

文章编号: 1001-1749(2023)01-0121-08

基于 Python 的 MTU 系列大地电磁测深仪 时间序列读写与应用

杨 凯

(中国地质调查局 西安矿产资源调查中心, 西安 710000)

摘要: 加拿大凤凰地球物理有限公司 MTU 系列大地电磁测深仪是国内大地电磁测深工作中较为常用的仪器,但是其特殊的时间序列数据格式让其后期分析处理较为不便,为了进一步分析研究时间序列为后续处理和反演工作提供便利,需要对时间序列文件进行读写操作。鉴于 Python 语言的众多优点,在充分掌握 MTU 系列大地电磁测深仪时间序列 TSn 文件存储格式的基础上,利用 Python 实现了 TSn 文件的读写操作和基于希尔伯特-黄变换的时频分析功能并编写了软件界面,为 Python 在大地电磁领域的进一步应用提供了参考。

关键词: 大地电磁; 时间序列; Python; MTU

中图分类号: P 631.2 **文献标志码:** A **DOI:** 10.3969/j.issn.1001-1749.2023.01.16

0 引言

Python 是一种面向对象、解释型计算机程序设计语言,诞生于 20 世纪 90 年代,现在它已经成为最受欢迎的程序设计语言之一,它具有简洁、易读及可扩展性等诸多优点,所以利用 Python 做科学计算的研究机构日益增多,丰富的第三方包为 Python 提供了快速的数组处理、数值运算及绘图功能,因此 Python 语言以及众多的功能库所构建的开发环境十分适合工程技术、科研人员处理实验数据、进行数据可视化及各类界面软件的开发,国内、外已有学者进行了基于 Python 的大地电磁处理研究,比如国外学者 Lars Krieger 等^[1]编写了基于 Python 的大地电磁处理库 MTpy,国内学者雷清等^[2]将 Python 应用于 TBL 文件处理,这些都为 Python 在大地电磁领域的应用开辟了路径。

大地电磁是较为常用的物探方法,因其施工便

利,勘探深度大,适应性强而广泛应用于能源、资源和工程勘查。目前国内主要采用的是加拿大凤凰公司的 MTU 系列大地电磁仪,该仪器具有轻便、稳定等众多优点,但是其时间序列独特的数据结构只能在其配套软件上读取和处理,比如 Synchro Time Series View 只能查看时间序列波形和求取功率谱,SSMT2000 只能在频率域内进行数据处理。随着近年来许多时间域处理技术日渐成熟,为了在时间域研究大地电磁信号,特别是开展时间域去噪等工作,急需可以方便读取时间序列的方法,为此许多学者对各原始数据的格式和读取方式展开了研究^[3-6],基于此,笔者利用 Python 的 Numpy 库(Numerical Python,是高性能科学计算和数据分析的 Python 基础包),SciPy 库(Scientific Python 科学数值运算库),PyEMD 库(经验模态分解库),Matplotlib 库(建立在 Numpy 数组基础上的多平台数据可视化程序库)和 PyQt5 库(创建 Python GUI 应用程序的工具包),Datetime 库(处理时间信息的库),在充分

收稿日期: 2021-02-18

基金项目: 中国地质调查局项目(DD20211552,DD20208008)

第一作者: 杨凯(1991-),男,硕士,工程师,主要从事物探数据处理工作,E-mail: yangkaicgs@163.com。

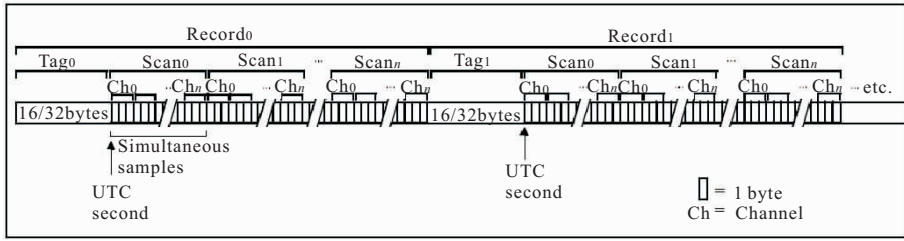


图 1 TSn 文件存储格式

Fig. 1 The data format of TSn files

表 1 Tag 格式一览表

Tab. 1 The meaning of each byte of the tag

字节	意义
0—7	UTC 时间的第一个 Scan 记录
0	秒
1	分
2	时
3	日
4	月
5	年(后两位数字)
6	星期几
7	世纪
8—9	盒子序列号(16 位整数)
10—11	记录中 Scan 的个数(16 位整数)
12	每次 Scan 的道数
13	Tag 的长度
14	状态代码
15	位饱和和标志
16	保留, 未来表示不同的 tag 或采样格式的标记
17	采样长度
18—19	采样率(单位时间的采样次数)
20	采样率单位
21	时钟状态
22—25	时钟误差(单位:微妙)
26—31	保留, 必须为 0

了解 MTU 大地电磁仪时间序列存储格式的基础上^[7], 编写了时间序列可视化读写分析程序, 并用实测数据验证了其正确性。

1 TSn 时间序列文件格式

MTU 系列大地电磁仪采集的原始文件有数据采集参数文件(*. TBL)、磁探头标定文件(*. CLC)、盒子标定文件(*. CLB)以及时间序列文件(*. TSn), 其中时间序列文件是原始数据最重要的

组成部分, 它存储了所有采集到的信息。时间序列文件后缀 n 对应不同的频带, *. TS2 是 24 000 Hz 的采样率数据, 存储高频数据; *. TS3 是 2 400 Hz 的采样率数据, 存储中高频数据; *. TS4 是 150 Hz 的采样率数据, 存储中低频数据; *. TS5 是 15 Hz 的采样率数据, 存储低频数据, 时间序列存储格式如图 1 所示。

从图 1 可知, 时间序列由若干个 Record(记录)组成, 每个 Record 又包含了一个 Tag(标记)和若干个 Scan(扫频), 每个 Scan 存储了该次扫频中每个 Channel(通道)的数据, 每个 Channel 数据以 24 位补码保存, 占 3 个字节, 在大地电磁工作中每个测点记录了 E_x 、 E_y 、 H_x 、 H_y 、 H_z 共 5 个分量, 对应 5 个 Channel。在 TSn 文件中 Tag 占 32 个字节, 它相当于数据头, 主要存储了本次 Record 的各种采集信息, 具体意义见表 1。

可见, 要读取时间序列必须先读取每个 Record 的 Tag 信息, 比较重要的有盒子序列号、记录中 Scan 的个数、采样率、采样率单位、每次 Scan 的道数以及采样时间等信息。而后设置相应参数读取每道 Scan 中的数据, 循环这个操作即可读取整个时间序列文件。

2 TSn 文件读写与应用

在充分了解 TSn 时间序列文件存储格式的基础上, 将利用 Python 语言对时间序列文件进行读写操作并进行了时频分析及图形化界面编制的初步应用。

2.1 TSn 文件读取

MTU 系列大地电磁仪器及其配套软件, 是一套完备的大地电磁测深软硬件系统, 自成一体, 但是也造成了自带软件封闭、功能单一等缺点。由于近年大地电磁时间序列时频分析及时间域处理技术的

迅速发展,为我们提供了新的大地电磁处理路径,也急需一种便捷快速的大地电磁时间序列读写方法,以便于做各种分析和处理。这里主要利用了 Python 的 Numpy 库、Matplotlib 库、PyQt5 库和 Datetime 库用以读写和显示时间序列图形。编程的主要思路为:①根据 TS_n 文件的特定格式,在 Record 的 Tag 中读取 Scan 的个数、采样率、采样率单位、每次 Scan 的道数以及采样时间等信息,并不断调整循环结构的参数直到读完整个 TS_n 文件,读取的原始数据为 24 位补码保存,每个占 3 个字节;②二进制格式转化为十进制,以 Record 为单位进行时间序列图形显示。各步骤及核心代码如下:

1) 首先将 TS_n 文件以 uint8 类型读入数组 series_arr 中。

```
series_arr = np.fromfile(file_name, dtype='uint8') # np 为模块 Numpy 的缩写, file_name 是文件路径
```

```
series_size = series_arr.size # 获取文件字节数
```

2) 在第一个 Record 读取 Tag 中的时间序列参数。

```
box_serial = bin_int_func(series_arr[8:10]) # 获取盒子序列号 bin_int_func 为自写二进制整数转 10 进制函数
```

```
scan_number = bin_int_func(series_arr[10:12]) # 获取 Scan 数
```

```
sampling_rate = bin_int_func(series_arr[18:20]) # 获取采样率
```

```
channel_number_arr = series_arr[12] # 获取存储 Channel 数的数组值
```

```
channel_number = bin_func('{:08b}'.format(channel_number_arr)) # 将二进制转换成十进制获取 Channel 数, bin_func 为自写二进制整型转十进制函数
```

```
record_number = int(series_size / (series_arr[13] + scan_number * channel_number * 3)) # 计算 Record 个数
```

```
for n in range(0, record_number): # 利用循环提取每个 Record 的开始采集时间
```

```
utc_arr = series_arr[(series_arr[13] + scan_number * channel_number * 3) * n : (series_arr[13] + scan_number * channel_number * 3) * n + 8]
```

```
utc_datetime = datetime.datetime.strptime(recognize_utc_para_func(utc_arr), "%Y/%m/%d %H:%M:%S") # 将时间数据格式化, datetime
```

为 Python 的一个时间模块用于处理各种时间数据, recognize_utc_para_func 为时间拼接函数, 将时分秒拼接起来。

```
utc_list.append(utc_datetime) # 将每个 Record 的采集开始时间存入 utc_list 列表中, 用于后续操作。
```

3) 最后根据读取的时间序列参数循环读取时间序列数据。

```
data_list = [] # 每次读取数据前置空列表
```

```
for i in range(m, n): # 该层循环遍历 Record, 由于 TSn 文件数据量较大, 一般采取片段方式读取和显示, 根据需要查看某个时间段的数据。
```

```
for j in range(0, scan_number): # 遍历 Scan
```

```
for k in range(0, channel_number): # 遍历 Channel
```

```
para_arr = series_arr[series_arr[13] + k * 3 + j * 3 * channel_number + (series_arr[13] + scan_number * channel_number * 3) * i : series_arr[13] + 3 + k * 3 + j * 3 * channel_number + (series_arr[13] + scan_number * channel_number * 3) * i] # 每三个字节为一组, 读取二进制数据
```

```
data_list.append(bin_int_func(para_arr)) # 将二进制数据转换为 10 进制后存入列表
```

```
data_arr = np.array(data_list).reshape(scan_number * (n - m), channel_number).T # 将一维数组转换为二维数组并转置得到存储每个电磁场分量的二维数组, 此数据即可成图或者进行下一步处理。
```

4) 关键函数解析。

```
def bin_int_func(series):
```

```
    binrow = "
```

```
    for i in range(1, len(series) + 1):
```

```
        binrowi = '{:08b}'.format(series[0 - i])
```

```
        binrow += binrowi
```

```
    return bin_func(binrow) # 该函数也是基于 bin_func 函数进行二进制转十进制的
```

```
def bin_func(bin_series): # 二进制整型转十进制函数
```

```
    Sign = int(bin_series[0]) # 读取符号位, 负数的符号位是 1, 正数为 0
```

```
    int_sum = 0
```

```
    if Sign == 1:
```

```
        negation_bin_series = "
```

```
        for i in range(len(bin_series)): # 负值每位取相反值
```

```
            v = str(1 - int(bin_series[i]))
```

```

negation_bin_series += v # 拼接二进制数据
bin_series = bin(int(negation_bin_series, 2)
+ int('1', 2))[2:] # 加 1 后再转为二进制
for i, e in enumerate(bin_series): # 二进制转
十进制
    int_sum += int(e) * 2 ** (len(bin_series)
-i-1)
int_sum = -int_sum # 取负值
else:
for i, e in enumerate(bin_series): 正值直接计
算其十进制数即可
    int_sum += int(e) * 2 ** (len(bin_series)
-i-1)
return int_sum

```

2.2 写 TSn 文件

在我们对时间序列进行处理后就需要重新写回 TSn 文件中用于后续处理,与读取刚好相反,核心步骤就是十进制转 24 位补码二进制,核心代码如下:

```

_, b_number = bin(d_number).split('b') # 将
十进制整数 d_number 转换为二进制数 b_number
if len(b_number) <= 24:

```

```

    if d_number >= 0:
        b_number_all = b_number.rjust(24, '0') # 如
果为正数左侧用 0 补齐至 24 位
    else:
        _, b_number_all = bin(2 ** 24 + d_num-
ber).split('b') # 如果为负数采取 24 位补码转换
        list = []
        for i in range(3): # 将 24 位二进制序列每 8 位
一组转换成 10 进制
            b_seg = b_number_all[(2 - i) * 8:(3 - i)
* 8]
            int_sum = 0
            for j, e in enumerate(b_seg): # 将三个十进制
数反方向存入列表 list
                int_sum += int(e) * 2 ** (len(b_seg) -
j - 1)
            list.append(int_sum)
        转换完成后替换 series_arr 数组中相应的数据
块,保存为 TSn 文件即可,具体代码如下:
        series_arr.tolist('*.TSn') # 与 fromfile 函数
相反, tofile 函数将数组写入文件。

```

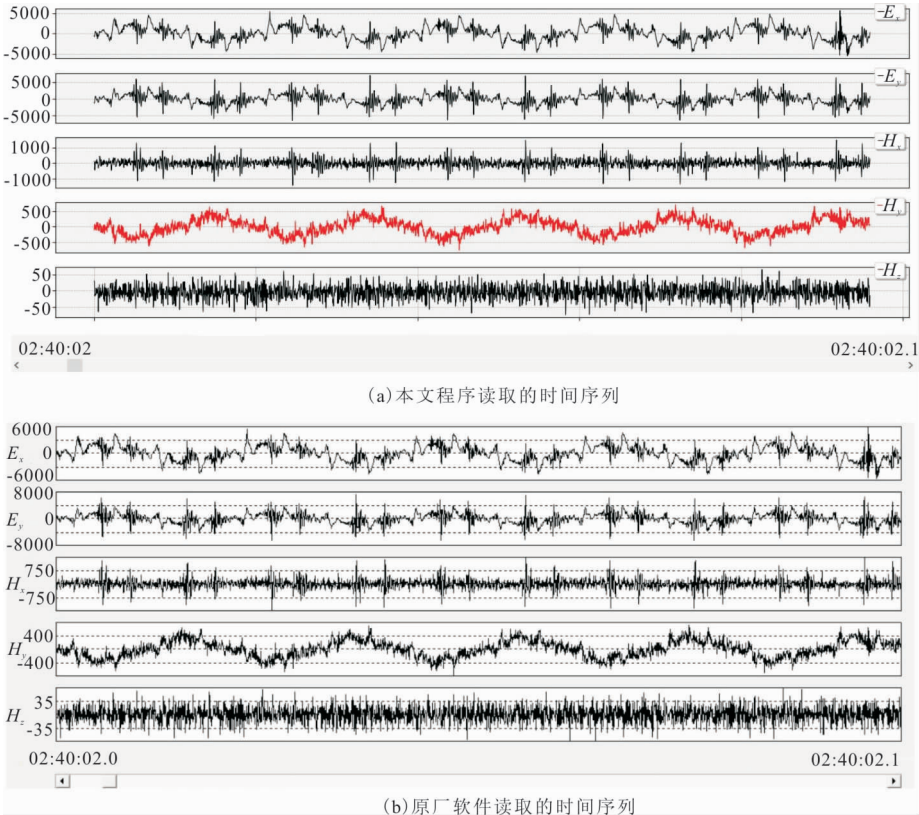


图 2 时间序列读取对比

Fig. 2 The comparison curves of AMT data from this program and original program

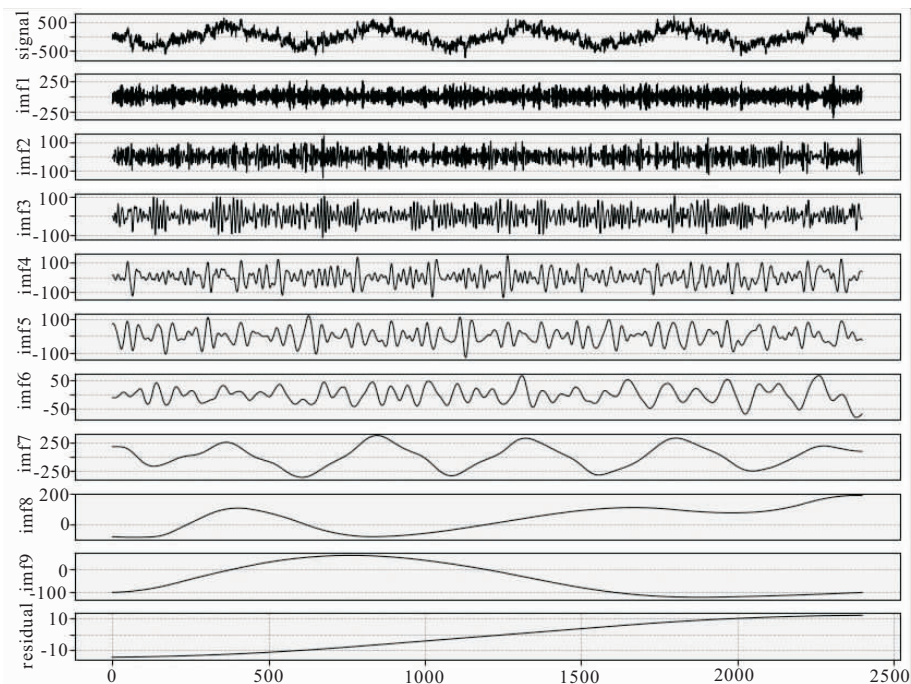


图 3 H_y 分量 EMD 分解图(Matplotlib 生成)

Fig. 3 EMD decomposition of H_y signal

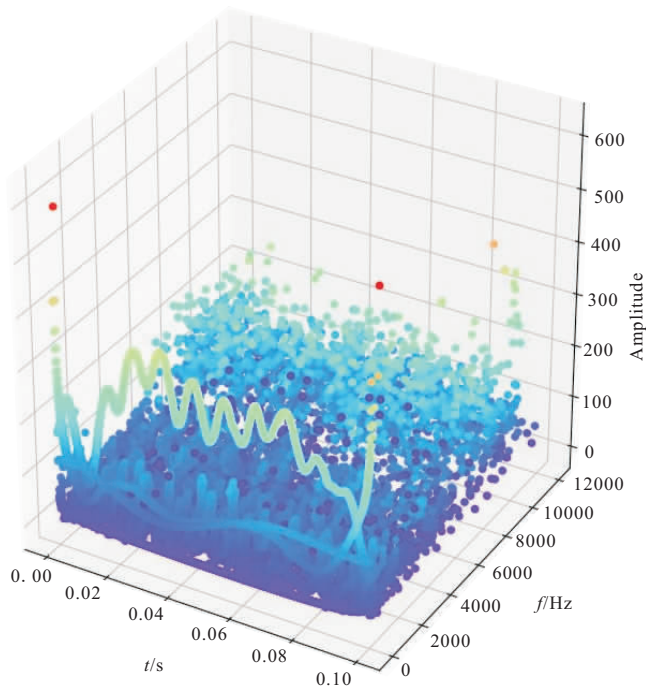


图 4 H_y 分量 HHT 三维时频图(Matplotlib 生成)

Fig. 4 HHT time-frequency spectrum of H_y signal

2.3 实际应用

由于 2020 年 10 月使用 MTU-5A 大地电磁仪在陕西安强县某地进行大地电磁测量时,500 m 范围内有一钻机施工,对数据造成了极大干扰,故将本文的读写及分析方法应用于该段干扰数据以分析

噪声特征,该段时间序列文件名为 4711A12A.TS2,数据采样率为 24 000 Hz,共采集了 5 道数据,分别为 E_x 、 E_y 、 H_x 、 H_y 、 H_z ,其中 H_z 为空采数据未接磁探头,用 Matplotlib 库进行可视化显示,选取 02 点 40 分 02 秒至 02 点 40 分 02.1 秒数据段进行

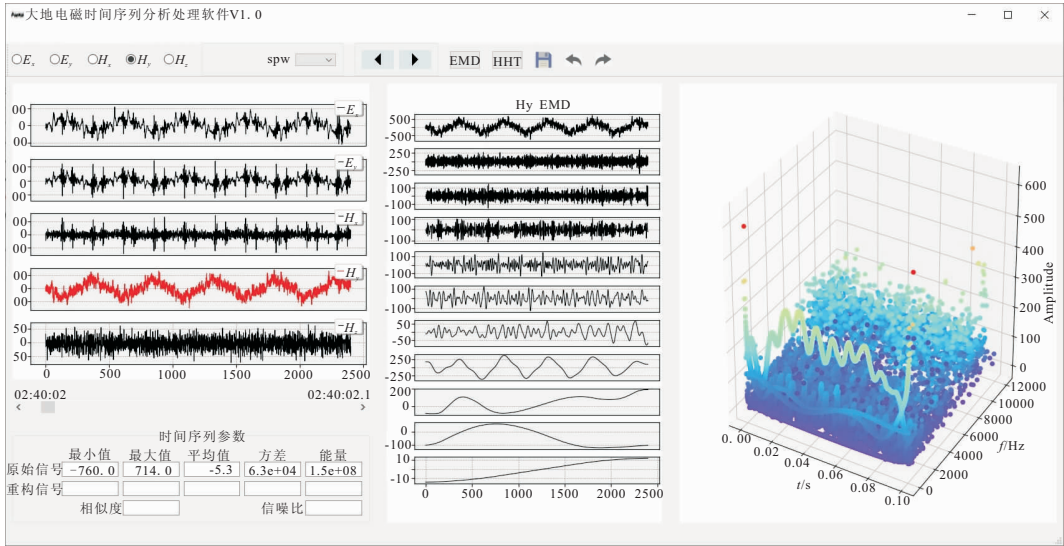


图 5 软件界面

Fig. 5 Software interface

比对(图 2),可知用本文方法读取的数据与原厂 Synchro Time Series View 软件所读取的完全一致,证明了本文方法的正确性和可行性。

在读取时间序列的基础上,将 Python 的 PyEMD 库、SciPy 库初步应用在了时间序列时频分析上,主要采用希尔伯特-黄变换(HHT)进行时频分析,用 PyEMD 库进行经验模态分解(EMD),SciPy 库实现希尔伯特变换^[8],同时选取实测时间序列 4711A12A.TS2 文件 02 点 40 分 02 秒至 02 点 40 分 02.1 秒数据段的 H_y 分量进行时间序列分析,由于钻机施工影响, H_y 数据有明显的低频的干扰,首先进行 EMD 分解得到各阶固有模态函数(IMF),可见 imf7 为主要干扰信号(图 3),然后对各 IMF 进行希尔伯特谱分析,得到希尔伯特瞬时谱,结合所有 IMF 的希尔伯特瞬时谱即可绘制三维时频图(图 4),三维时频图显示钻机产生的干扰信号呈现出连续周期性条带状分布且占据了信号的主要能量,较好地揭示了钻机干扰的时频特征,为下一步处理特别是时间域去噪提供了依据^[9]。

2.4 图像化程序界面

在上述功能的基础上,用 PyQt5 库创建 GUI 应用程序,将不同的功能和显示窗口整合在一起,制作了大地电磁时间序列读写分析的图形界面,增加了时间序列参数显示模块,增强了人机交互功能,满足了数据处理人员分析时间序列时频特征的需要,帮助数据处理人员制定更好的时间序列去噪方法,以提高大地电磁数据信噪比,软件界面见图 5。

3 结论

1)在充分掌握了 MTU 系列大地电磁仪 TSn 时间序列文件存储格式后,利用 Python 的 Numpy 库读写文件,编写了 24 位补码保存的二进制数据转十进制程序,利用 Matplotlib 库进行 TSn 文件可视化显示,利用 SciPy 库和 PyEMD 库实现了时间序列 HHT 时频分析,利用 PyQt5 库将各功能模块整合成了图形界面软件,增强了人机交互功能,验证了 Python 读写及分析处理时间序列的可行性。

2)利用本程序可以快速方便地分析大地电磁时间序列,特别是其 HHT 时频分析功能,可以将时域信号转换到时频域,并通过 Python 的三维可视化功能呈现出来,帮助研究人员更全面地对时间序列信号进行判断,从而选择更加科学的选择去噪手段,以达到提高信噪比的目的。

3)基于本程序已实现的功能,可以开展大地电磁时间序列的进一步处理,比如利用 EMD 进行噪声分离和阈值去噪,利用 VMD 进行噪声分离,开展基于 HHT 阻抗估计的研究,包括利用 Python 强大的机器学习库进行统计学方面的处理研究等^[10-17]。此外,还可以依托 Python 逐渐发展起来的专业大地电磁数据处理库 MTPy 完善 Python 的大地电磁处理功能,充分发挥 Python 开源共享的优势,集众家之所长,进一步拓宽大地电磁处理渠道^[18]。

参考文献:

- [1] LARS KRIEGER, JARED R PEACOCK, J. M-Tpy: A Python toolbox for magnetotellurics[J]. Computers and Geosciences, 2014(72):167-175.
- [2] 雷清, 刘桂梅, 王身龙, 等. 基于 Python 对 MTU 和 V8 系列大地电磁测深仪 TBL 数据研究与应用[J]. 地球物理学进展, 2019, 34(5):2065-2070.
LEI Q, LIU G M, WANG S L, et al. Research and application for raw data of MTU and V8 series magnetotelluric instruments based on Pylhon[J]. Progress in Geophysics, 2019, 34(5):2065-2070. (In Chinese)
- [3] 刘俊峰, 孙宝山, 程云涛. V8 MT 时间序列数据文件读写[J]. 工程地球物理学报, 2015, 12(5):660-664.
LIU J F, SUN B S, CHENG Y T. The Reading and writing of the time series data file of MT in V8[J]. Chinese Journal of Engineering Geophysics, 2015, 12(5):660-664. (In Chinese)
- [4] 仇根根, 方慧, 朱正君, 等. 基于 MATLAB 平台对 V5-2000 大地电磁测深仪 TBL 文件的分析与应用[J]. 物探化探计算技术, 2012, 34(2):178-181.
QIU G G, FANG H, ZHU Z J, et al. MT so - under TBL file based on the MATLAB platform[J]. Computing Techniques for Geophysical and Geochemical Exploration(in chinese), 2012, 34(2):178-181. (In Chinese)
- [5] 柳建新, 刘春明, 马捷, 等. V5-2000 大地电磁测深仪文件头数据格式研究[J]. 物探化探计算技术, 2007, 29(4):359-362.
LIU J X, LIU C M, MA J, et al. The data format of MT sounding instrument V5-2000[J]. Computing Techniques for Geophysical and Geochemical Exploration, 2007, 29(4):359-362. (In Chinese)
- [6] 王丽坤, 陈进超. 基于 V8 系统大地电磁时间序列数据文件读写[J]. 四川地质学报, 2017, 37(4):674-677.
WANG L K, CHENG J C. Read and Writing of V8 Series magnetotelluric time data file[J]. Acta Geologica Sichuan, 2017, 37(4):674-677. (In Chinese)
- [7] PHOENIX COMPANY of CANADA. Data Processing User Guide [M]. Version3. 0. Spokane; Phoenix Company of Canada, 2005.
- [8] 汤井田, 蔡剑华, 任政勇, 等. Hilbert-Huang 变换与大地电磁信号的时频分析[J]. 中南大学学报(自然科学版), 2009, 40(3):1399-1405.
TANG J T, CAI J H, REN Z Y, et al. Hilbert-Huang transform and time-frequency analysis of magnetotelluric signal[J]. Journal of Central South University (Science and Technology), 2009, 40(3):1399-1405. (In Chinese)
- [9] 汤井田, 化希瑞, 曹哲民, 等. Hilbert-Huang 变换与大地电磁噪声压制[J]. 地球物理学报, 2008, 51(2):603-610.
TANG J T, HUA X D, CAO Z M, et al. Hilbert-Huang transform and noise suppression of magnetotelluric sounding data[J]. Chinese Journal of Geophysics, 2008, 51(2):603-610. (In Chinese)
- [10] 蔡建华, 王先春, 胡惟文. 基于经验模态分解与小波阈值的 MT 信号去噪方法[J]. 石油地球物理勘探, 2013, 48(2):303-307.
CAI J H, WANG X C, HU W W. De-noising of MT signal based on empirical mode decomposition and wavelet threshold method[J]. OGP, 2013, 48(2):303-307. (In Chinese)
- [11] 蔡剑华, 汤井田, 王先春. 基于经验模态分解的大地电磁资料人文噪声处理[J]. 中南大学学报(自然科学版), 2011, 42(6):1786-1790.
CAI J H, TANG J T, WANG X C. Human noise elimination for magnetotelluric data based on empirical mode decomposition[J]. Journal of Central South University(Science and Technology), 2011, 42(6):1786-1790. (In Chinese)
- [12] 倪鹏, 邓居智, 刘庆成, 等. 基于经验模态分解(EMD)的小波阈值大地电磁信号去噪方法研究[C]. 第十届中国国际地球电磁学术讨论会, 2011:207-210.
NI P, DENG J Z, LIU Q C, et al. A study of wavelet threshold denoising based on empirical mode decomposition(EMD) for magnetotelluric data[C]. The 10th China International Geo-Electromagnetic Workshop, 2011:207-210. (In Chinese)
- [13] LI J, ZHANG X, TANG J T. Noise suppression for magnetotelluric using variational mode decomposition and detrended fluctuation analysis[J]. Journal of Applied Geophysics, 2020(180):104127, 1-13.
- [14] 李晋, 张贤, 蔡锦. 利用变分模态分解(VMD)和匹配追踪(MP)联合压制音频大地电磁(AMT)强干扰[J]. 地球物理学报, 2019, 62(10):3866-3884.
LI J, ZHANG X, CAI J. Suppression of strong interference for AMT using VMD and MP[J]. Chinese Journal of Geophysics, 2019, 62(10):3866-3884. (In Chinese)
- [15] 胡佃波, 焦磊明, 庞羲. 基于 VMD 的大地电磁信号去噪研究[J]. 能源与环境, 2020, 42(5):72-77.
HU D B, JIAO L M, PANG X. Research on denoising of magnetotelluric signal based on VMD[J]. China Energy and Environmental Protection, 2020, 42(5):72-77. (In Chinese)
- [16] 蔡建华. 基于 Hilbert-Huang 变换的大地电磁信号处

理方法与应用研究[D]. 长沙:中南大学,2010.

CAI J H. Study on processing method of magnetotelluric signal and its application based on Hilbert—Huang transform[D]. Changsha: Central South University, 2010. (In Chinese)

[17] 邓琰, 汤吉. 大地电磁测深方法数据处理进展[J]. 地球物理学进展, 2019, 34(4): 1411—1422.

DENG Y, TANG J. Advance in magnetotelluric data processing[J]. Progress in Geophysics, 2019, 34(4): 1411—1422. (In Chinese)

[18] ALISON KIRKBY, JARED R PEACOCK, JINGMING DUAN & FEI ZHANG. Development of the MTpy software package for magnetotelluric data analysis[J]. ASEG Extended Abstracts, 2018; 1, 1—3.

Reading/writing and application for time series files of MTU series magnetotelluric instruments based on Python

YANG Kai

(Xi'an Center of Mineral Resources Survey, China Geological Survey, Xi'an 710000, China)

Abstract: MTU series magnetotelluric sounding instruments made by Canadian Phoenix Geophysics Company, that are widely used in magnetotelluric sounding, they have many advantages, such as portability and stability, but the unique data structure of their Time Series files can only be read and processed on their supporting software. For example, Synchro Time Series View can only View the waveform of the Time Series and obtain the power spectrum, SSMT2000 can only process data in the frequency domain. In recent years, many time domain processing technologies have become more and more mature. In order to study magnetotelluric signals in the time domain, especially to carry out time domain noise analysis and other work, it is urgent to find a convenient way to read time series files. This paper, because of the many advantages of Python language based on the analysis TSn file storage format of MTU series magnetotelluric sounding instruments, realized the operation of TSn files reading/writing and time—frequency analysis function based on Hilbert—Huang transform by using Python. The author has written graphical interface software, which provides a reference for the further application of Python in the magnetotelluric field.

Keywords: magnetotelluric; time series; python; MTU