import numpy as np

from obspy import Trace, Stream, UTCDateTime

import matplotlib.pyplot as plt

import os

# def read\_ascii\_file(filename):

# data = np.loadtxt(filename)

# displacement\_data = data[:, 1] # 假设第二列包含位移数据

# tr = Trace(data=displacement\_data)

# tr.stats.starttime = UTCDateTime() # 您可以设置合适的起始时间

# tr.stats.sampling\_rate = 100 # 设置为实际的采样率

# return tr

def read\_ascii\_file(filename):

data = np.loadtxt(filename)

displacement\_data = data[:, 1] # 假设第二列包含位移数据

tr = Trace(data=displacement\_data)

tr.stats.network = "XX" # 您可以根据需要设置网络代码

tr.stats.station = os.path.basename(filename).split(".")[0] # 使用文件名作为台站名称

tr.stats.location = "00" # 您可以根据需要设置位置代码

tr.stats.channel = "X" # 假设是X分量，您可以根据需要进行调整

tr.stats.starttime = UTCDateTime() # 您可以设置合适的起始时间

tr.stats.sampling\_rate = 100 # 设置为实际的采样率

return tr

def read\_multiple\_ascii\_files(filenames):

st = Stream()

for file in filenames:

tr = read\_ascii\_file(file)

st.append(tr)

return st

def get\_files\_with\_extensions(directory, extensions):

files = []

for file in os.listdir(directory):

for ext in extensions:

if file.endswith(ext):

files.append(file)

break

return files

# # 对文件进行分类

# def sort\_files\_by\_component(files):

# components = {'X': [], 'Y': [], 'Z': []}

# for file in files:

# component = file.split('.')[-2][-1]

# if component in components:

# components[component].append(file)

# return components

def plot\_section\_subplot(x\_stream, ax, scale=1, yscale=1e9):

min\_y, max\_y = float('inf'), float('-inf')

for i, tr in enumerate(x\_stream):

t = np.arange(0, tr.stats.npts / tr.stats.sampling\_rate, tr.stats.delta)

y = tr.data \* scale \* yscale + i

ax.plot(t, y, color='k', linewidth=0.5)

min\_y = min(min\_y, y.min())

max\_y = max(max\_y, y.max())

ax.set\_ylim(min\_y - 1, max\_y + 1)

ax.set\_xlabel('Time (s)')

ax.set\_ylabel('Displacement (m)')

def create\_2d\_array(stream):

data\_list = []

for tr in stream:

data\_list.append(tr.data)

return np.array(data\_list)

# def plot\_2d\_image(data\_2d, cmap='seismic', interpolation=None, vmin=-1, vmax=1):

# fig, ax = plt.subplots(figsize=(12, 6))

# im = ax.imshow(data\_2d, cmap=cmap, aspect='auto', interpolation=interpolation, vmin=vmin, vmax=vmax)

# ax.set\_xlabel('Time (samples)')

# ax.set\_ylabel('Trace index')

# ax.set\_title('X Component 2D Image')

# fig.colorbar(im, ax=ax)

# plt.show()

main\_dir = r"E:\workdata\_lhy\OUTPUT\_FILES"

# 获取符合条件的文件

semv\_files = get\_files\_with\_extensions(main\_dir, ['.semv'])

# 对文件进行分类

sorted\_files = sort\_files\_by\_component(semv\_files)

# # 获取X分量的文件路径

# x\_files = sorted\_files['X']

# full\_paths = [os.path.join(main\_dir, file) for file in x\_files]

# # 读取X分量文件并创建Stream对象

# x\_stream = read\_multiple\_ascii\_files(full\_paths)

def sort\_files\_by\_component(files):

components = {'X': [], 'Y': [], 'Z': []}

for file in files:

station, component = os.path.basename(file).split('.')[0], file.split('.')[-2][-1]

if component in components:

components[component].append((station, file))

return components

print(x\_files)

# 获取X分量的文件路径

x\_files = sorted\_files['X']

full\_paths = [os.path.join(main\_dir, file) for \_, file in x\_files]

# 读取X分量文件并创建Stream对象

x\_stream = read\_multiple\_ascii\_files(full\_paths)

# 创建二维数组

x\_data\_2d = create\_2d\_array(x\_stream)

# 绘制二维图像

# plot\_2d\_image(x\_data\_2d, vmin=x\_data\_2d.min() \* 0.5, vmax=x\_data\_2d.max() \* 0.5)

# plot\_2d\_image(x\_data\_2d, cmap='jet', vmin=x\_data\_2d.min(), vmax=x\_data\_2d.max())

# plot\_2d\_image(x\_data\_2d, cmap='jet', vmin=vmin, vmax=vmax)

# vmin = np.percentile(x\_data\_2d, 10)

# vmax = np.percentile(x\_data\_2d, 90)

# plot\_2d\_image(x\_data\_2d, cmap='seismic', vmin=vmin, vmax=vmax)

# vmin = np.percentile(x\_data\_2d, 25)

# vmax = np.percentile(x\_data\_2d, 75)

# plot\_2d\_image(x\_data\_2d, cmap='seismic', vmin=vmin, vmax=vmax)

# 尝试更改np.percentile的参数

vmin = np.percentile(x\_data\_2d, 5)

vmax = np.percentile(x\_data\_2d, 95)

# 对数据进行归一化，然后再应用对数变换

x\_data\_2d\_norm = (x\_data\_2d - np.min(x\_data\_2d)) / (np.max(x\_data\_2d) - np.min(x\_data\_2d))

x\_data\_2d\_log\_norm = np.log10(np.abs(x\_data\_2d\_norm) + 1)

vmin\_log\_norm = np.percentile(x\_data\_2d\_log\_norm, 10)

vmax\_log\_norm = np.percentile(x\_data\_2d\_log\_norm, 90)

def plot\_2d\_image(data\_2d, cmap='seismic', interpolation=None, aspect='auto', vmin=-1, vmax=1):

fig, ax = plt.subplots(figsize=(12, 6))

im = ax.imshow(data\_2d, cmap=cmap, aspect=aspect, interpolation=interpolation, vmin=vmin, vmax=vmax)

ax.set\_xlabel('Time (samples)')

ax.set\_ylabel('Trace index')

ax.set\_title('X Component 2D Image')

fig.colorbar(im, ax=ax)

plt.show()

# 使用aspect='equal'参数来保持图像的纵横比

plot\_2d\_image(x\_data\_2d\_log\_norm, cmap='seismic', aspect='equal', vmin=vmin\_log\_norm, vmax=vmax\_log\_norm)

# 调整plt.subplots函数中的figsize参数

def plot\_2d\_image\_adjusted\_size(data\_2d, cmap='seismic', interpolation=None, aspect='auto', figsize=(12, 6), vmin=-1, vmax=1):

fig, ax = plt.subplots(figsize=figsize)

im = ax.imshow(data\_2d, cmap=cmap, aspect=aspect, interpolation=interpolation, vmin=vmin, vmax=vmax)

ax.set\_xlabel('Time (samples)')

ax.set\_ylabel('Trace index')

ax.set\_title('X Component 2D Image')

fig.colorbar(im, ax=ax)

plt.show()

# 您可以根据需要调整图像的宽度和高度

plot\_2d\_image\_adjusted\_size(x\_data\_2d\_log\_norm, cmap='seismic', aspect='equal', figsize=(16, 8), vmin=vmin\_log\_norm, vmax=vmax\_log\_norm)

# x\_data\_2d\_log = np.log10(np.abs(x\_data\_2d) + 1) # 添加1以避免对0取对数

# vmin\_log = np.percentile(x\_data\_2d\_log, 10)

# vmax\_log = np.percentile(x\_data\_2d\_log, 90)

# plot\_2d\_image(x\_data\_2d\_log, cmap='seismic', vmin=vmin\_log, vmax=vmax\_log)

# x\_data\_2d\_log = np.log10(np.abs(x\_data\_2d) + 1) # 添加1以避免对0取对数

# plot\_2d\_image(x\_data\_2d\_log, cmap='seismic', vmin=x\_data\_2d\_log.min(), vmax=x\_data\_2d\_log.max())

[('DB', 'DB.X01.FXX.semv'), ('DB', 'DB.X02.FXX.semv'), ('DB', 'DB.X03.FXX.semv'), ('DB', 'DB.X04.FXX.semv'), ('DB', 'DB.X05.FXX.semv'), ('DB', 'DB.X06.FXX.semv'), ('DB', 'DB.X07.FXX.semv'), ('DB', 'DB.X08.FXX.semv'), ('DB', 'DB.X09.FXX.semv'), ('DB', 'DB.X10.FXX.semv'), ('DB', 'DB.X11.FXX.semv'), ('DB', 'DB.X12.FXX.semv'), ('DB', 'DB.X13.FXX.semv'), ('DB', 'DB.X14.FXX.semv'), ('DB', 'DB.X15.FXX.semv'), ('DB', 'DB.X16.FXX.semv')]

