

数值天气预报课程作业

吴浩杰 20221170212

1 实习八

空间平滑

1.1 实习目的

通过编程实现空间平滑，使学生掌握边界和区域内网格点空间平滑的基本原理及计算方法。

1.2 实习内容

(1) 采用兰勃特投影方式，对欧洲中心 ERA5 再分析资料 1979 年 1 月 10 日 00 时的 500hPa 重力位势高度场进行九点空间平滑。

Listing 1: 空间平滑

```
1  import os
2  import numpy as np
3  import xarray as xr
4  import matplotlib.pyplot as plt
5  import cartopy.crs as ccrs
6  from cartopy.util import add_cyclic_point
7
8  def load_500hpa_height(nc_file):
9      """
10     读取 NetCDF，只加载 500 hPa 第一个时次的位势（z），
11     并转换为高度场（m），返回 height(lat, lon)、lon、lat。
12     """
13     with xr.open_dataset(nc_file, engine='netcdf4',
14                          decode_times=False) as ds:
15         # 先选层，再按位置选第一个时次，避免 KeyError
16         z0 = ds['z'].sel(pressure_level=500).isel(valid_time=0)
17         height = (z0 / 9.80665).values # 转换为 m
```

```

17
18     # 经度转换到 [-180,180) 并排序
19     lon = ((ds.longitude.values + 180) % 360) - 180
20     idx = np.argsort(lon)
21     lon = lon[idx]
22     height = height[:, idx]
23
24     lat = ds.latitude.values
25
26     return height, lon, lat
27
28 def smooth_9point(data):
29     """
30     高效九点平滑（分离卷积）：
31     1) 纬度方向边界复制进行 [1,2,1]/4 卷积
32     2) 经度方向周期循环进行相同卷积
33     """
34     # 纬度方向平滑
35     pad_lat = np.pad(data, ((1,1),(0,0)), mode='edge')
36     lat_s = (pad_lat[: -2, :] + 2*pad_lat[1: -1, :] + pad_lat
37              [2:, :]) * 0.25
38
39     # 经度方向平滑
40     pad_lon = np.concatenate((lat_s[:, -1:], lat_s, lat_s
41                              [:, :1]), axis=1)
42     smooth = (pad_lon[:, : -2] + 2*pad_lon[:, 1: -1] + pad_lon
43             [:, 2:]) * 0.25
44
45     return smooth
46
47 def plot_contour(ax, height, lon, lat, subtitle):
48     """
49     在 Lambert Conformal 投影上，只画黑色等值线，
50     样式参考示例图，无填色和 colorbar。
51     """
52     h_cyc, lon_cyc = add_cyclic_point(height, coord=lon)
53
54     ax.set_extent([60, 180, 20, 80], crs=ccrs.PlateCarree())
55     ax.coastlines('50m', linewidth=1)
56
57     # 网格线
58     gl = ax.gridlines(crs=ccrs.PlateCarree(), draw_labels=
59                       True,

```

```

56         linestyle='--', linewidth=0.5, color='
           gray')
57     gl.top_labels = False
58     gl.right_labels = False
59     gl.xlocator = plt.FixedLocator(np.arange(60, 181, 30))
60     gl.ylocator = plt.FixedLocator(np.arange(20, 81, 10))
61     gl.xlabel_style = gl.ylabel_style = {'size': 8}
62
63     # 等值线
64     levels = np.arange(4800, 6000, 120)
65     cs = ax.contour(lon_cyc, lat, h_cyc, levels=levels,
66                   colors='black', linewidths=1.2,
67                   transform=ccrs.PlateCarree())
68     ax.clabel(cs, fmt='%d', inline=True, fontsize=8)
69
70     # 左对齐小标题
71     ax.set_title(subtitle, fontsize=10, loc='left')
72
73 if __name__ == '__main__':
74     # 文件路径
75     nc_path = r'D:\zhuomian\shuzhi\shiyang8\500hpa.nc'
76     assert os.path.exists(nc_path), f"文件不存在: {nc_path}"
77
78     # 中文字体设置
79     plt.rcParams['font.sans-serif'] = ['SimHei']
80     plt.rcParams['axes.unicode_minus'] = False
81
82     # 1. 加载并预处理数据
83     height, lon, lat = load_500hpa_height(nc_path)
84
85     # 2. 快速九点平滑
86     height_s = smooth_9point(height)
87
88     # 3. 绘图
89     fig = plt.figure(figsize=(10, 4))
90     proj = ccrs.LambertConformal(
91         central_longitude=120,
92         central_latitude=50,
93         standard_parallels=(30, 60)
94     )
95     ax1 = fig.add_subplot(1, 2, 1, projection=proj)
96     plot_contour(ax1, height, lon, lat, '(a) 原始数据')
97

```

```

98     ax2 = fig.add_subplot(1, 2, 2, projection=proj)
99     plot_contour(ax2, height_s, lon, lat, '(b) 九点平滑后的
      结果')
100
101     # 总标题
102     fig.suptitle('1979年1月10日 500 hPa 重力位势高度场',
      fontsize=12)
103     plt.tight_layout(rect=[0, 0.03, 1, 0.95])
104     plt.show()

```

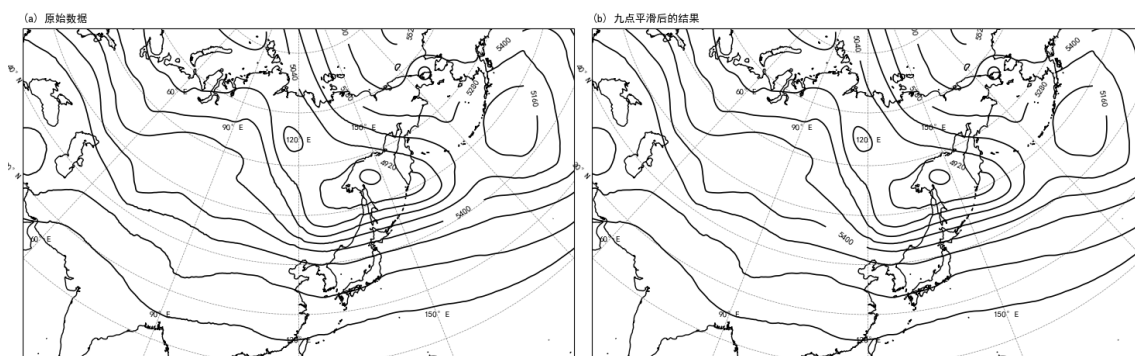


Figure 1: 平滑前后对比

可见九点平滑的等值线更为光滑。

2 实验八习题一

2.1 习题一

绘图比较五点平滑和九点平滑的响应函数 R 随波长的变化，分析它们对各波长振幅衰减的影响。

Listing 2: 实验八习题一

```

1  # 习题一.py
2  import numpy as np
3  import matplotlib.pyplot as plt
4
5  # ----- 中文字体配置 -----
6  plt.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 微软雅黑
7  plt.rcParams['axes.unicode_minus'] = False           # 负号正常显示
8
9  # ----- 参数设置 -----
10 S = 0.5 # 平滑系数

```

```

11 dx = 1      # 网格间距
12 dy = dx     # 有时 dy 可不同，这里两者相同
13 L = np.arange(1, 37) # 波长范围 1,2,...,36
14
15 # ----- 计算响应函数 -----
16 # 先把角度转换为弧度：180./L * dx 是度
17 argx = np.deg2rad(180.0 / L * dx)
18 argy = np.deg2rad(180.0 / L * dy)
19
20 # 五点平滑：R5 = 1 - S*(sin^2(...)_x + sin^2(...)_y)
21 R5 = 1.0 - S * (np.sin(argx)**2 + np.sin(argy)**2)
22 # 九点平滑：R9 = (1 - 2 S sin^2(..._x)) * (1 - 2 S sin^2(..._y))
23 R9 = (1.0 - 2.0 * S * np.sin(argx)**2) * (1.0 - 2.0 * S * np
    .sin(argy)**2)
24
25 # ----- 准备绘图数据：波长从 2 到 36 -----
26 x = L[1:]      # 2,3,...,36
27 R5_p = R5[1:]   # 对应的响应值
28 R9_p = R9[1:]
29 diff_p = R5_p - R9_p # R5 - R9
30
31 # ----- 开始绘图 -----
32 plt.figure(figsize=(8, 5))
33
34 # 五点平滑曲线
35 plt.plot(x, R5_p, 'k-', linewidth=1.5, label=r'$R_{5}(1/2,L)$')
36 # 九点平滑曲线
37 plt.plot(x, R9_p, 'k--', linewidth=1.5, label=r'$R_{9}(-1/2,L)$')
38 # 差值曲线
39 plt.plot(x, diff_p, 'k-.', linewidth=1.5, label=r'$R_{5}(1/2,L) - R_{9}(-1/2,L)$')
40
41 # 坐标范围与刻度
42 plt.xlim(0, 36)
43 plt.ylim(-0.5, 1)
44 plt.xticks(
45     [0, 2, 4, 8, 12, 16, 20, 24, 28, 32, 36],
46     ['0', '2', '4', '8', '12', '16', '20', '24', '28', '32', '36'])
47 )
48 plt.yticks([-0.5, 0, 0.5, 1], ['-0.5', '0', '0.5', '1'])

```

```

49
50 # 标签与标题
51 plt.title('五点平滑、九点平滑的响应函数随波长的变化',
           fontsize=14)
52 plt.xlabel(r'$L/\Delta x$', fontsize=12)
53 plt.ylabel('响应函数 R', fontsize=12)
54
55 # 栅格线（点状）
56 plt.grid(linestyle=':', linewidth=0.5)
57
58 # 图例
59 plt.legend(loc='upper right', fontsize=12)
60
61 # 显示
62 plt.tight_layout()
63 plt.show()

```

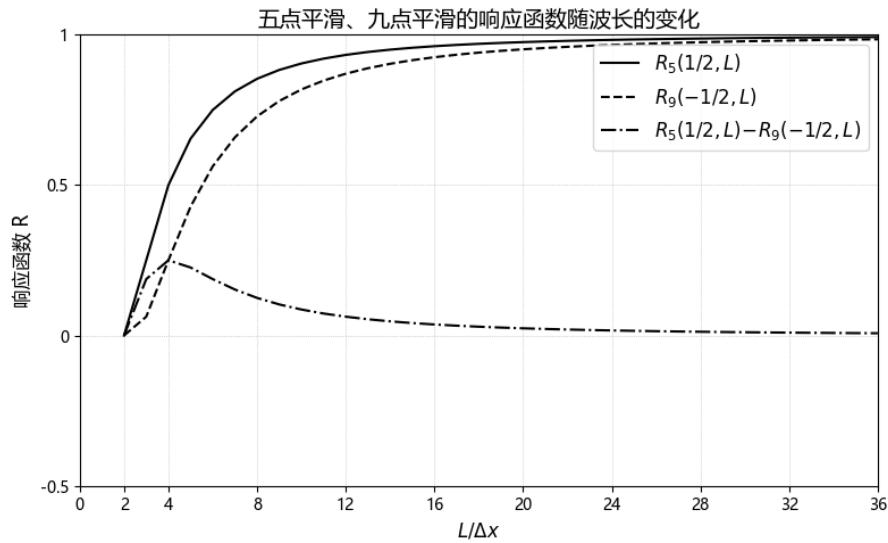


Figure 2: 实验八习题

当 $L = 2\Delta x$ 时, $R_9(S, k, l) - R_5(S, k, l) = 0$, 五点平滑和九点平滑的平滑效应相同。当 $L > 2\Delta x$ 时, $R_9(S, k, l) - R_5(S, k, l) < 0$, 九点平滑的平滑效应大于五点平滑, 并且随波长增大, 即对于长波的衰减, 九点平滑要比五点平滑强得多。总的来看, 五点平滑和九点平滑都是随着波长增大, 响应函数越接近 1。