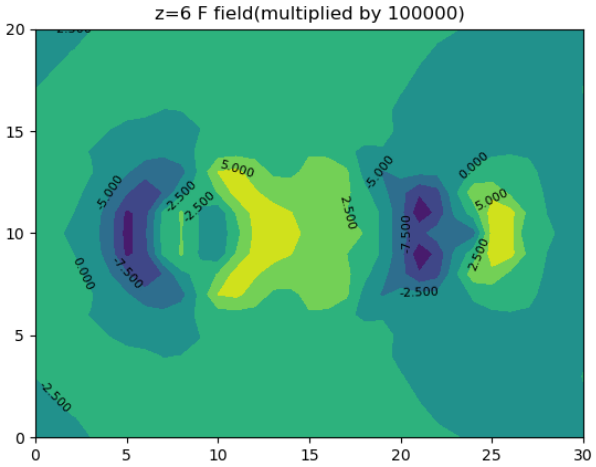
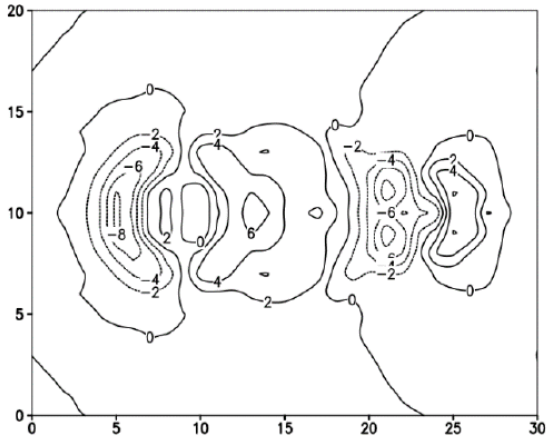
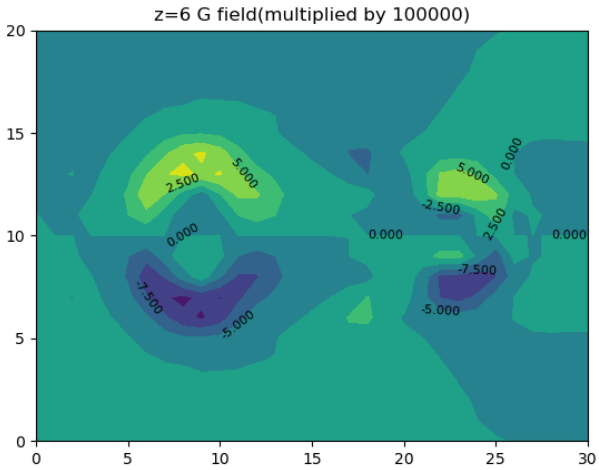
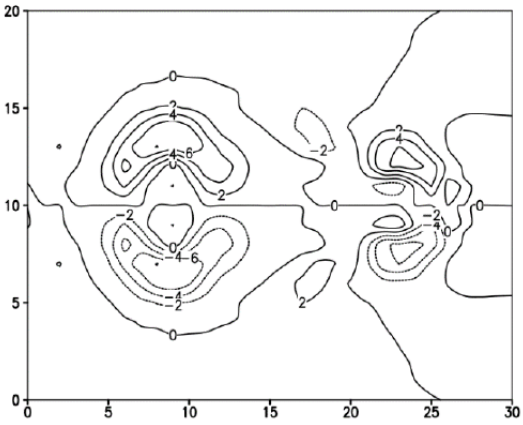
高等應用數學　　HW6　　黃展皇 110621013

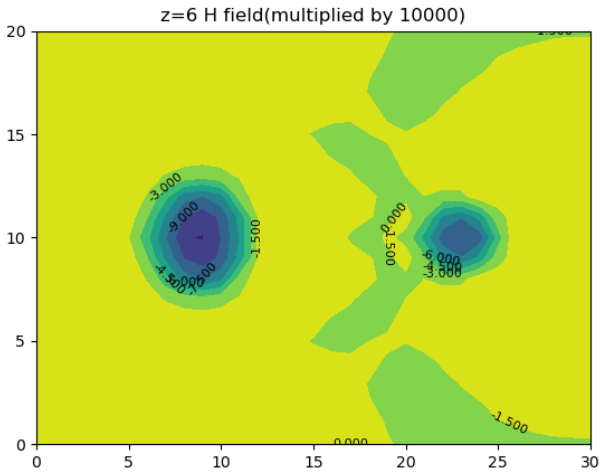
1. Plot the fields of F、G and H(z=6)：

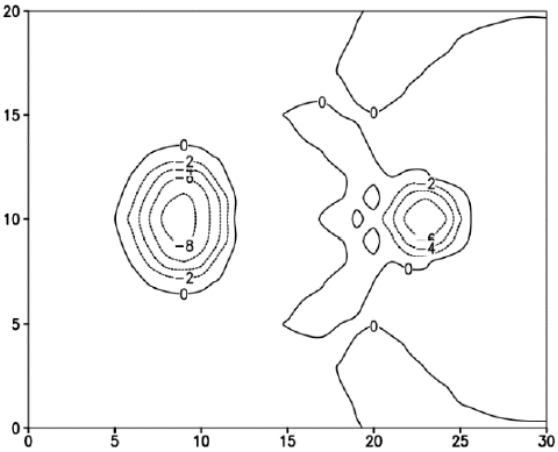












可以看到繪製出來的圖對比作業附圖來說都是對的，而且更漂亮😋

2. retrieve pressure and temperature perturbation fields (π',$\theta c'$) at each layer.

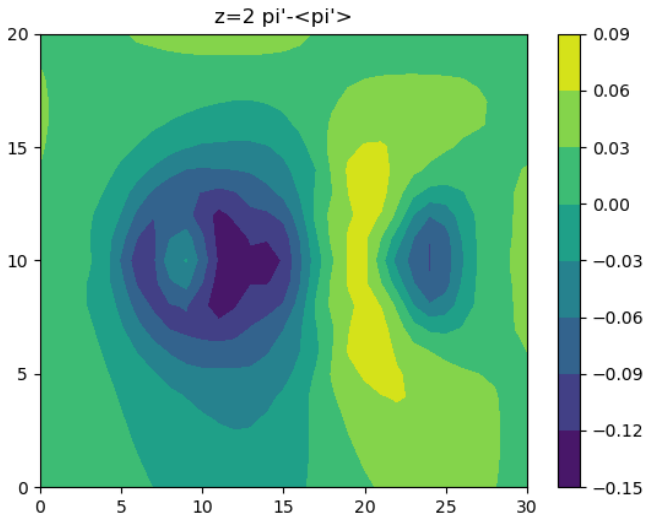
作法：π'反演採用 SOR，eps=1.5，結束條件是每一點相對誤差均小於 0.001，輸入該層 F、G 即可得到該層π'。pi 矩陣代表π'一開始都是 0.0，定義 Neumann 邊界條件使用中插法、第一內圈 F、G 以及第二內圈 pi 得到外圈 pi，並進行內圈 SOR：

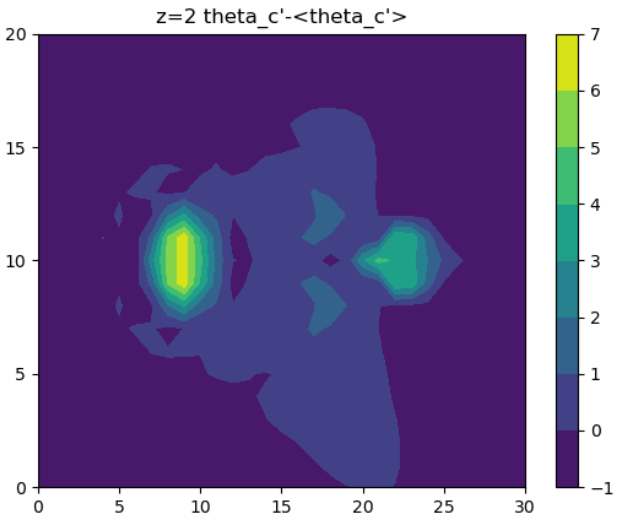
sor_result = -0.25*dx*dy*(median_interpolation(F[y, x-1], F[y, x+1], dx) + median_interpolation(G[y-1, x], G[y+1, x], dy)) + (pi[y+1, x]+pi[y-1, x]+pi[y, x+1]+pi[y, x-1])/4

pi[y, x] += eps * (sor_result - pi[y, x])，重複以上步驟迭代即可得到π'場。

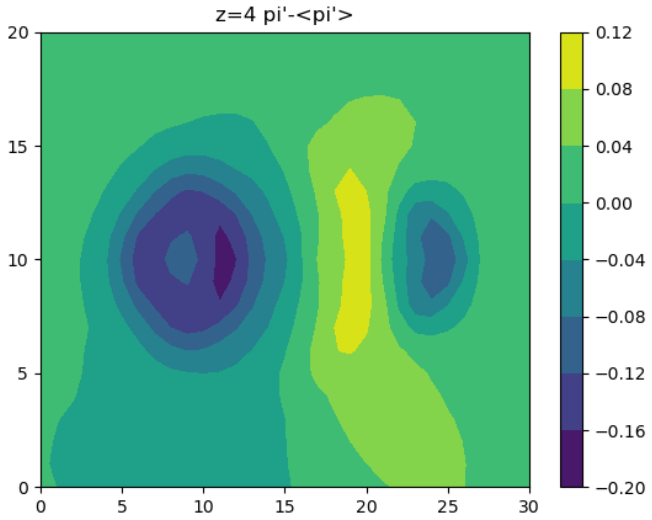
$\theta c'$反演則使用該層上下的π'場以及該層 H 場還有 theta_0_avg, theta_v0_avg，套用 theta_c = (dpidz-H)*theta_0_avg*theta_v0_avg/g 計算即可知道$\theta c'$場。

(z=2) π'-<π'>場　　$\theta c'$-<$\theta c'$>場

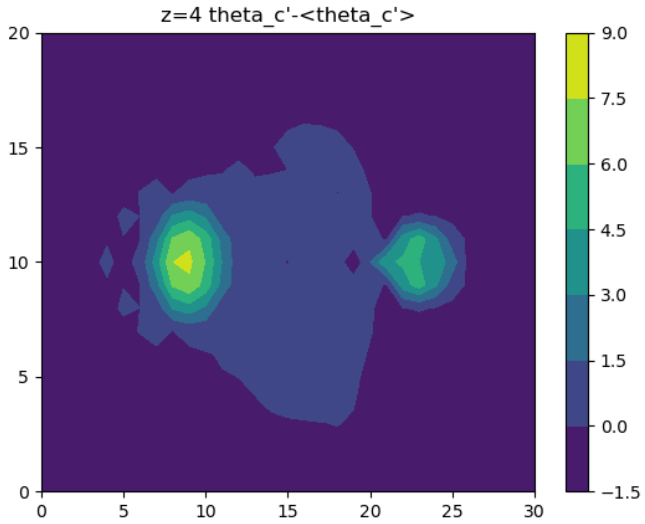




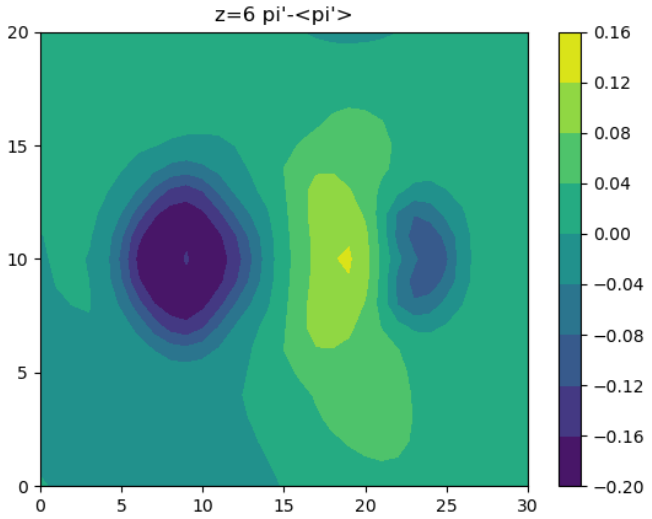
(z=4) π'-<π'>場



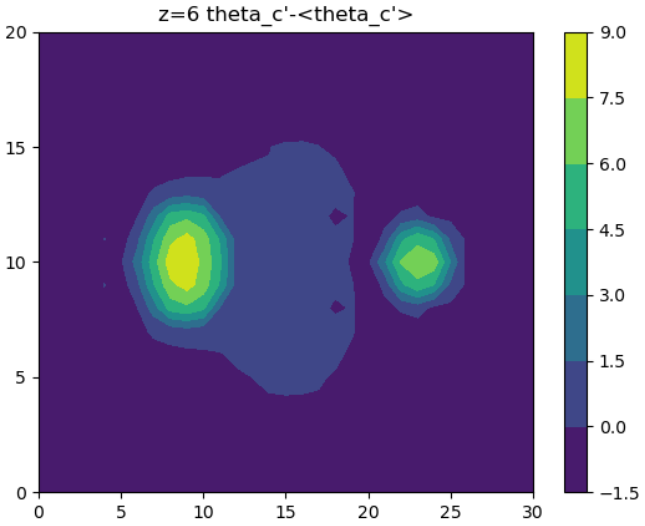
$\theta c'$-<$\theta c'$>場



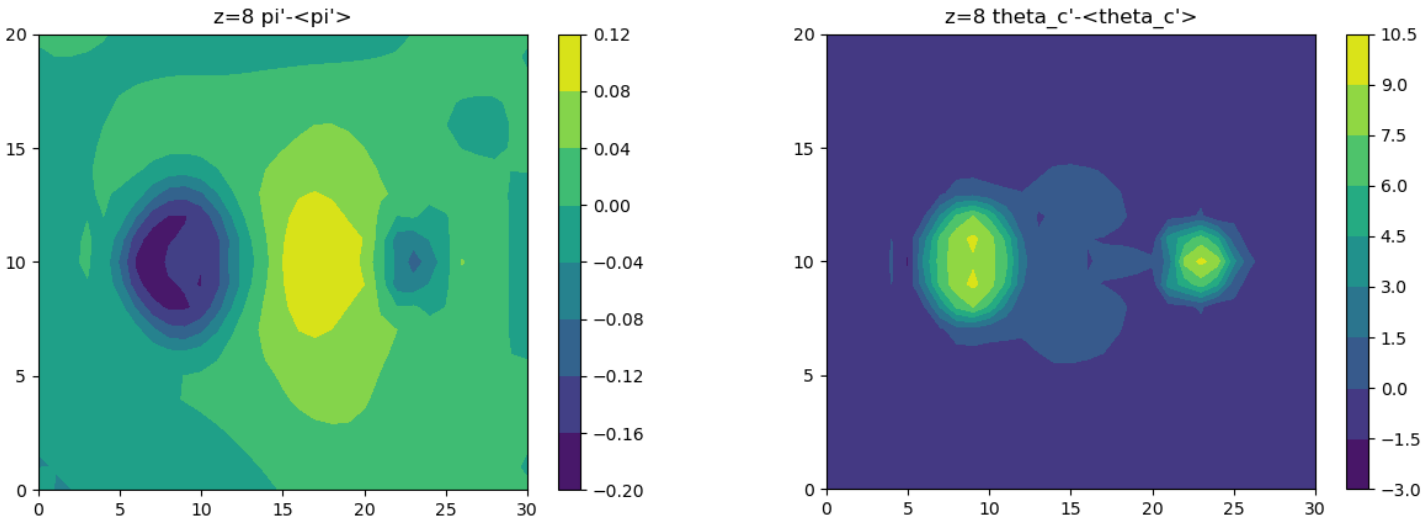
(z=6) π'-<π'>場



$\theta c'$-<$\theta c'$>場



(z=8) π'-<π'>場　　$\theta c'$-<$\theta c'$>場



可以看到反演出來的π'-<π'>場在中心有相對高值，並在左右各有相對低值，

又以左側為更低；而$\theta c'$-<$\theta c'$>場則是呈現左右兩個明顯的熱胞柱。

原始碼：(環境：win10、conda4.10.3、python3.8.8、numpy1.19.5)

```
import os
import numpy as np
import matplotlib.pyplot as plt
x_n, y_n, z_n = 31, 21, 11
dx, dy, dz = 1000, 1000, 250

# read hw6 data, 31*21*11*5(x,y,z,var), z:1~11
def read_hw6_data(z, parameter):
    parameter_list = ['ff', 'gg', 'hh', 'theta0', 'thetav0',
'ans_thetac']
    data = np.fromfile('homework6.bin', dtype='<f4', count=-1, sep='')
    init_lndex = x_n*y_n*z_n*parameter_list.index(parameter) +
x_n*y_n*(z-1)
    return np.reshape(data[init_lndex:init_lndex+x_n*y_n], (y_n, x_n))
# plot parameter
```

```
def plot_parameter(plane_data, title, scale):
    plane_data_new = np.array(plane_data)
    assert id(plane_data) != id(plane_data_new)
    plane_data_new *= scale
    C = plt.contourf(plane_data_new) #, levels=[i for i in range(-10,
10, 2)]
    plt.clabel(C, fontsize=8, colors='black', inline=False)
    plt.title(title)
    plt.xticks(range(0, 31, 5))
    plt.yticks(range(0, 21, 5))
    plt.savefig('6-'+title)
    #plt.show()
    plt.close()
# Input pre, post, and d and output interpolation differential.
def median_interpolation(front, behind, d):
    return (behind-front)/(2*d)

def cal_pi(F, G, z_name):
    # if pi has been cal
    filepath = os.path.join('.', '6-'+z_name+'_pi.npy')
    if os.path.isfile(filepath):
        pi = np.load(os.path.join('.', '6-'+z_name+'_pi.npy'))
    else:
        # pi B.C. init
        pi = np.zeros((y_n, x_n), dtype=np.float64)
        for y in range(1, y_n-1):
            pi[y, 0] = -2*dx*F[y, 1] + pi[y, 2]
            pi[y, x_n-1] = 2*dx*F[y, x_n-2] - pi[y, x_n-3]
        for x in range(1, x_n-1):
            pi[0, x] = -2*dy*F[1, x] + pi[2, x]
            pi[y_n-1, x] = 2*dy*F[y_n-2, x] - pi[y_n-3, x]

        # SOR + B.C. renew
        count = 0
        eps = np.array([1.5], dtype=np.float64)[0]
        threshold = np.array([10**-3], dtype=np.float64)[0]
        while True:
            old_pi = np.array(pi) # call by values
```

```
            count += 1

            for y in range(1, y_n-1):
                for x in range(1, x_n-1):
                    #sor_result = 2*dx*dy*(median_interpolation(F[y, x-
1], F[y, x+1], dx) + median_interpolation(G[y-1, x], G[y+1, x], dy)) +
(pi[y+1, x]+pi[y-1, x]+pi[y, x+1]+pi[y, x-1])/4
                    sor_result = -0.25*dx*dy*(median_interpolation(F[y,
x-1], F[y, x+1], dx) + median_interpolation(G[y-1, x], G[y+1, x], dy))
+ (pi[y+1, x]+pi[y-1, x]+pi[y, x+1]+pi[y, x-1])/4
                    pi[y, x] += eps * (sor_result - pi[y, x])

            skip_flag = False
            max_rela_e = np.array([0.0], dtype=np.float64)[0]
            for y in range(1, y_n-1):
                for x in range(1, x_n-1):
                    if old_pi[y, x] == np.array([0.0],
dtype=np.float64)[0]:
                        max_rela_e = 10*threshold # abs fail
                        skip_flag = True
                    if skip_flag == True:
                        break
                    max_rela_e = max(max_rela_e, abs((pi[y, x]-old_pi[y,
x])/old_pi[y, x]))
                if skip_flag == True:
                    break
            ''' MSE
            skip_flag = False
            mse = np.array([0.0], dtype=np.float64)[0]
            for i in range(2, n-2):
                for j in range(2, n-2):
                    if old_lamb[i][j] == np.array([0.0],
dtype=np.float64)[0]:
                        mse = 10*mse_threshold # abs fail
                        skip_flag = True
                    if skip_flag == True:
                        break
                    mse += (lamb[i][j]-old_lamb[i][j])**2
```

```
                if skip_flag == True:
                    break
            '''
            # B.C. renew
            for y in range(1, y_n-1):
                pi[y, 0] = -2*dx*F[y, 1] + pi[y, 2]
                pi[y, x_n-1] = 2*dx*F[y, x_n-2] - pi[y, x_n-3]
            for x in range(1, x_n-1):
                pi[0, x] = -2*dy*G[1, x] + pi[2, x]
                pi[y_n-1, x] = 2*dy*G[y_n-2, x] - pi[y_n-3, x]


            if skip_flag or count%100==0:
                print(count, skip_flag, max_rela_e)
            if max_rela_e < threshold:
                break
            #if count == 4000:
            #    break
            #if count % 1000 == 0:
            #    plt.contourf(pi)
            #    plt.colorbar()
            #    plt.show()
        # -c
        pi_avg = np.mean(pi[0:y_n, 0:x_n])
        print('pi_avg:', pi_avg)
        pi -= pi_avg
        np.save('6-'+z_name+'_pi.npy', pi)
    plt.contourf(pi)
    plt.colorbar()
    plt.title(z_name+" pi'-<pi'>")
    plt.xticks(range(0, 31, 5))
    plt.yticks(range(0, 21, 5))
    plt.savefig('6-'+z_name+'_pi')
    #plt.show()
    plt.close()

def cal_theta_c(H, theta_0, theta_v0, z_name):
    # if pi has been cal
    filepath = os.path.join('.', '6-'+z_name+'_theta_c.npy')
```

```
    if os.path.isfile(filepath):
        theta_c = np.load(os.path.join('.', '6-'+z_name+'_theta_c.npy'))
    else:
        assert z_name in ['z='+str(i) for i in range(2, 11)]
        middle_num = int(z_name.split('=')[-1])
        under = np.load('6-z='+str(middle_num-1)+'_pi.npy')
        up = np.load('6-z='+str(middle_num+1)+'_pi.npy')
        dpidz = median_interpolation(under, up, dz)

        theta_0_avg, theta_v0_avg = np.mean(theta_0), np.mean(theta_v0)
        g = 9.8
        H -= np.mean(H)
        theta_c = (dpidz-H)*theta_0_avg*theta_v0_avg/g
        np.save('6-'+z_name+'_theta_c.npy', theta_c)

    plt.contourf(theta_c)
    plt.colorbar()
    plt.title(z_name+" theta_c'-<theta_c'>")
    plt.xticks(range(0, 31, 5))
    plt.yticks(range(0, 21, 5))
    plt.savefig('6-'+z_name+'_theta_c')
    #plt.show()
    plt.close()


if __name__ == '__main__':
    for z in range(2, 10, 2):
        f = read_hw6_data(z=z, parameter='ff')
        g = read_hw6_data(z=z, parameter='gg')
        h = read_hw6_data(z=z, parameter='hh')
        theta_0 = read_hw6_data(z=z, parameter='theta0')
        theta_v0 = read_hw6_data(z=z, parameter='thetav0')
        ans_thetac = read_hw6_data(z=z, parameter='ans_thetac')

        plot_parameter(f, 'z={} F field(multiplied by
{})'.format(str(z), str(10**5)), scale=10**5)
        plot_parameter(g, 'z={} G field(multiplied by
{})'.format(str(z), str(10**5)), scale=10**5)
```

```
        plot_parameter(h, 'z={} H field(multiplied by
{})'.format(str(z), str(10**4)), scale=10**4)
        plot_parameter(theta_0, 'z={} theta0 field'.format(str(z)),
scale=1)
        plot_parameter(theta_v0, 'z={} thetav0 field'.format(str(z)),
scale=1)
        plot_parameter(ans_thetac, 'z={} ans_thetac
field'.format(str(z)), scale=1)

        cal_pi(f, g, z_name='z='+str(z))

        f_under = read_hw6_data(z=z-1, parameter='ff')
        g_under = read_hw6_data(z=z-1, parameter='gg')
        cal_pi(f_under, g_under, z_name='z='+str(z-1))
        f_up = read_hw6_data(z=z+1, parameter='ff')
        g_up = read_hw6_data(z=z+1, parameter='gg')
        cal_pi(f_up, g_up, z_name='z='+str(z+1))
        cal_theta_c(h, theta_0, theta_v0, z_name='z='+str(z))
```