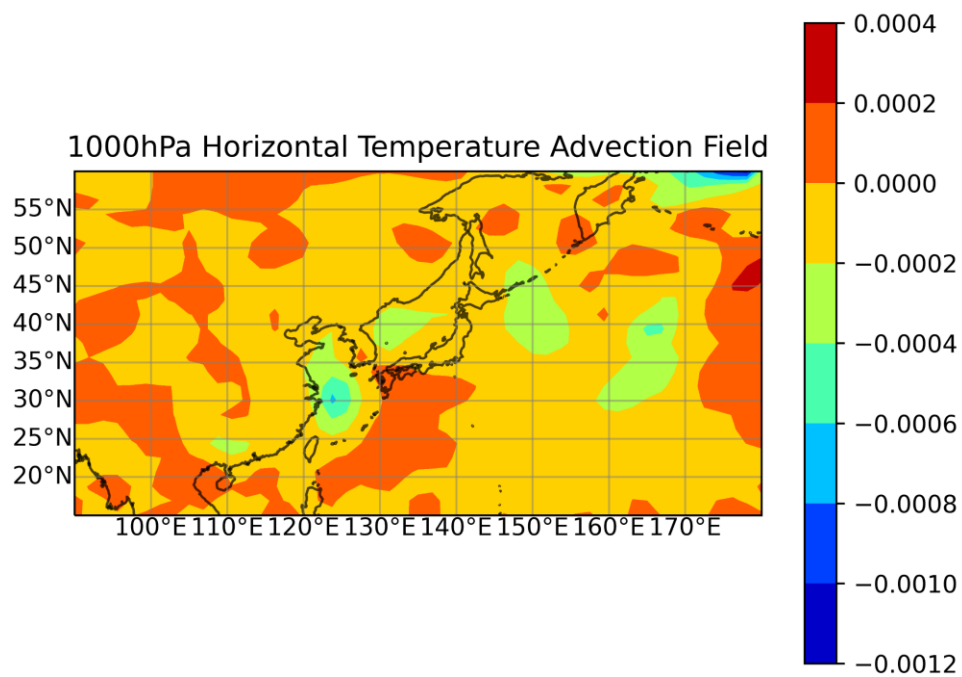
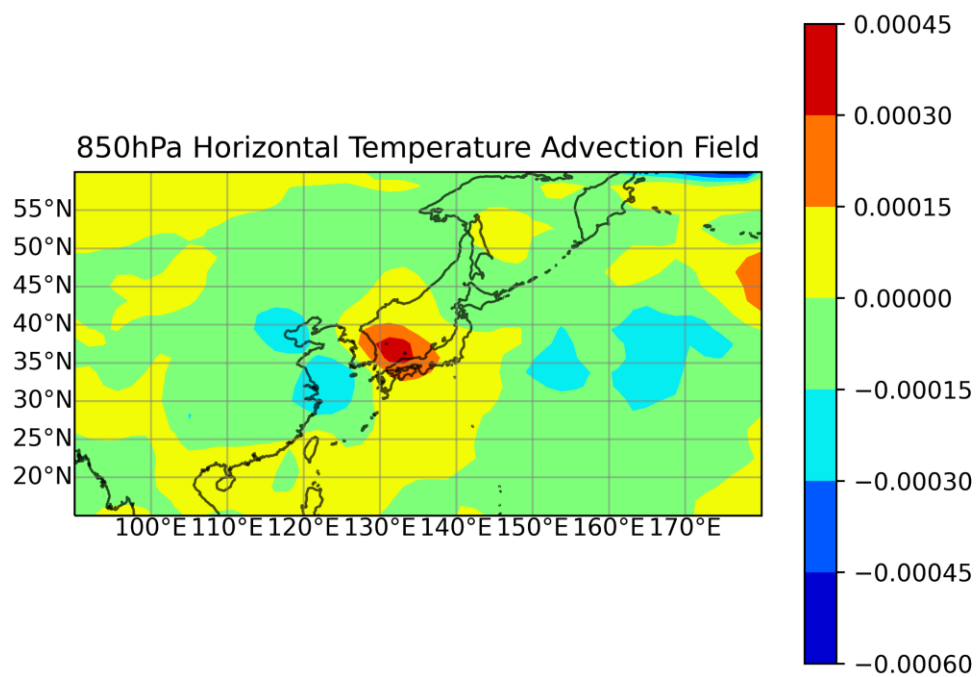
資料順序如下：

- (1) 水平溫度平流 * 5 層 (高到低)
- (2) 散度 * 5 層 (高到低)
- (3) 相對渦度 * 5 層 (高到低)
- (4) 絕對渦度平流 * 5 層 (高到低)
- (5) 計算與繪圖程式碼 + 註解

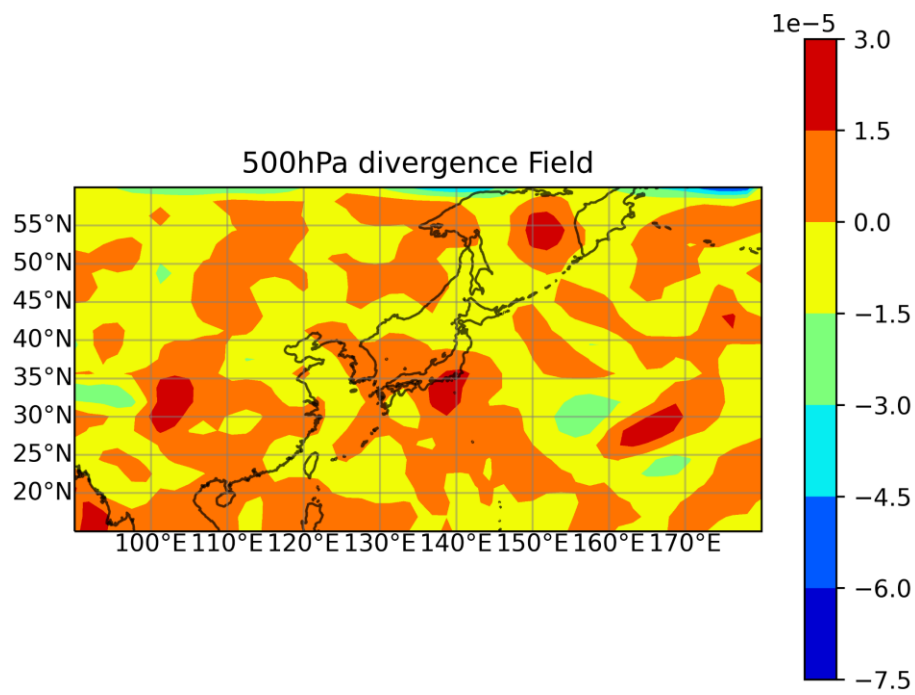
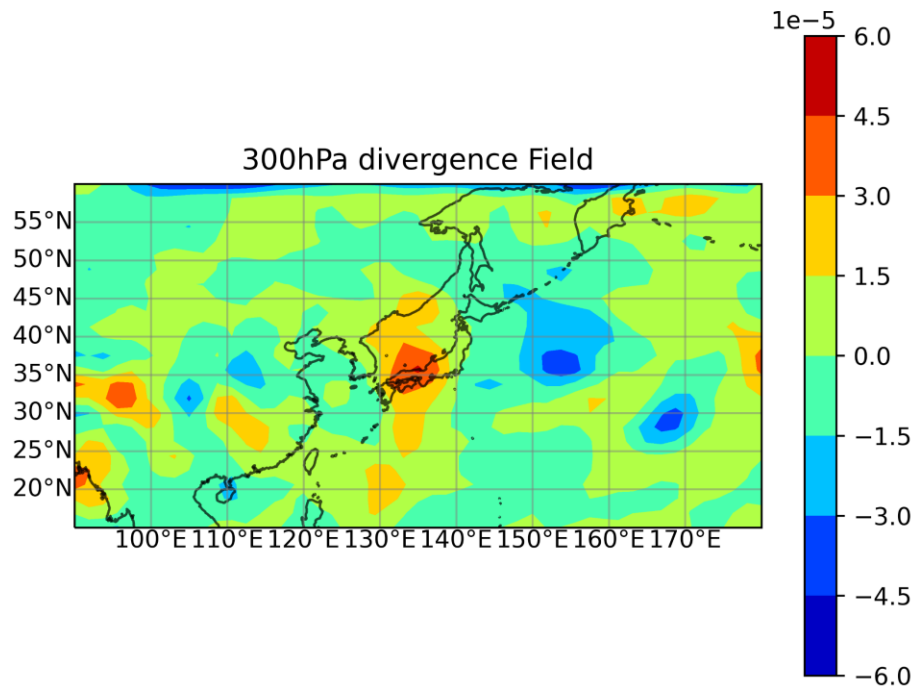
水平溫度平流 * 5 層 (高到低)

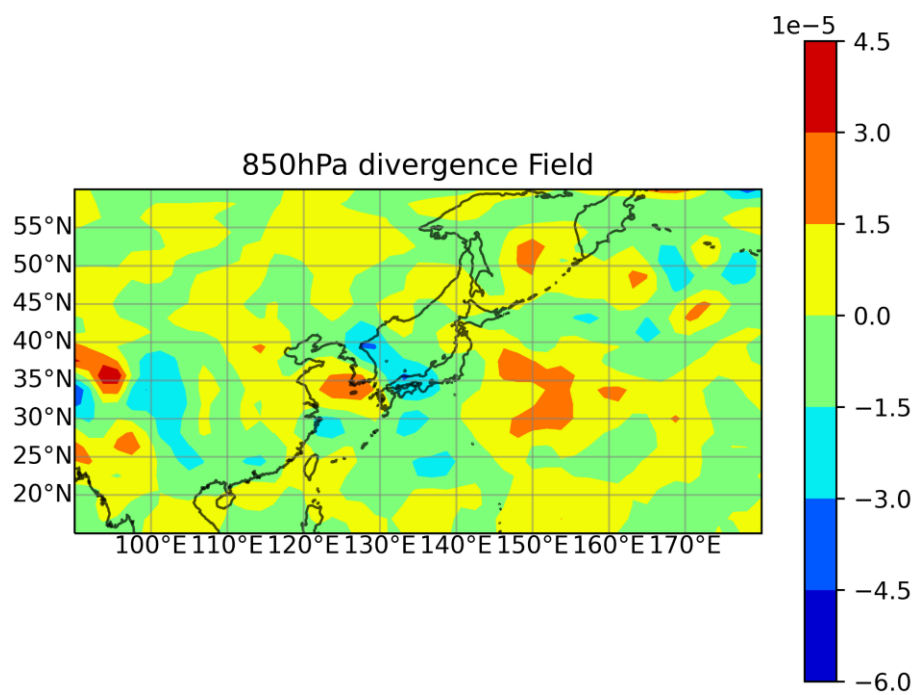
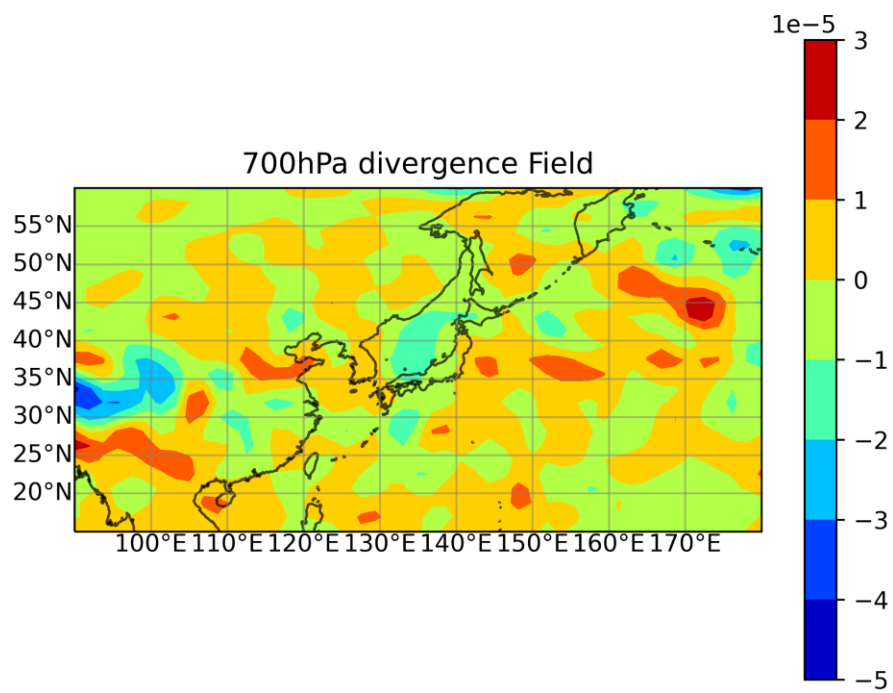


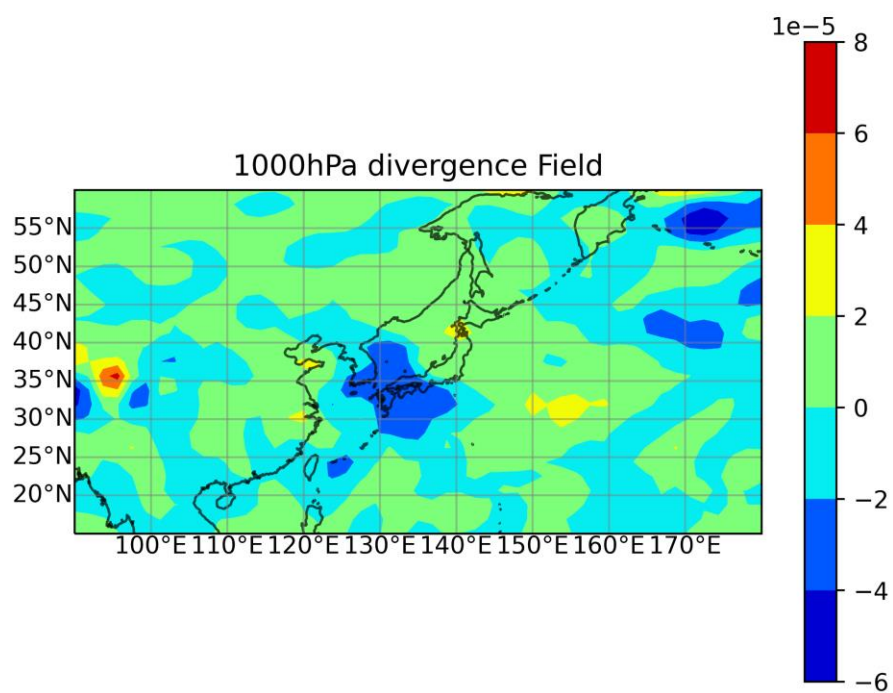




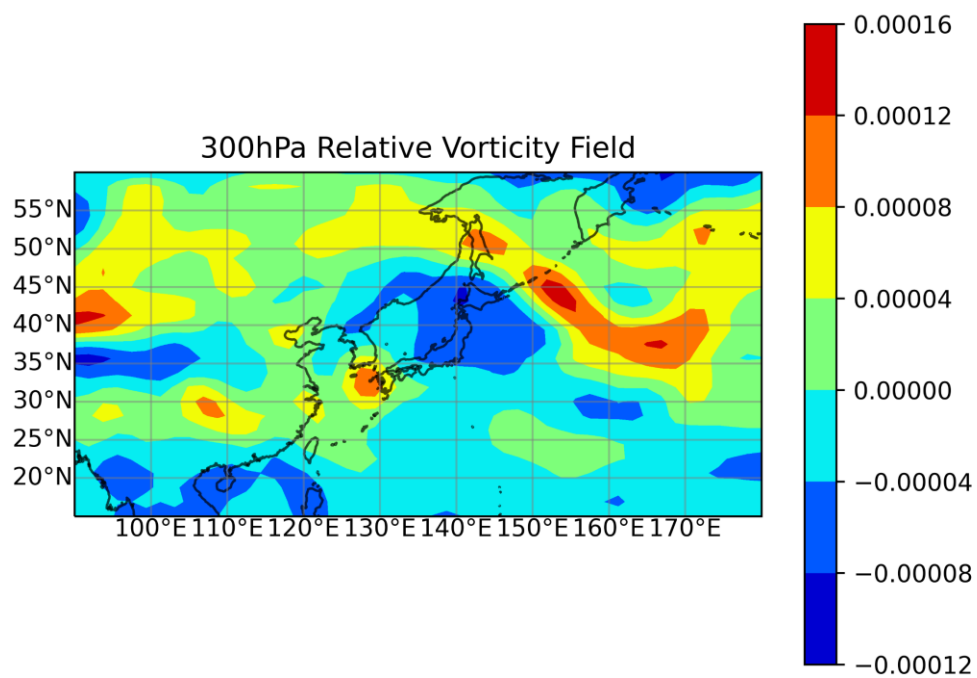
散度 * 5 層 (高到低)

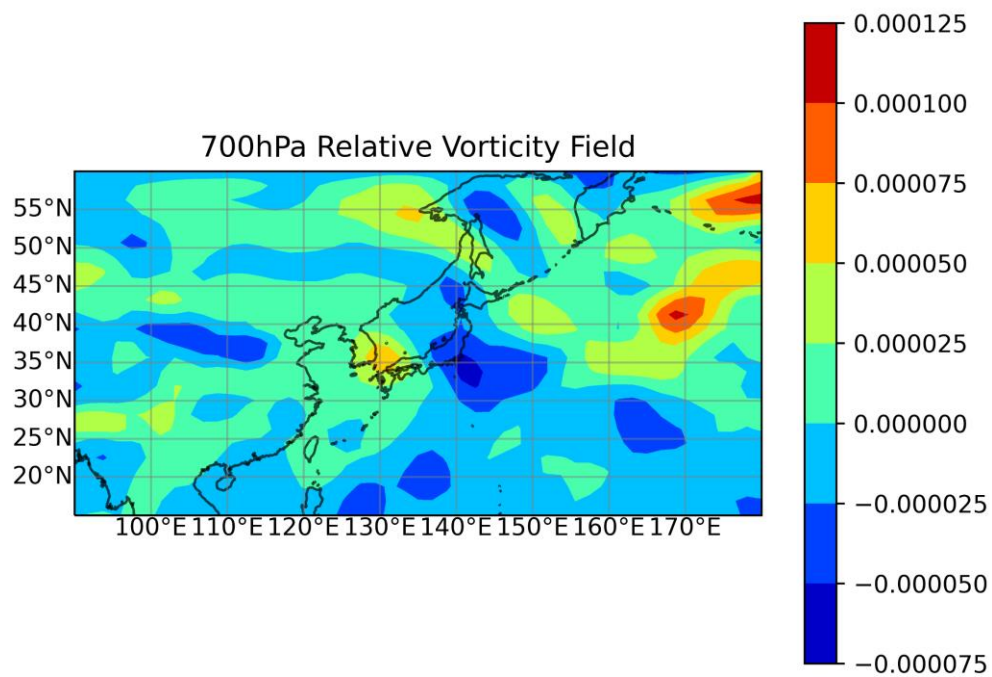
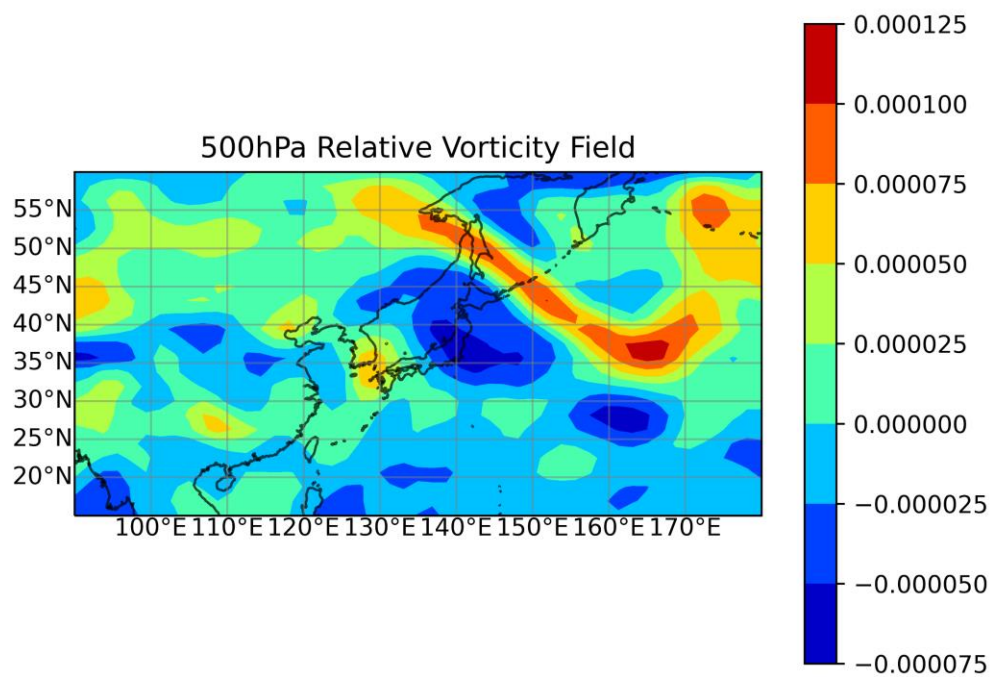


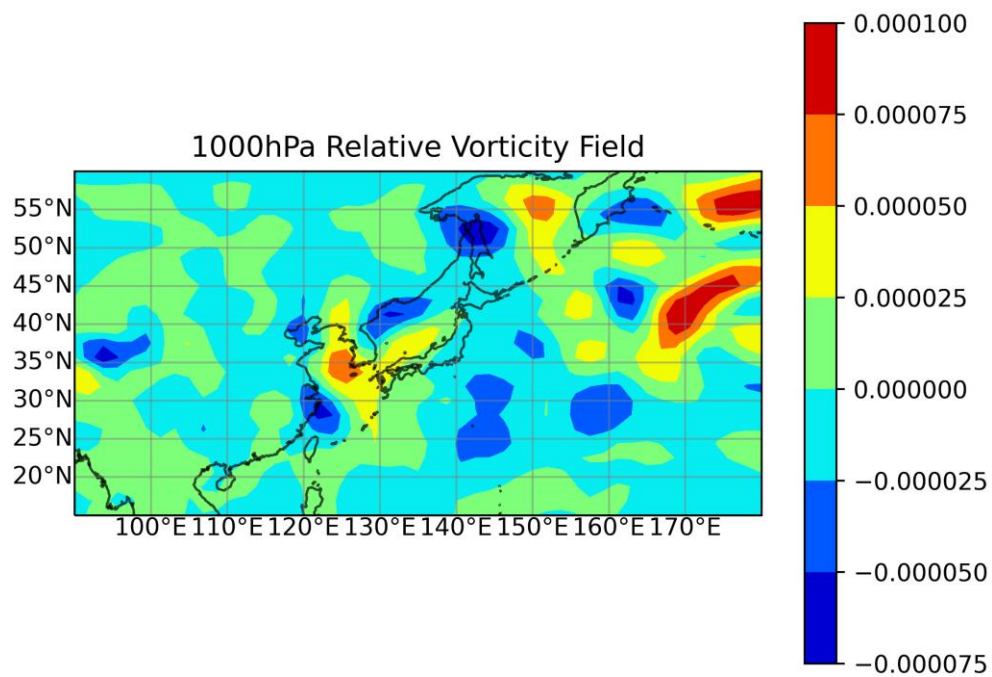
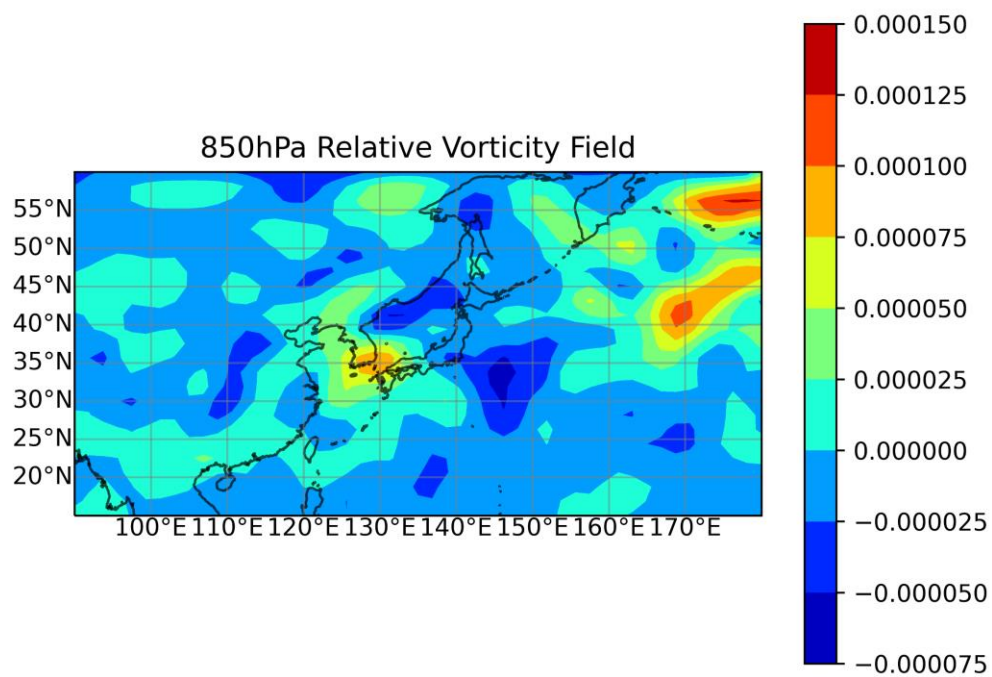




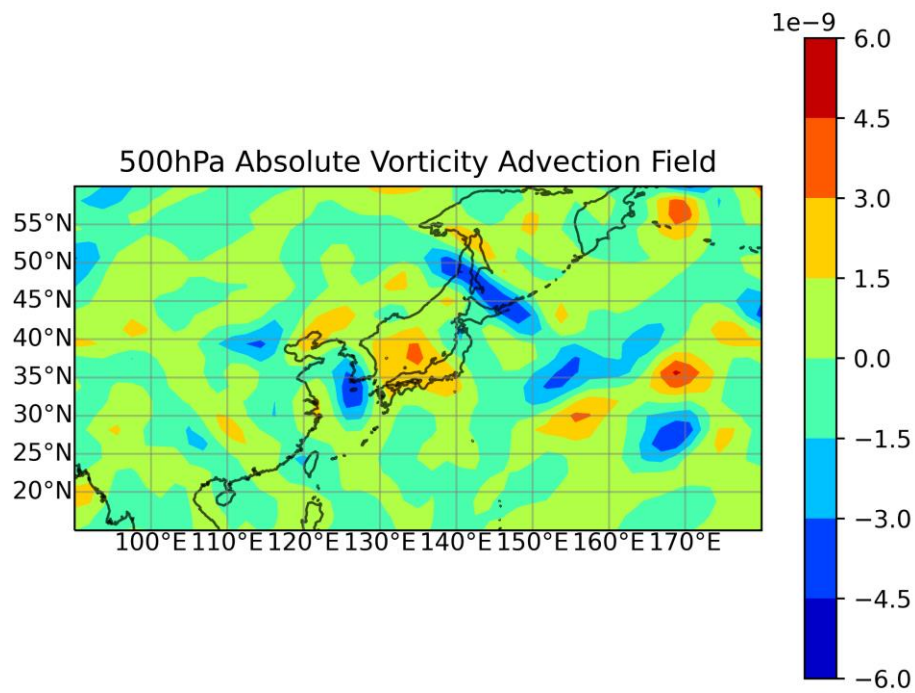
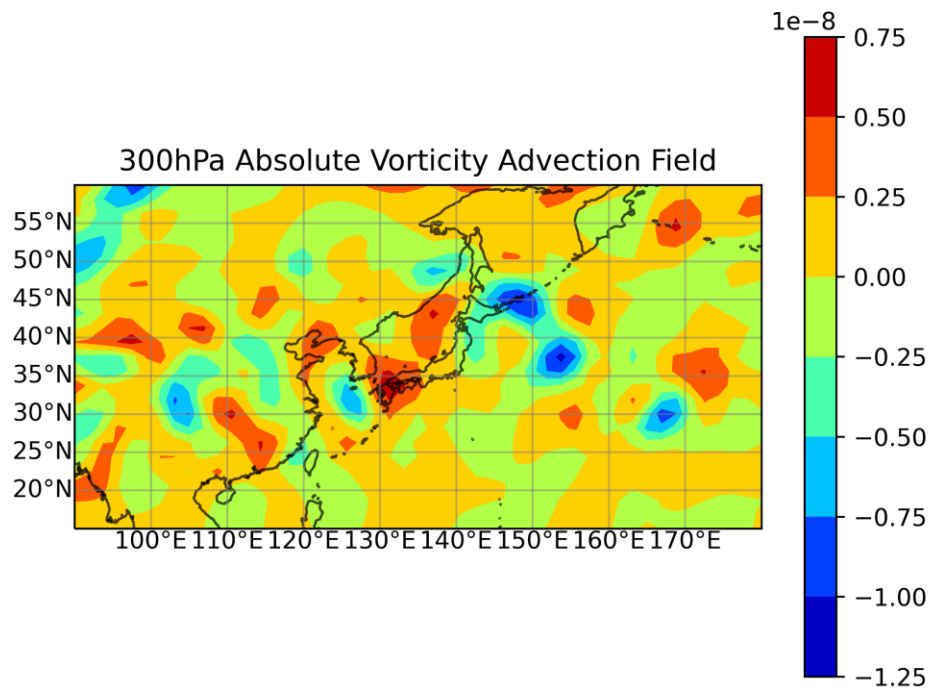
相對渦度 * 5 層 (高到低)

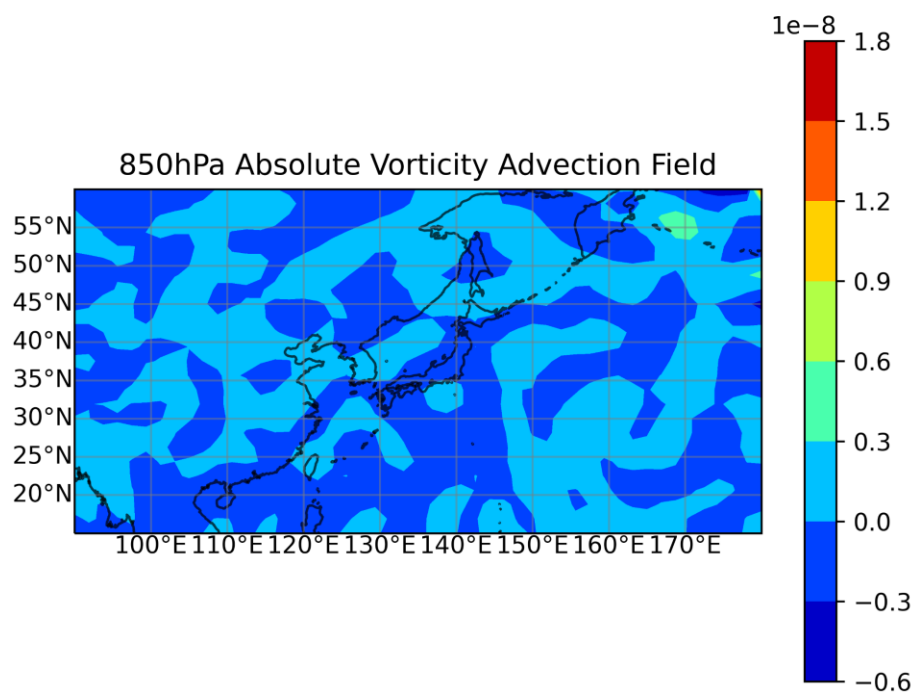
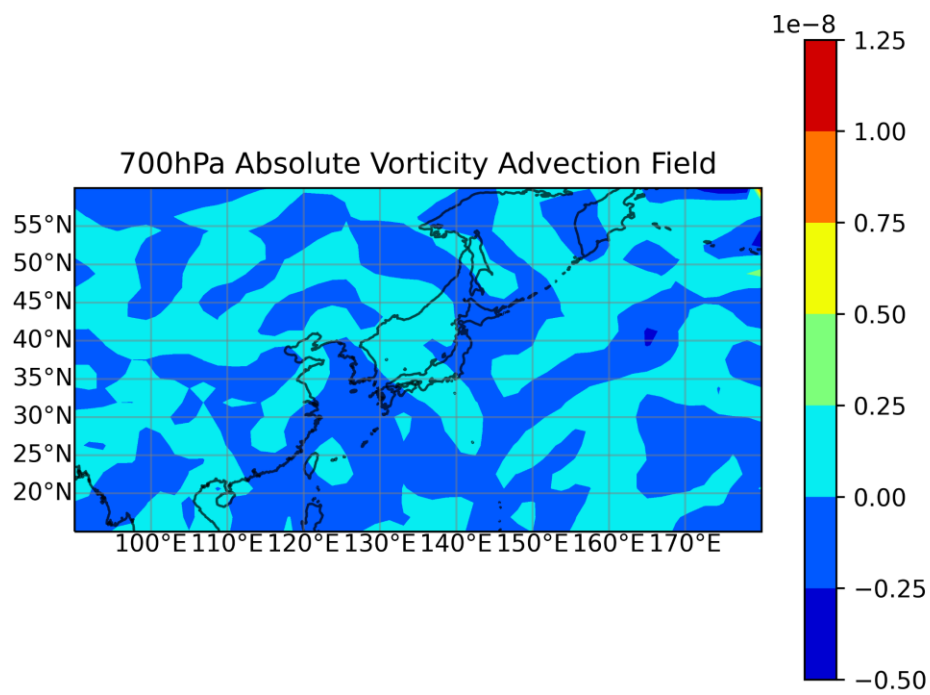


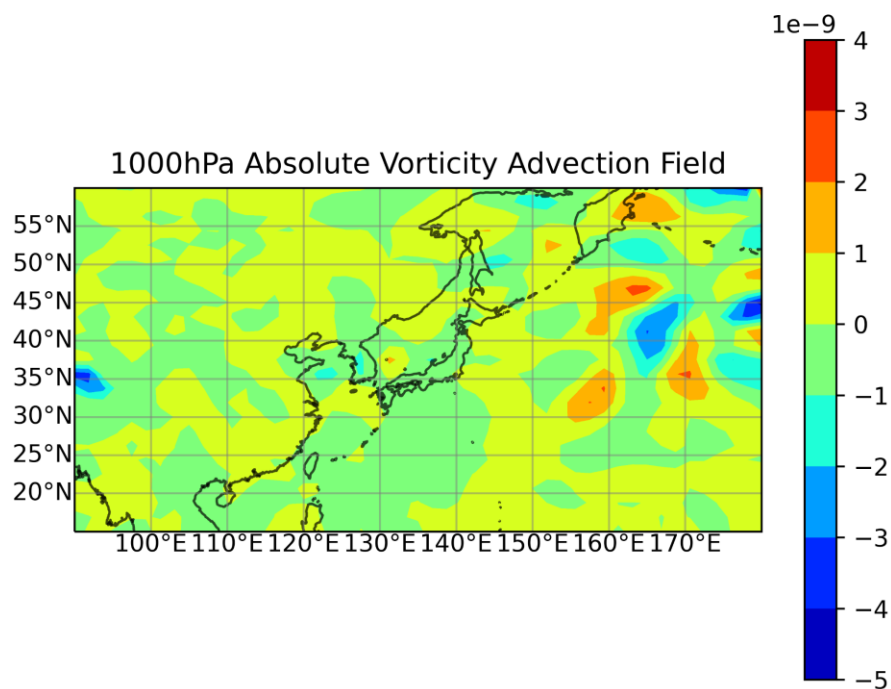




絕對渦度平流* 5 層 (高到低)







(5) 計算與繪圖程式碼 + 註解

Import requestment 如下

```
import os
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as mticker
import math
import sys
print(sys.prefix) # show what virtual env I am in
import cartopy.crs as ccrs
from cartopy.mpl.gridliner import LONGITUDE_FORMATTER, LATITUDE_FORMATTER
```

```
# read binary data, analyze to 49x*25y*5(1000 850 700 500 300)*4(H U V
T) = 24500 np.array, return wanted plane data
def read_bindata_return_wanted(hw1_root_path, filename, pressure, parameter):
```

此方法為讀 bin 檔資料的 function，輸出想要的平面資料

```
# Input x y and output the corresponding latitude and longitude coordinates
def xy_to_lonlat(x, y):
# Input y and output the corresponding latitude coordinates
def y_to_lat(y):
```

此兩方法轉換 x, y 值為經緯度

```
# Input pre, post, and d and output interpolation differential.
def median_interpolation(front, behind, d):
# Input pre, here, and d and output the pre-interpolated differential.
def front_interpolation(front, here, d):
# Input here value and post value and output post-interpolation differential.
def behind_interpolation(here, behind, d):
```

分別為前插、中差與後差法的程式實現

```
# Enter flat data, draw on a map and save it.
def plot_in_map(plane_data, title, pressure, var_name):
```

在地圖上畫出每張不同的圖，負責繪圖控制

```
# Input u, v, t field and pressure layer to create a horizontal temperature advection field and plot.
def plot_horizontal_temperature_advection(u, v, t, pressure):
# Input u, v field and pressure layer to create a divergence field and plot.
def plot_divergence(u, v, pressure):
# Input u, v field and pressure layer to create a relative vorticity field and plot.
def plot_relative_vorticity(u, v, pressure):
# Input u, v field and pressure layer to create a relative vorticity field and plot.
def plot_absolute_vorticity_advection(u, v, pressure):
```

處理每種不同所要的參數的實際計算，並引用 `plot_in_map` 繪圖

```
if __name__ == "__main__":
```

主程式，做許多常參數的初始化，以及 `names = locals()` 即讀取 python 程式中的變數名稱，並且用迴圈幫我快速創建變數，最後使用 `plot` 字首的 function 完成繪圖並儲存。

(若有 debugger 如 `vscode` 裡面內建的會在之後引用這些變數時顯示危險，不過作業系統如果一樣就沒事啦放心)