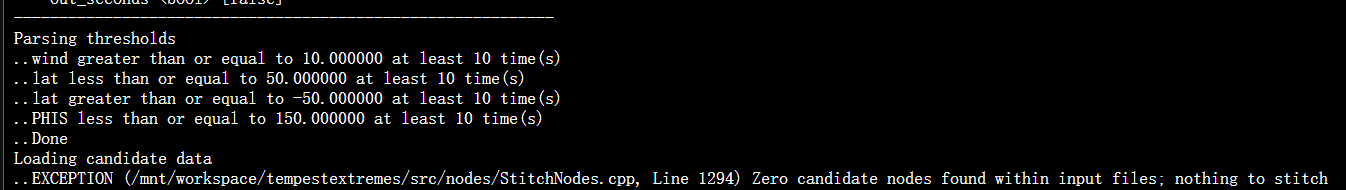
1. **TE\_geos.sh** file

For running the TE\_geos.sh file, the result is:

Loading candidate data

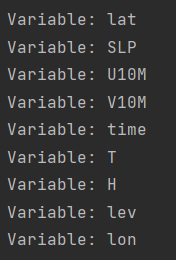
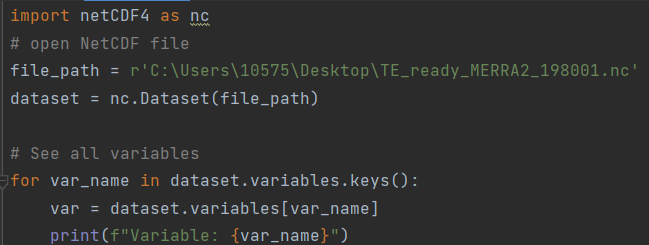
..EXCEPTION (/mnt/workspace/tempestextremes/src/nodes/StitchNodes.cpp, Line 1294) Zero candidate nodes found within input files; nothing to stitch

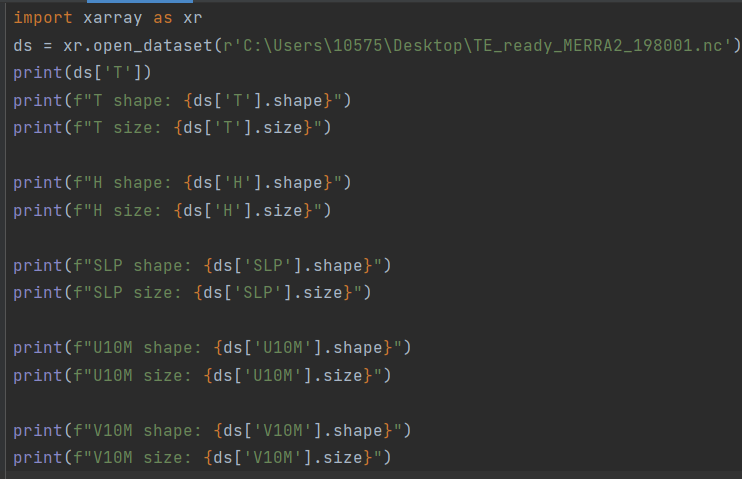


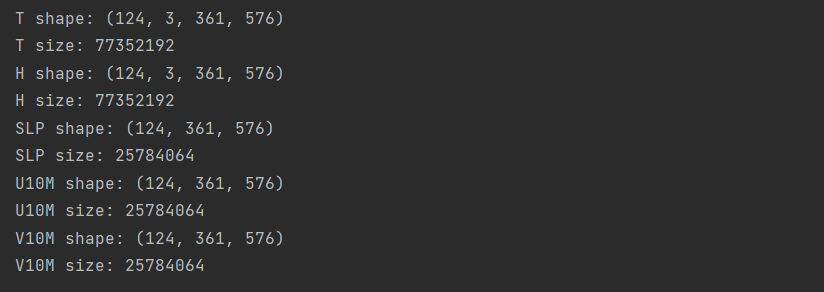
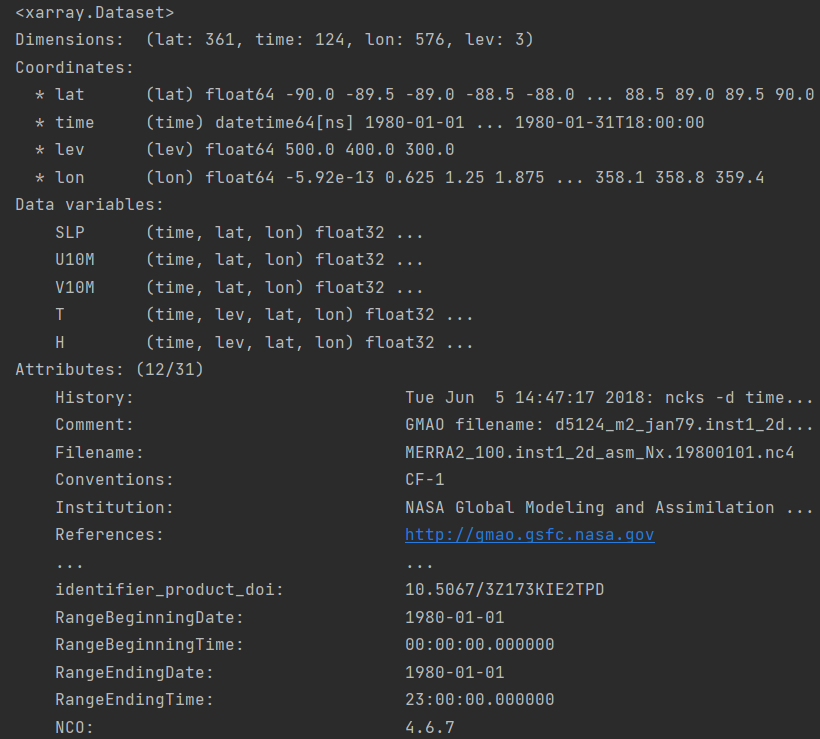
This means that the input data does not meet the detection criteria, resulting in no cyclone nodes being found that satisfy the requirement.

1. **TE\_ready\_MERRA2\_198001.nc** (NetCDF) file

①See all variables



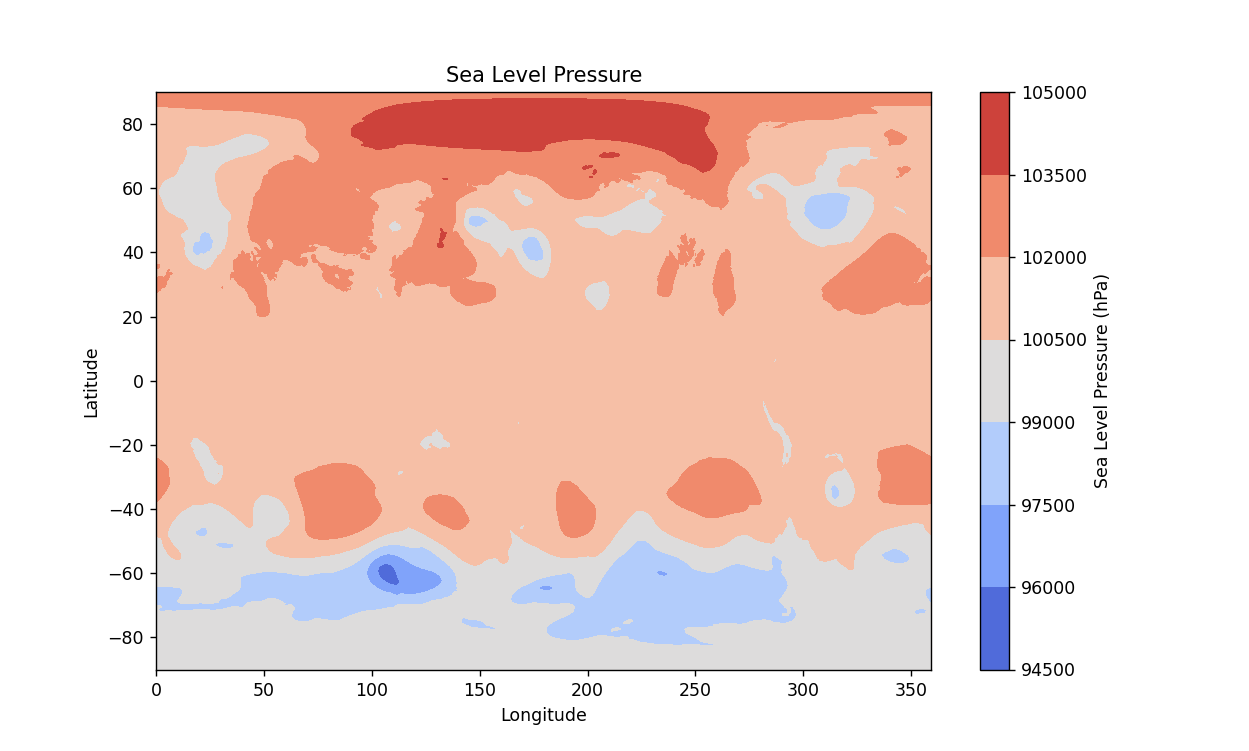




Problem: Coordinates are used to define the position of data points in multidimensional space, while data variables are what are actually stored and analyzed.

②Different variables in certain time

1. **SLP, lat, lon**



slp = dataset.variables['SLP'][:]

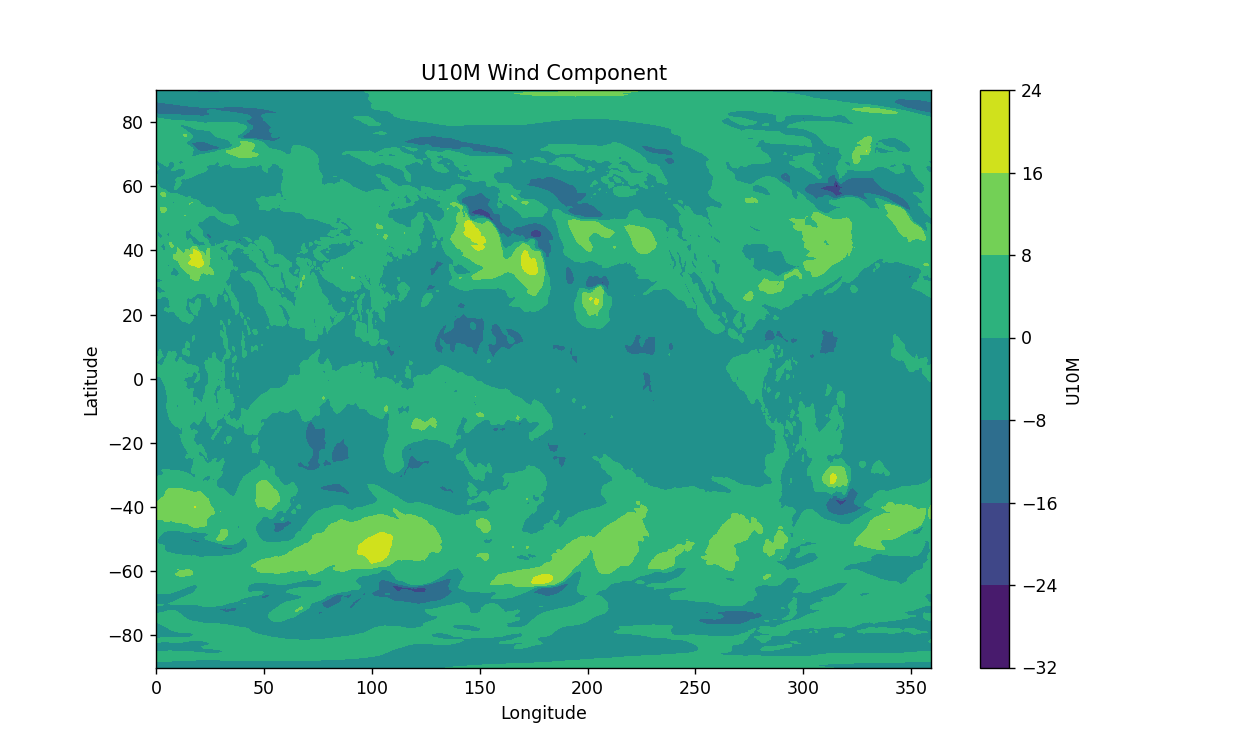
lat = dataset.variables['lat'][:]

lon = dataset.variables['lon'][:]

plt.figure(figsize=(10, 6))

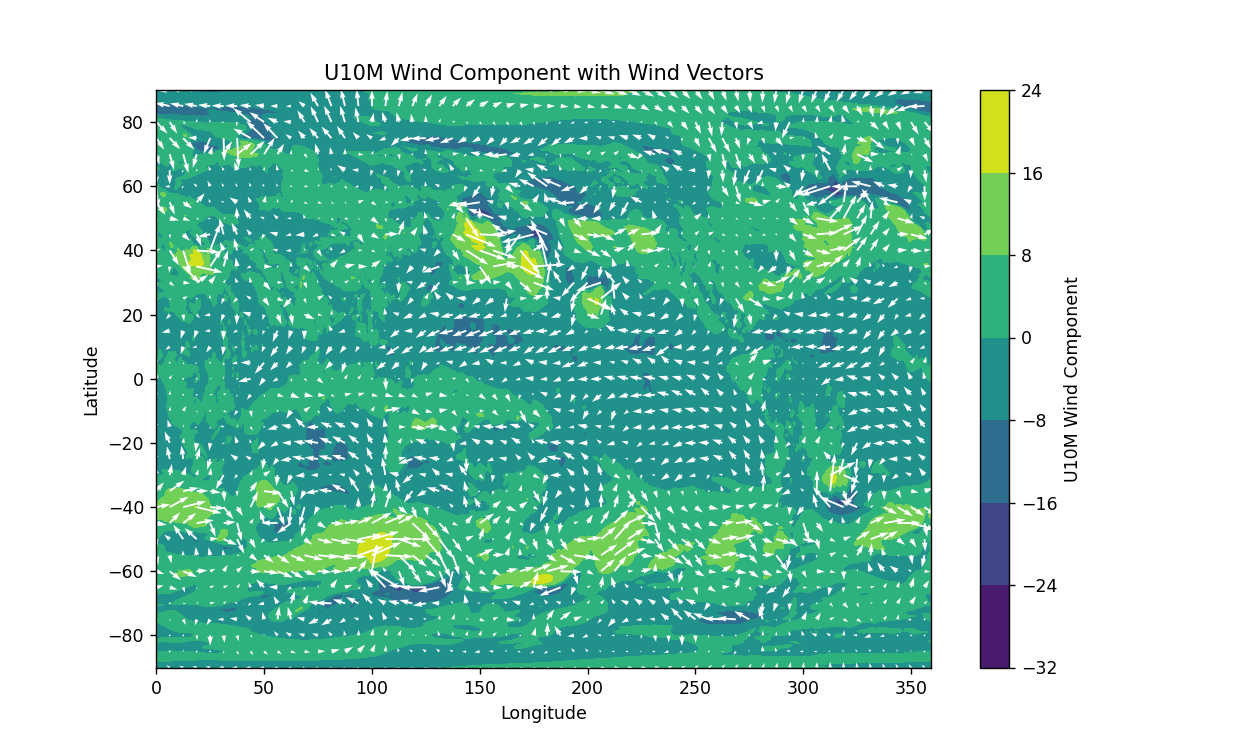
plt.contourf(lon, lat, slp[0, :, :], cmap='coolwarm')

1. **U10M, lon, lat**



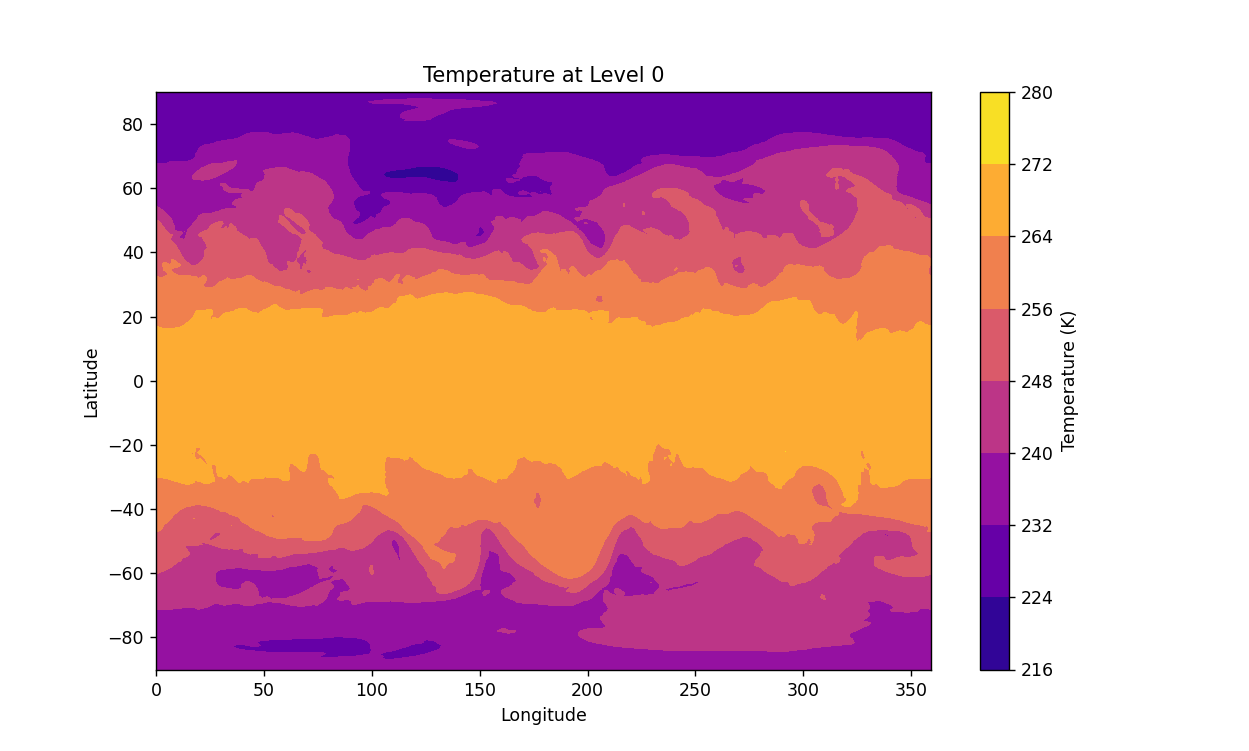
lat = dataset.variables['lat'][:]  
lon = dataset.variables['lon'][:]  
u10m = dataset.variables['U10M'][:]  
plt.figure(figsize=(10, 6))  
plt.contourf(lon, lat, u10m[0, :, :], cmap='viridis')

1. **U10M, V10M, lat, lon**



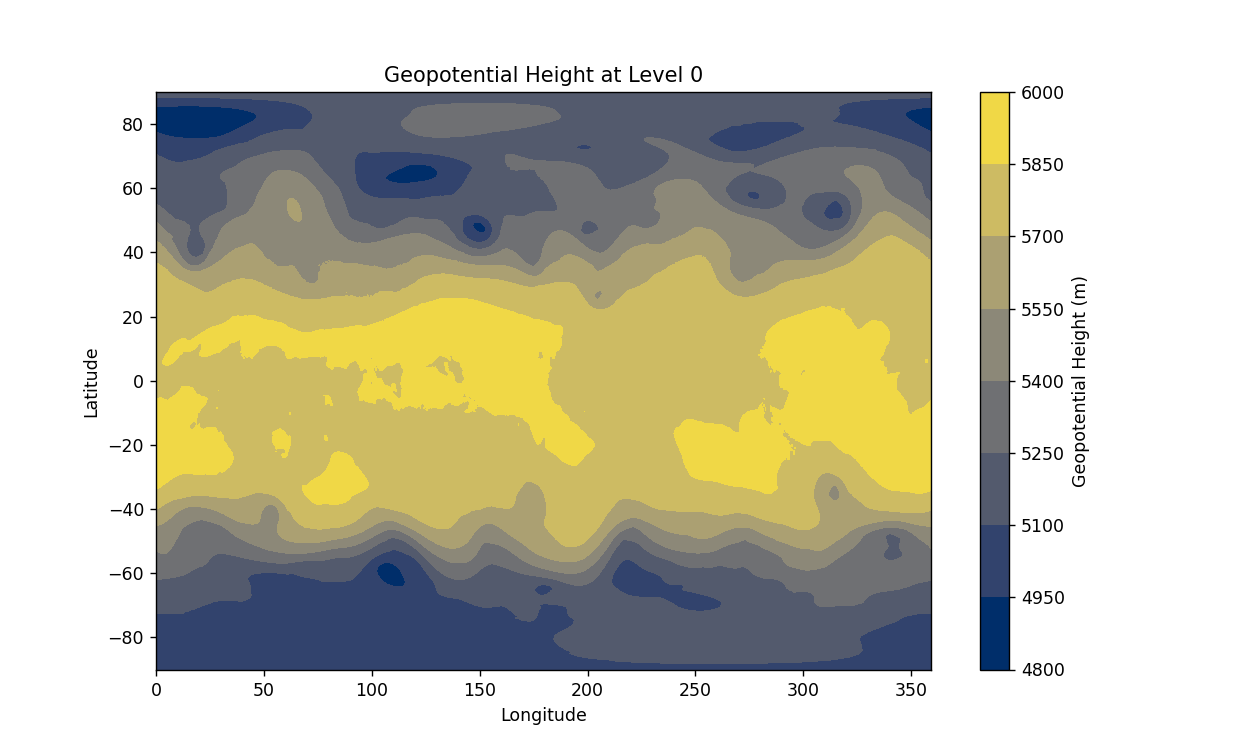
plt.figure(figsize=(10, 6))  
plt.contourf(lon, lat, u10m[0, :, :], cmap='viridis')  
plt.colorbar(label='U10M Wind Component')  
plt.quiver(lon[::10], lat[::10], u10m[0, ::10, ::10], v10m[0, ::10, ::10], color='white')

1. **T, lon, lat**



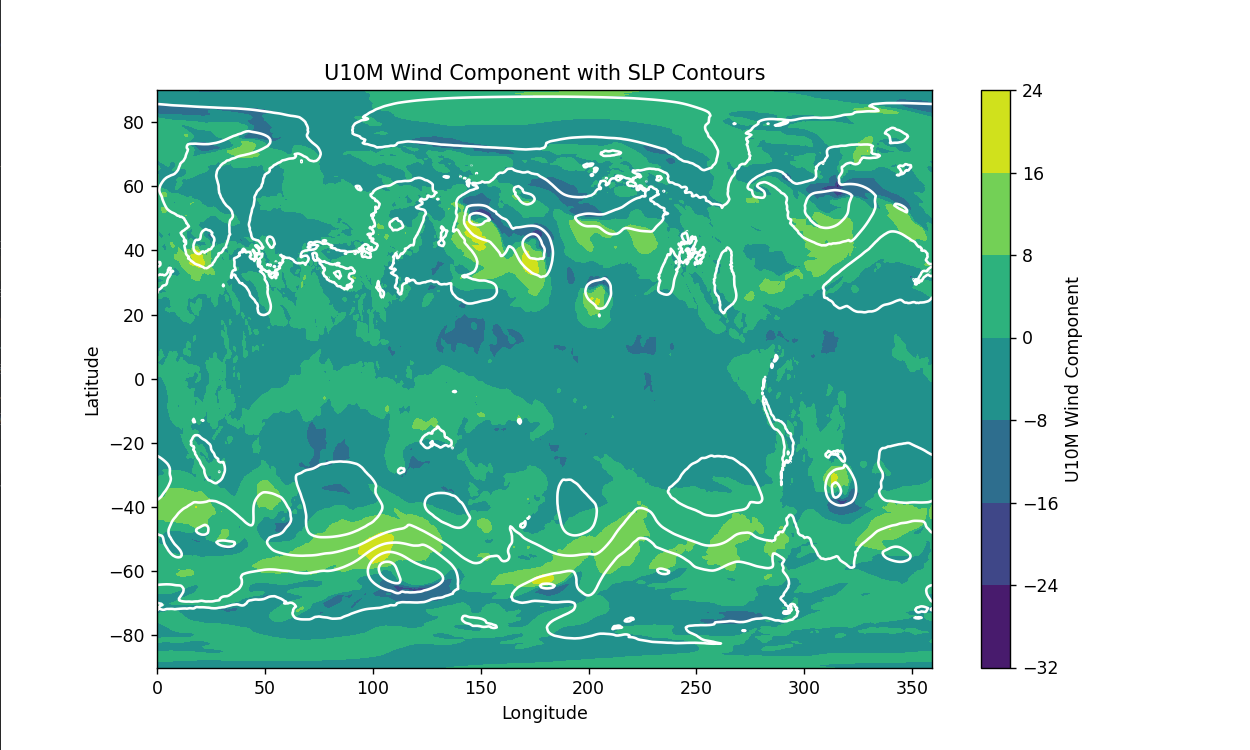
t = dataset.variables['T'][:]  
plt.figure(figsize=(10, 6))  
plt.contourf(lon, lat, t[0, 0, :, :] , cmap='plasma')

1. **H, lon, lat**



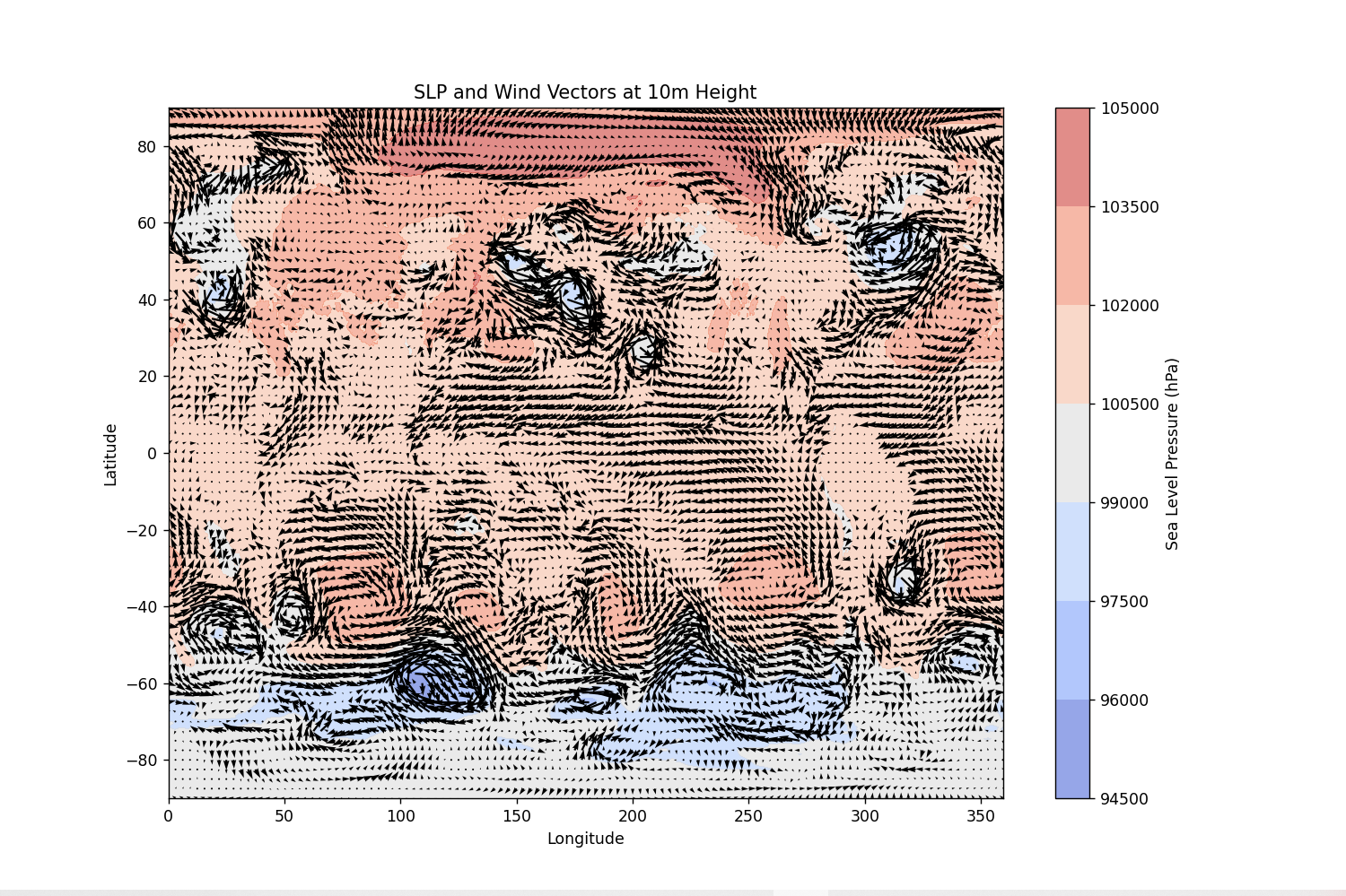
h = dataset.variables['H'][:]  
plt.figure(figsize=(10, 6))  
plt.contourf(lon, lat, h[0, 0, :, :], cmap='cividis')

1. **U10M, SLP, lon, lat**



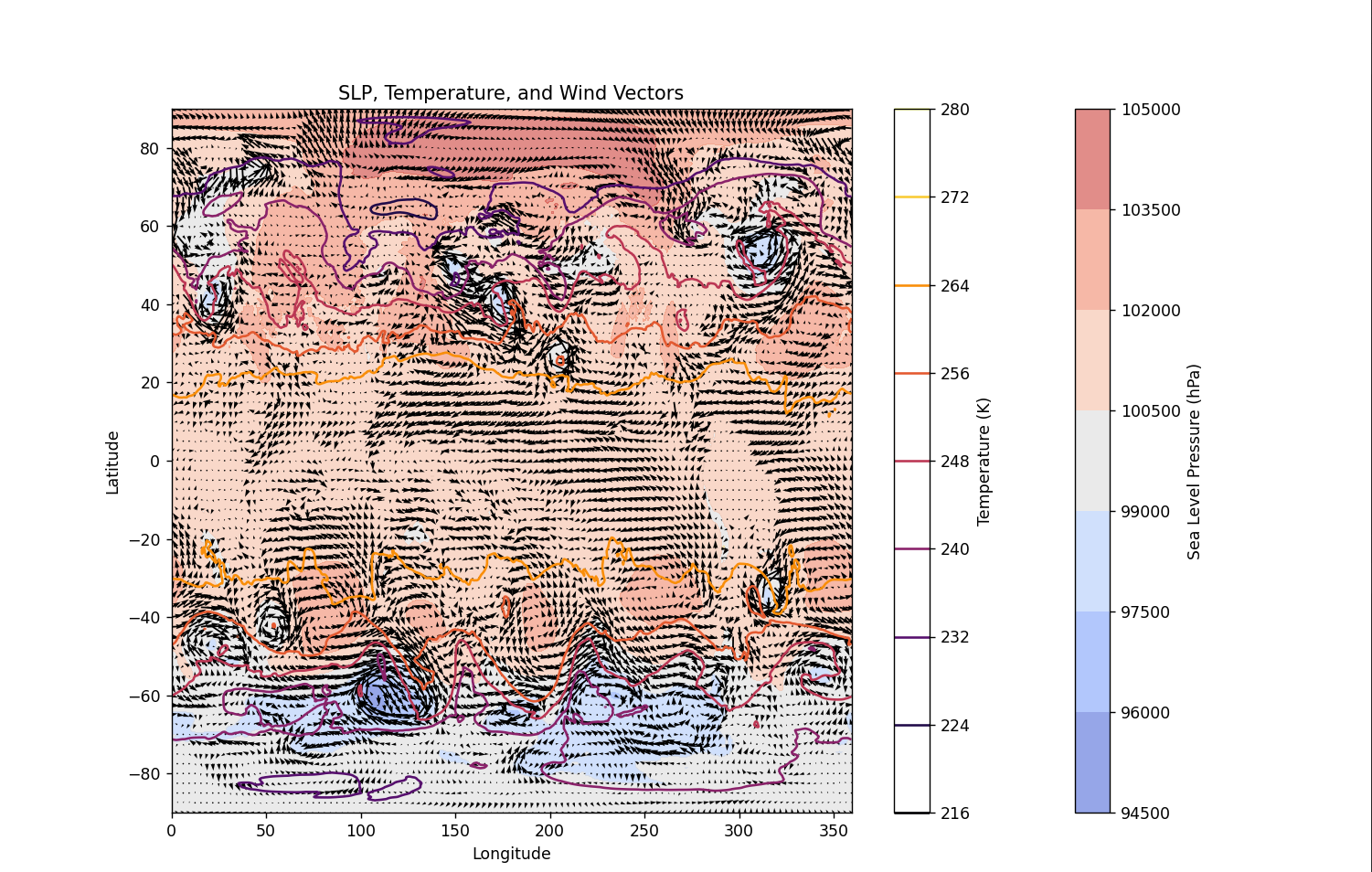
plt.figure(figsize=(10, 6))  
plt.contourf(lon, lat, u10m[0, :, :], cmap='viridis')  
plt.colorbar(label='U10M Wind Component')  
plt.contour(lon, lat, slp[0, :, :], colors='white')

1. **SLP, U10M, V10M**



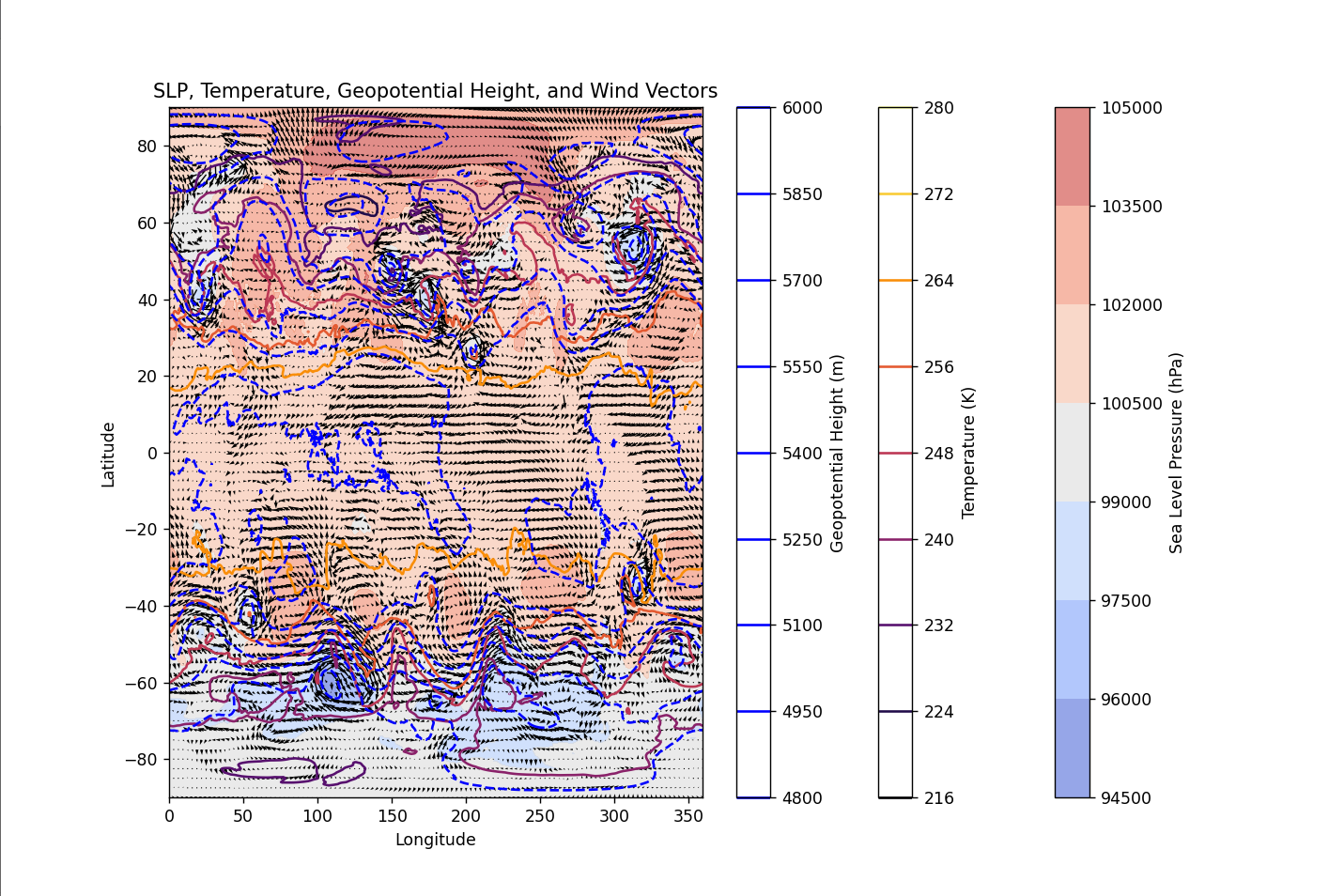
plt.contourf(lon, lat, slp[0, :, :], cmap='coolwarm', alpha=0.6)  
plt.colorbar(label='Sea Level Pressure (hPa)')  
plt.quiver(lon[::5], lat[::5], u10m[0, ::5, ::5], v10m[0, ::5, ::5], scale=500, color='black')

1. **SLP、U10M、V10M、Temperature**



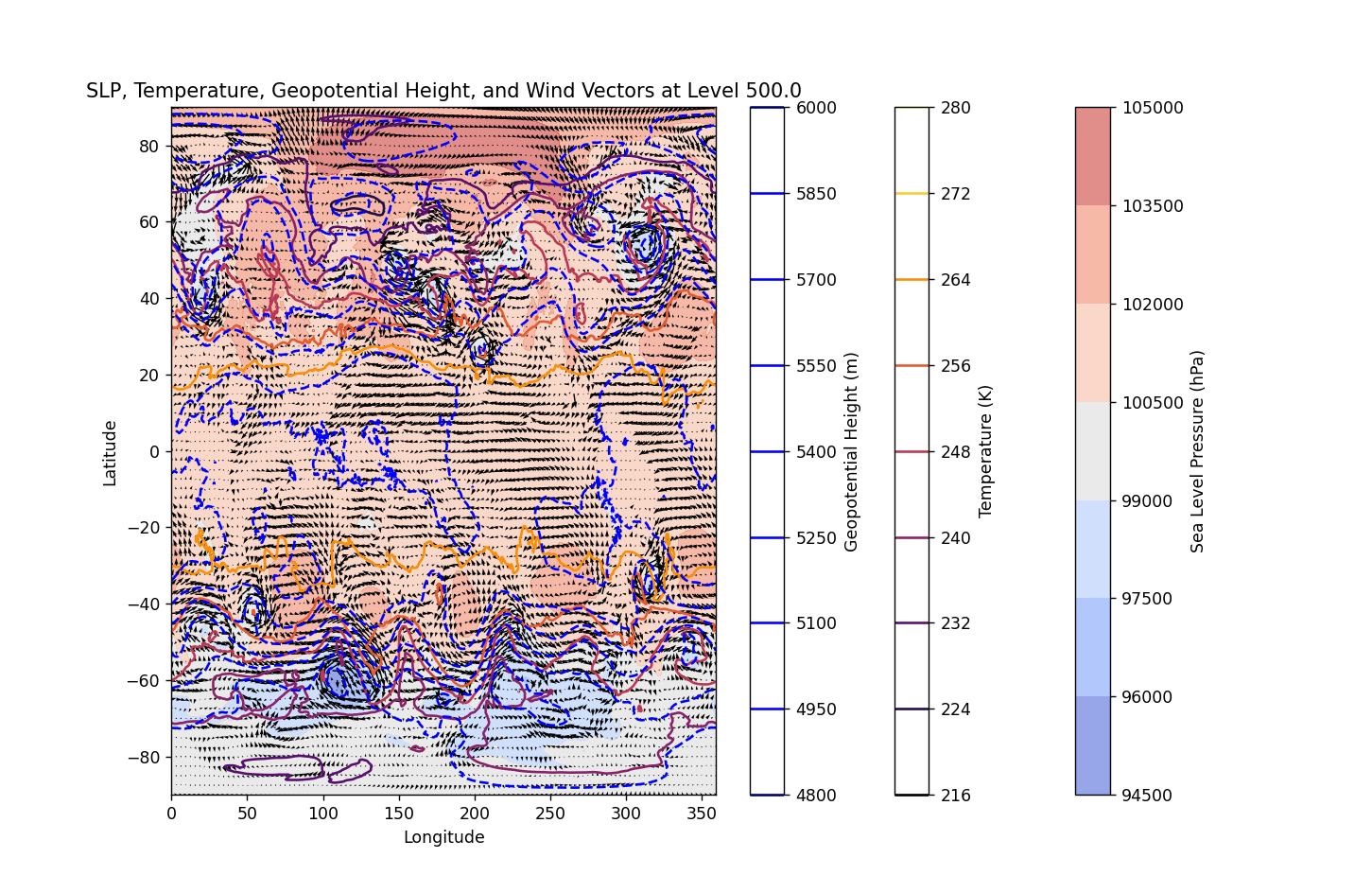
plt.contourf(lon, lat, slp[0, :, :], cmap='coolwarm', alpha=0.6)  
plt.colorbar(label='Sea Level Pressure (hPa)')  
plt.contour(lon, lat, temperature, cmap='inferno')  
plt.colorbar(label='Temperature (K)')  
plt.quiver(lon[::5], lat[::5], u10m[0, ::5, ::5], v10m[0, ::5, ::5], scale=500, color='black')

1. **SLP、U10M、V10M、Temperature、Geopotential Height**



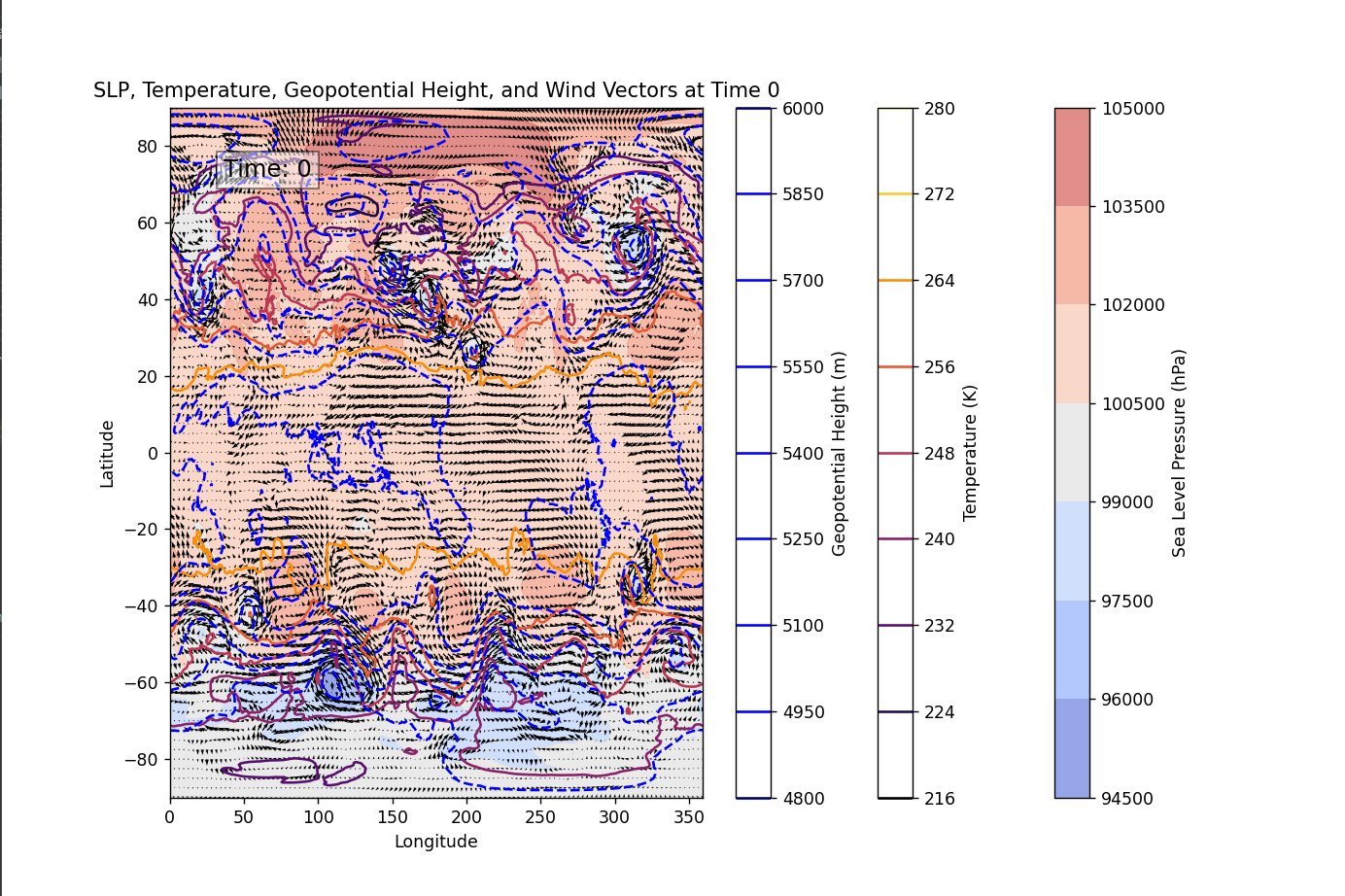
height = dataset.variables['H'][0, 0, :, :] # 在第0层高度上  
plt.figure(figsize=(12, 8))  
plt.contourf(lon, lat, slp[0, :, :], cmap='coolwarm', alpha=0.6)  
plt.colorbar(label='Sea Level Pressure (hPa)')  
plt.contour(lon, lat, temperature, cmap='inferno')  
plt.colorbar(label='Temperature (K)')  
plt.contour(lon, lat, height, colors='blue', linestyles='dashed')  
plt.colorbar(label='Geopotential Height (m)')  
plt.quiver(lon[::5], lat[::5], u10m[0, ::5, ::5], v10m[0, ::5, ::5], scale=500, color='black')

1. **SLP、U10M、V10M、Temperature、Geopotential Height、lev**



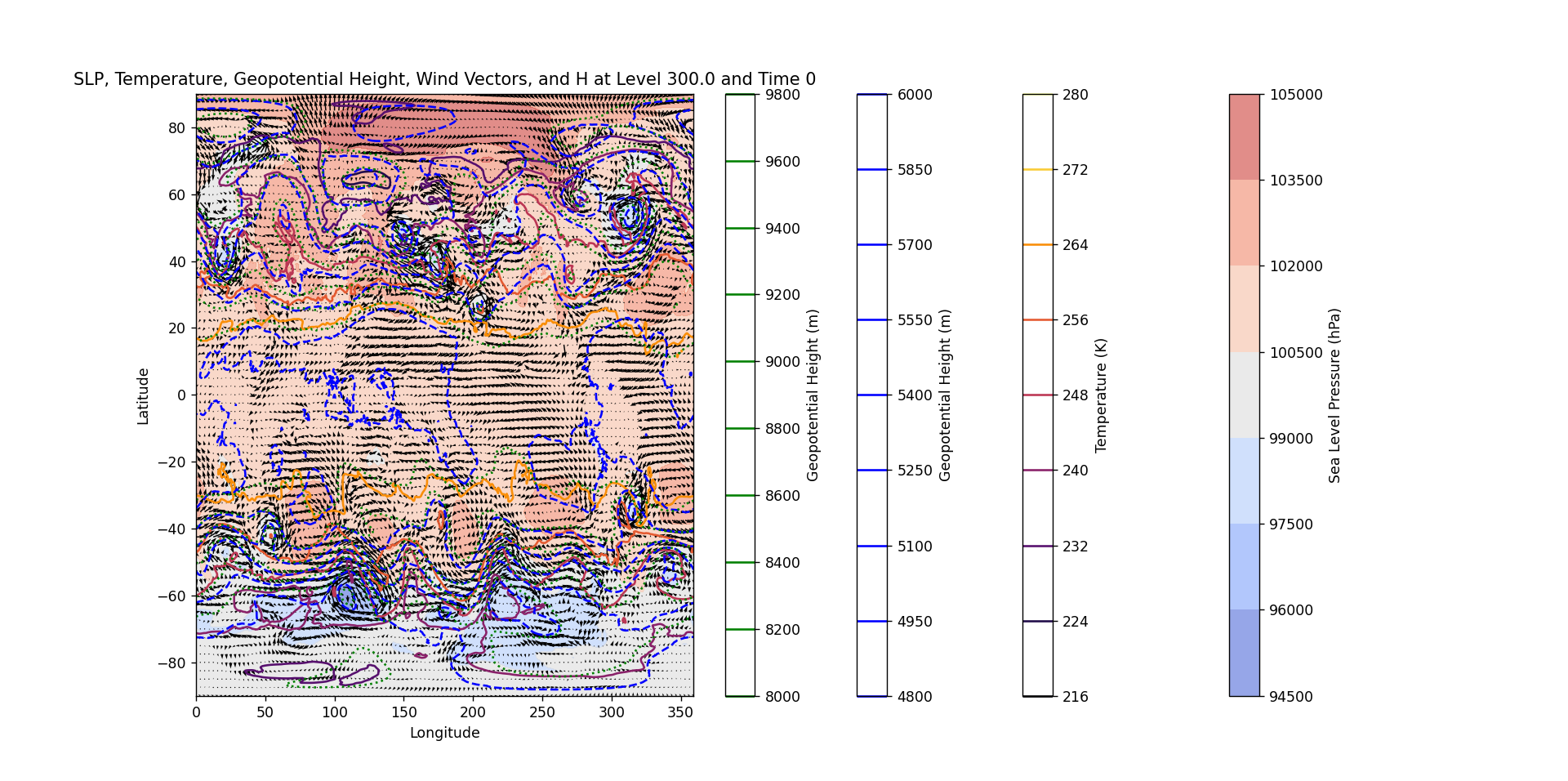
lev = dataset.variables['lev'][:]  
selected\_level = 0   
plt.figure(figsize=(12, 8))  
plt.contourf(lon, lat, slp[selected\_level, :, :], cmap='coolwarm', alpha=0.6)  
plt.colorbar(label='Sea Level Pressure (hPa)')  
plt.contour(lon, lat, temperature, cmap='inferno')  
plt.colorbar(label='Temperature (K)')  
plt.contour(lon, lat, height, colors='blue', linestyles='dashed')  
plt.colorbar(label='Geopotential Height (m)')  
plt.quiver(lon[::5], lat[::5], u10m[selected\_level, ::5, ::5], v10m[selected\_level, ::5, ::5], scale=500, color='black')

1. **SLP、U10M、V10M、Temperature、Geopotential Height、lev、time**



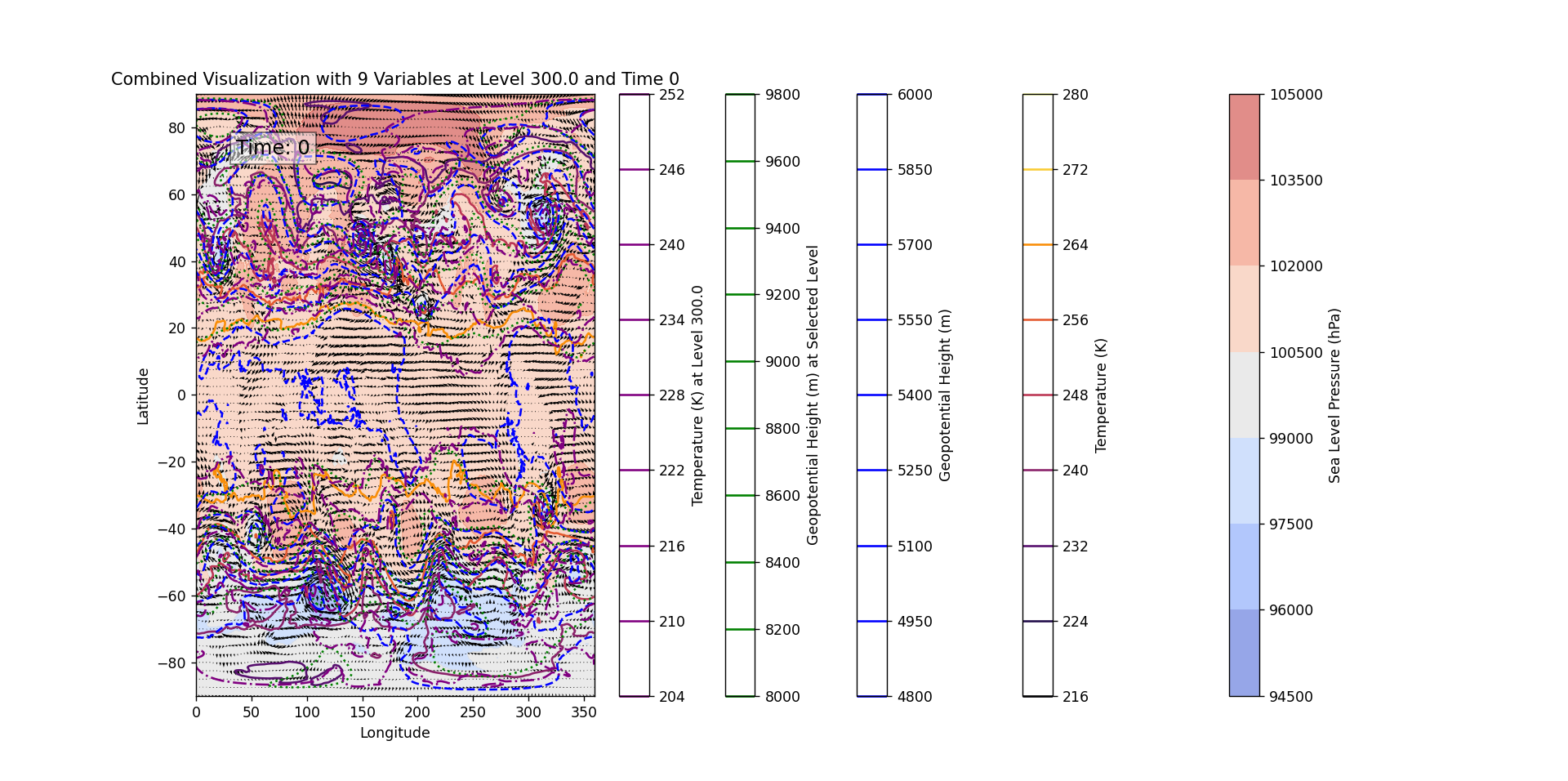
plt.contourf(lon, lat, slp[0, :, :], cmap='coolwarm', alpha=0.6)  
plt.colorbar(label='Sea Level Pressure (hPa)')  
plt.contour(lon, lat, temperature, cmap='inferno')  
plt.colorbar(label='Temperature (K)')  
plt.contour(lon, lat, height, colors='blue', linestyles='dashed')  
plt.colorbar(label='Geopotential Height (m)')  
plt.quiver(lon[::5], lat[::5], u10m[0, ::5, ::5], v10m[0, ::5, ::5], scale=500, color='black')  
plt.text(0.1, 0.9, f'Time: {time[0]}', transform=plt.gca().transAxes, fontsize=14, bbox=dict(facecolor='white', alpha=0.5))

1. **SLP、U10M、V10M、Temperature、Geopotential Height、lev、time、H**



plt.contourf(lon, lat, slp[selected\_time, :, :], cmap='coolwarm', alpha=0.6)  
plt.colorbar(label='Sea Level Pressure (hPa)')  
plt.contour(lon, lat, temperature, cmap='inferno')  
plt.colorbar(label='Temperature (K)')  
plt.contour(lon, lat, height, colors='blue', linestyles='dashed')  
plt.colorbar(label='Geopotential Height (m)')  
plt.quiver(lon[::5], lat[::5], u10m[selected\_time, ::5, ::5], v10m[selected\_time, ::5, ::5], scale=500, color='black')  
plt.contour(lon, lat, dataset.variables['H'][selected\_time, selected\_level, :, :], colors='green', linestyles='dotted')  
plt.colorbar(label='Geopotential Height (m)')

1. **SLP、U10M、V10M、Temperature、Geopotential Height、lev、time、H、T**



plt.contourf(lon, lat, slp[selected\_time, :, :], cmap='coolwarm', alpha=0.6)  
plt.colorbar(label='Sea Level Pressure (hPa)')  
plt.contour(lon, lat, temperature, cmap='inferno')  
plt.colorbar(label='Temperature (K)')  
plt.contour(lon, lat, height, colors='blue', linestyles='dashed')  
plt.colorbar(label='Geopotential Height (m)')  
plt.quiver(lon[::5], lat[::5], u10m[selected\_time, ::5, ::5], v10m[selected\_time, ::5, ::5], scale=500, color='black')  
plt.contour(lon, lat, dataset.variables['H'][selected\_time, selected\_level, :, :], colors='green', linestyles='dotted')  
plt.colorbar(label='Geopotential Height (m) at Selected Level')  
plt.contour(lon, lat, dataset.variables['T'][selected\_time, selected\_level, :, :], colors='purple', linestyles='dashdot')  
plt.colorbar(label=f'Temperature (K) at Level {lev[selected\_level]}')  
plt.text(0.1, 0.9, f'Time: {time[selected\_time]}', transform=plt.gca().transAxes, fontsize=14, bbox=dict(facecolor='white', alpha=0.5))

③Different variables in dynamic time

1. SLP, lon, lat



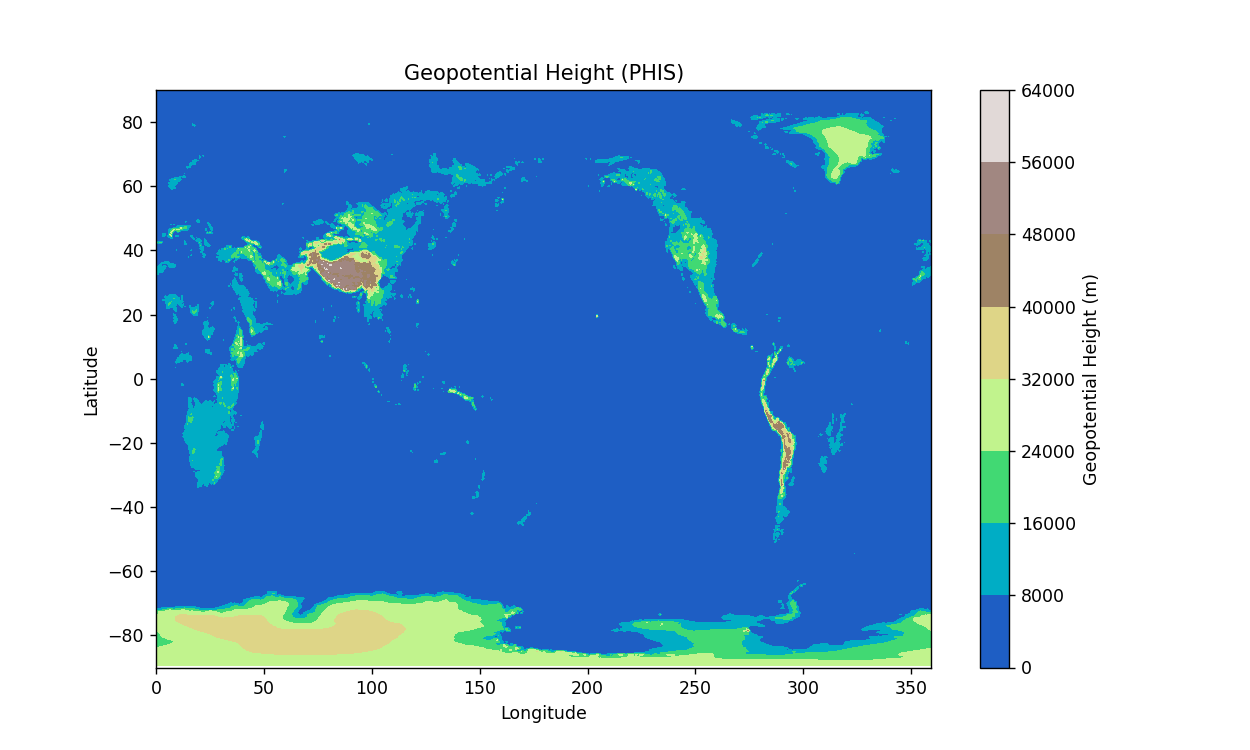
slp = dataset.variables['SLP'][:]  
lat = dataset.variables['lat'][:]  
lon = dataset.variables['lon'][:]  
time = dataset.variables['time'][:]  
  
  
start\_day = 0 # 第一天  
end\_day = 7 # 选择7天  
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(15, 10), constrained\_layout=True)  
lat\_mid = lat.shape[0] // 2  
lon\_mid = lon.shape[0] // 2  
  
for i in range(start\_day, end\_day):  
 ax = axes.flatten()[i]  
 cont = ax.contourf(lon, lat, slp[i, :, :], cmap='coolwarm')  
 ax.set\_title(f"Day {i + 1}: Sea Level Pressure")  
 ax.set\_xlabel('Longitude')  
 ax.set\_ylabel('Latitude')  
fig.colorbar(cont, ax=axes[:, -1], orientation='vertical', fraction=0.05, pad=0.05)

II.

Please see python code for dynamic graph.

Problem: How to predict?

1. **MERRA2.topo.nc** (NetCDF) file



phis = dataset.variables['PHIS'][:]   
lat = dataset.variables['lat'][:]   
lon = dataset.variables['lon'][:]   
plt.figure(figsize=(10, 6))  
plt.contourf(lon, lat, phis, cmap='terrain')  
plt.colorbar(label='Geopotential Height (m)')

1. **Reading paper**
2. **Problems:**

The shape of earth and where is the data?