文章编号：1001-1749(2023)01-0108-07

基于 Python 的断面解释图目标层埋深数据提取程序开发

王会波[1]，邱崇涛[2,3,4]，张 伟[2,3,4]，牛 禹[2,3,4]，谢明宏[2,3,4]

(1. 中核资源发展有限公司，北京 100013；
2. 核工业航测遥感中心，石家庄 050061；
3. 中核集团 铀资源地球物理勘查中心重点实验室，石家庄 050061；
4. 河北航空探测和遥感技术重点实验室，石家庄 050061)

摘 要： 在电法勘探项目中，目标层埋深图是主要提交成果之一。这里结合反演电阻率推断解释断面图中常见的地质情况，分析总结了不同地质情况下的目标层埋深数据提取方式，并提出了圈定埋深数据边界的算法。利用 Python 语言及其第三方库的强大计算和绘图功能，实现了埋深数据的提取及其边界的圈定，为下一步数据成图提供借鉴。实践结果表明，该程序能有效地结合不同地质情况，借助图形进行交互操作，能快速、准确地提取目标层埋深数据，具有较强的实用性。

关键词： 目标层；Python；推断解释断面图；交互式提取；埋深数据；边界圈定
中图分类号： P 631.3 **文献标志码：** A **DOI：**10.3969/j.issn.1001-1749.2023.01.14

0 引言

在地球物理勘探中，目标层埋深是地质人员最为关注的技术指标之一，而物探推测的目标层埋深图是开展早期钻探工作最直观的图件之一。因此，快速地绘制出目标层埋深图是物探工作非常重要一项任务。目前做法是借助 MapGIS 或其他制图软件(autoCAD)进行等间距手工量取，误差较大，一旦出错，很难发现出错环节。即便充分利用软件中的一些编辑功能或技巧，可以适当提高工作效率，但对于地质环境复杂、解释断面多或者剖面长的情况，仍无法避免工作效率低、精度差等弊端[1]。若期间解释断面图的发生修改，上述工作还需反复。为了能快速、准确地提取断面解释图中目标层的埋深数据，笔者利用 Python 语言开发了一套提取断面图目标层埋深数据程序，通过埋深数据插值与填充、实测点坐标转换、数据合并、边界圈定等操作，最终形成适合数据网格化成图的表格数据(包含坐标数据)。

1 数据提取方式

数据提取包括埋深数据提取和数据边界的圈定两部分内容。

1.1 埋深数据的提取

断面图中的地层界线往往并不连续，主要包括如下几种情况：①断层引起的地层错断；②地层缺失；③超出仪器的探测深度；④岩体出露。从横向上看，在断面图常常会出现缺失现象，而纵向上，则会出现地层界线重叠现象。为了准确性提取深度数

收稿日期：2021-11-17
基金项目：中国核工业地质局基础地质专项(202024－4)
第一作者：王会波(1981－)，男，硕士，高级工程师，研究方向为物探方法及解释研究，E-mail：wanghb@cnhcrm.com。

表 1　不同地质现象的数据提取方式

Tab. 1　Data extraction ways according to different situations

序号	地质情况	提取方式	备注
1	地层错断	1. 对缺失数据进行插值 2. 剔除重叠数据	正、逆断层引起
2	地层缺失	赋空值(NaN)	
3	探测深度限制	赋指定的深度值	深于断面图最大测深值
4	出露地表	赋实测地表高程值	无需修改图件

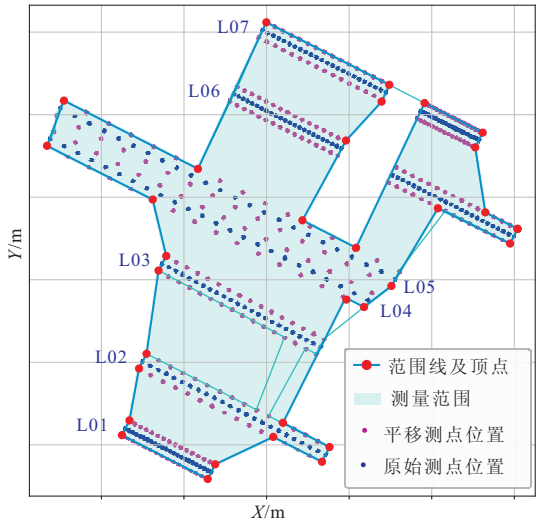

图 1　利用程序圈定目标层边界结果

Fig. 1　Boundary of bed of interest by program

据，根据断面图出现不同现象，选取相应的提取方式(表 1)，进行数值填充或剔除(插值、赋空值或指定值)，从而形成一套连续的目标层埋深数据。

1.2　埋深数据边界的圈定

一个测区大多由数条断面构成，而断面图常出现目标层缺失现象。因此目标层埋深数据范围与实测测区范围存在一定差异。将所有测点视为欧式平面中二维离散点集，使用 alphaShapes(凹包)算法提取其边缘线，为快速网格化成图奠定条件[2]。在物探测量中，常因地形、人文干扰等因素，实测点会发生一定的偏移，为保证圈定出“平整”的数据范围，应将测点(剖面线)拟合为一条直线(图 1)。当然，测点直线拟合也可降低 alpha－shapes 算法的复杂度。

从图 1 可看出，圈定的数据边界总体上较为平整，同时也外扩测点范围；L06 线、L07 线中段因地层缺失出现空白区。实测点(蓝色点)两边进行了适当的外扩。

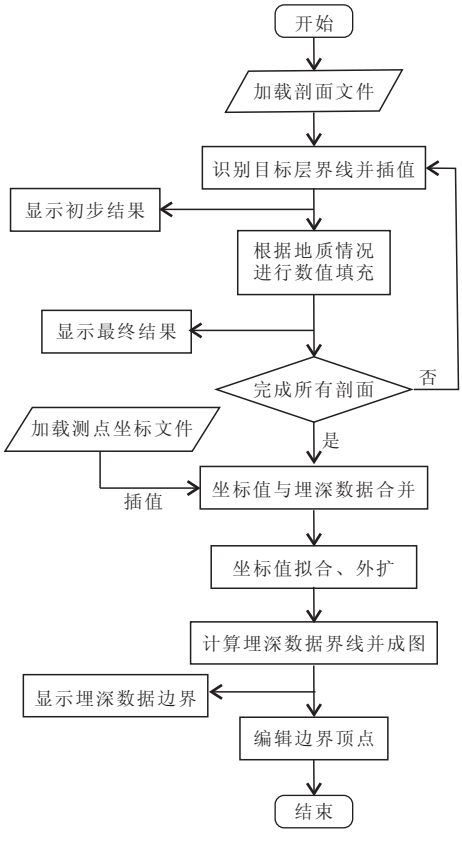

图 2　数据提取程序设计流程

Fig. 2　Flow of program design for data extraction

2　程序设计

2.1　Python 语言简介

Python 是一门简单易学且功能强大的编程语言，拥有高效的高级数据结构，并且能够用简单而又高效的方式进行面向对象编程[3]。加之在地图绘制、地理空间数据的处理与转换方面具有丰富的类库，在地球科学上显示出更明显的优势[4]。程序设计中主要调用了 Pandas(数据清洗)、Numpy(数值计算)、Scipy(数据插值)、Matplotlib(绘图)、Shape-

ly(图形计算)、PyQT5(界面设计)、alphaShape(凹包算法)等第三方库。总之 Pyhton 优雅的语法和极为丰富的第三方库,可大大地缩短程序开发周期。

2.2 程序设计流程

由于勘探剖面较多,为了避免不同测线埋深界面混淆,便于后期数据管理,应以单剖面为提取单位,即每次仅提取一条剖面中的一套埋深数据,再将所有埋深数据合并,加载平面坐标后,最终形成适于网格化成图的一套数据集(图 2)。

1)加载剖面文件后,识别目标层界线。由于推测界线由手工勾勒,节点稀疏且不均匀,并与测站点位置无法对应,因此对界线进行数据插值计算。

2)根据断面中出现各种现象,进行数值填充(缺失部分用空值(NaN)填充)或剔除,使目标层顶埋深界面成为一条不间断的连续曲线。

3)加载实测站点坐标,并对测点坐标插值,插值间距与上述目标层界线相同,然后与所有的剖面数据合并为一个数据集,此时将空值剔除。至此完成了埋深数据提取所有工作。

4)圈定目标层数据范围。利用最终数据集中的 X、Y 坐标值,以各剖面为单位分别进行线性拟合,再制作出相应的平行线,最后利用 alpha Shape 算法圈定目标层数据范围。

2.3 关键技术

2.3.1 数据插值

由于断面图勾勒目标层界线的节点稀疏且不均匀,所以需要进行数据插值。其目的有:①获取足够的数据量,满足网格化需求;②与实测测点坐标匹配,为下一步数据合并提供判定条件。Scipy 库提供了多种一维数据插值(interpolate. interp1d)方法[5-6]。其中线性插值(linear)用于因断层错断引起的数据缺失(前、后相邻目标层界线的尾点和首点之间),主要原因为由断层引起地层错断面较为“平整”,若用三次插值法,可能会产生数据变形,尤其是在断距较大时;而三次插值(cubic)应用于目标层界线,主要是为了保证数据的准确且平滑,利于后期的数据网格化处理。

```
#导入 scipy 库中一维函数
from scipy. interpolate import interp1d
#获得原始数据特征参数
func = interp1d(fn, raw_h, kind='linear')
# func = interp1d(fn, raw_h, kind='cubic')
#生成插值数据(横向),step 为间距
interp_fn = np. arange(min(fn), max(fn), step)
#根据特征参数获取纵向数据
interp_h= func(interp_fn)
```

为保证数据精度,插值步长(step)设为 1 或 2,代表实际距离的 1 m 或 2 m。

2.3.2 圈定目标层边界

依照图 1 所示,目标层边界至少具有 2 个特点:① 包含所有数据点且边界线平整;② 范围略大于实测范围。

为保证界线平整,可使用 Scipy. optimize 库中 curve_fit 函数进行数据拟合,拟合方程为 Y=aX+b。代码如下:

```
def fitting1(x, a, b):
#定于拟合方程
return a * x + b
# x, y 为原始数据集
a1, b1 = optimize. curve_fit(fitting1, x, y)[0]
# x2, y2 为直线拟合后的数据集
x2 = np. linspace(x[0], x[-1], 24, endpoint=True)
y2 = np. round(a1 * x2 + b1, 2)
# 垂向合并数据
xy = np. vstack((x2, y2)). T
```

为实现数据范围外扩,可借助 Shapely 库 parallel_offset 函数。

```
#导入 Shapely 库
from shapely. geometry import LineString
line = LineString(xy) #生成折线
#生成折线的平行线,dis 为偏移距离
# res 为输出精度,left 和 right 标识平行线位置
line_L = line. parallel_offset(dis, 'left', res, 2, 1.0)
line_R = line. parallel_offset(dis, 'rightt', res, 2, 1.0)
```

目标层位置数据整理好,可引用 alphashape 库[7](Kenneth E. Bellock)圈定其边界。

```
#导入 alphashape 库
import alphashape
#获取边界线,pnts 原始坐标点数组
# a_value 为 alpha 系数
range = alphashape. alphashape(pnts, a_value)
```

```
    #Shapely 格式转为列表数据
    range = list(range.exterior.coords)
```

2.3.3　图形显示

在程序运行中，先后提供了 3 种图件，即数据提取结果、数据插值、填充后结果和埋深数据平面范围，用来实时显示运行结果。图形绘制均使用 Matplotlib 库。前两种利用 plot(线)和 scatter(点)函数绘制，用于判定数据正确与否。代码如下：

```
    #绘制地形线
    ax1.plot(x, y, color='g', label=u'地形线')
    #绘制目标层(线)节点
    ax1.scatter(fn, top_h, color='m', label=u'目标层数据')
    #设定 X、Y 轴同比例，防止图形变形
    plt.axis('equal')
    #显示网格，易于数据查看
    plt.grid(True)
```

在圈定埋深数据边界时，虽对测点数据进行直线拟合、平行线制作等预处理，但因 alpha 系数选定问题，难以得到最佳范围。Matplotlib 库中具有事件处理和拾取(Event Handling and Picking)功能，用于多边形(边界)顶点编辑[8,9]。事件包括鼠标拖动(移动测点位置)、delete 键控制(删除点)、insert 键控制(插入点)、窗体关闭(保存编辑结果)。由于篇幅限制，仅列举部分关键代码。

```
    def save_xy(self, event):
    "关闭绘图窗口的回调函数。
    关闭后，自动保存编辑后的坐标数据"
    with open(self.file_name, 'w') as f:
    # 获取边界线的节点坐标
    xy_data = self.poly.get_xydata()
    #遍历 xydata 列表，写入文件
    for x, y in xydata:
    f.write( xy = str(x) + ',' + str(y) + '\n')
    def on_draw(self, event):
    #获取子图的所有信息
    bg = self.canvas.copy_from_bbox (self.ax.bbox)
    #绘图(多边形)
    self.ax.draw_artist(self.poly)
    def on_key_press(self, event):
    # 键盘按钮的回调函数
    if not event.inaxes:
    return
    elif event.key == 'delete':
    #激活'delete'键事件
    ind = self.get_ind_under_point(event)
    #按下'delete'键，删除多边形顶点
    if ind is not None:
    self.poly.xy = np.delete(self.poly.xy,
    ind, axis=0)
    #建立图板
    fig = plt.figure(figsize=(8.4, 8.4))
    #建立子图布的轴(axes)对象
    ax1 = fig.add_subplot(1, 1, 1)
    #转换范围框数据为多边形
    poly = mp.Polygon(myRange, color='c', alpha=0.2,
    label=u'数据边界', animated=True)
    #多边形加载到轴(axes)对象
    ax.add_patch(poly)
    #在图板上建立支持事件(event)画布
    canvas = poly.figure.canvas
    #将绘图事件连接到函数(on_draw)
    canvas.mpl_connect('draw_event', self.on_draw)
    #将键盘事件(按下)连接到函数(on_key_press)
    canvas.mpl_connect('key_press_event', self.on_key_press)
    #将事件(关闭)连接到函数(save_xy)
    canvas.mpl_connect('close_event', self.save_xy)
```

窗体关闭后，自动保存包含数据边界的 X、Y 坐标数据文件，经比例尺换算后，可转为 MapGIS 的 Wal 文件，用于图形裁剪。Surfer 用户可制作 Bln 文件，用于图形白化。

2.3.4　数据读写与清洗

pandas 库功能强大且便捷易用[10]，贯穿整个程序中，用于数据读写与清洗。语句如下：

```
    #根据平距，合并数据(坐标和埋深数据)
    df_all = pd.merge(df_coord, df1, how='outer', on='Fn')
    #剔除空值(NaN)
    df1.dropna(axis=0, how='any', inplace=True)
    #多个文件合并为一个文件，fileSet 文件名列表
```

```
for i, a_file in enumerate(fileSet):
    #读取单文件
    df = pd.read_csv(a_file)
    #文件输出
    if i == 0:
    #保留数据头信息
    df.to_csv(outFile, index=False)
    else:
    #不保留数据头信息,追加模式
    df.to_csv(outFile, index=False, header=None, mode='a+')
```

3 应用实例

以 2019 年－2020 年在松辽北地区进行了可控源音频大地电测测量,项目共完成了 7 条剖面,总测线长度约 4 000 km,目标层为嫩江组(K_2n)。

3.1 数据准备

程序运行前,需要准备的数据有两种:① 实测坐标文件;②解释断面图,格式可为 SVG 格式(由 Surfer 导出)[11] 或者 MapGIS 明码格式(线文件 WAL)。

3.1.1 实测坐标文件

实测坐标文件格式为 CSV(逗号分隔文本文件)。实测坐标数据集须包括 4 项:平距(Dis)、GPS 实测坐标(X、Y、H)(表 2)。

第一行为文件头,注意大小写区分,但排列顺序无任何要求,多余项也不影响程序运行,如测线号(Line)、站点号(Stn)等。所以使用 Pandas 读取数据极为方便。

表 2 坐标文件格式示例

Tab.2 Example of coordinate format in file

Line	Stn	Dis	X	Y	H
L02	M018	0	270020	5321628	218
L02	M018	200	270166	5321547	222
L02	M022	400	270343	5321455	225
L02	M022	600	270522	5321364	228
L02	M026	800	270702	5321271	230
L02	M026	1000	270880	5321179	231
…	…	…	…	…	…

3.1.2 断面图修改

大多情况下,解释断面图由 Surfer 或 MapGIS 软件完成。为了区分与边框、标识线及其他地质界线,将需目标层设定为特定色(图 3),如绿色(Green)或品色(Magenta),其他属性无需改动。

3.2 交互操作

3.2.1 埋深数据的提取

在数据提取过程中,将先弹出目标层埋深数据原始示意图(图 4),结合实际地质情况和测线布置,获取最终正确的埋深数据(图 5)。

从图 5 中可看出,在平距 8 100 m～8 500 m、19 800 m～19 900 m 间,因断层(fault)引起的“空缺”将通过数据插值补齐;平距 33 500 m～42 900 m、58 900 m～63 800 m 间为地层缺失(lack),赋予空值(NaN)。

假若在平距 33 500 m～42 900 m 间为目标层出露,此段数据将赋予地表高程值;58 900 m～63 800 m 间埋深较大而未探测到,此段数据将赋予指定埋深值(大于断面图最深值),结果如图 6 所示。

3.2.2 数据边界编辑

数据文件合并后,程序弹出目标层数据边界示意图(图 7)。图形中的边界顶点可编辑,参考站点位置,调整边界范围,结果见图 1。

3.3 输出结果

输出结果包括,目标层埋深数据及其边界,网格化成图后如图 8 所示。

4 结论

从反演解释断面图中提取地层埋深数据是绘制目标层顶(底)埋深等值线平面图的基础。在提取过程中,充分考虑了断面图中出现的各类地质现象,采用相应的数据提取方式,确保了数据的准确性。利用 Python 语言开发此类程序具有如下优点。

1)利用 Matplotlib 绘图功能,不仅可实时显示运算结果,而且还可利用事件处理和拾取功能编辑、修改图元并保存其最终结果。

2)利用 Scipy 进行插值方法,提高了数据提取精度。

3)充分利用 Pandas、Shapely、alphaShapes 等第三方库,在数据清理、图形计算等方面的强大功能,简化了代码,使程序更易于维护。

总之该程序结合地质情况利用图形界面进行交互式操作,确保了数据的准确性,为快速提取目标

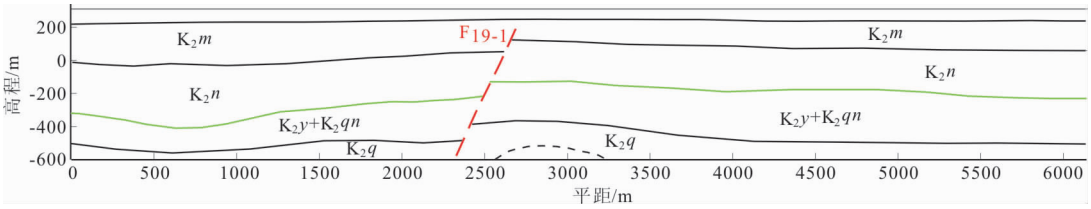

图 3　断面图修改后的效果(局部)

Fig. 3　Effect of section modified (part)

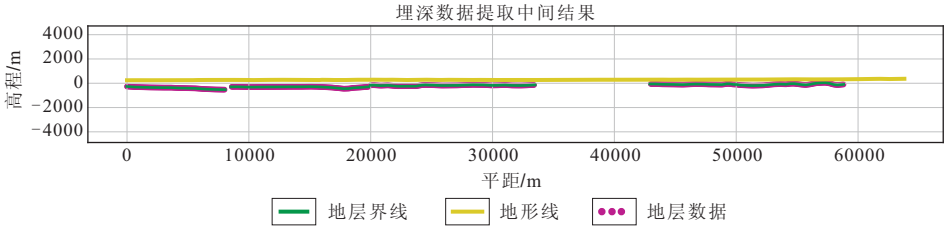

图 4　数据提取初步结果示意图

Fig. 4　Chart of preliminary results of data extraction

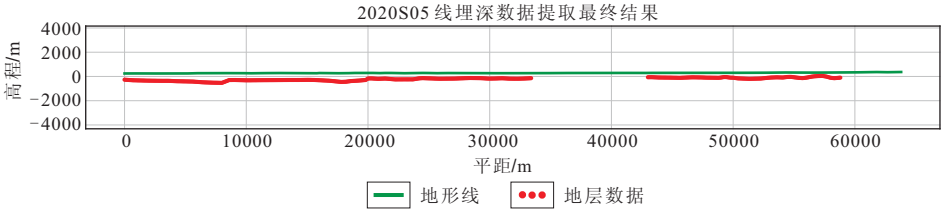

图 5　数据提取最终结果示意图(1)

Fig. 5　Chart of final results of data extraction (1)

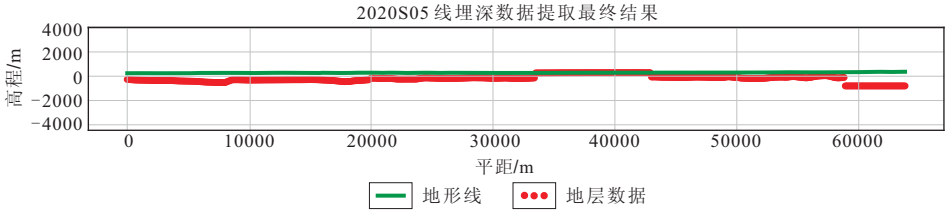

图 6　数据提取最终结果示意图(2)

Fig. 6　Chart of final results of data extraction (2)

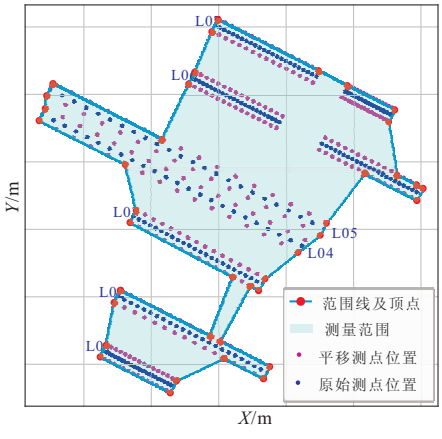

图 7　目标层边界圈定示意图(原始)

Fig. 7　Chart of extend of interest bed (origin)

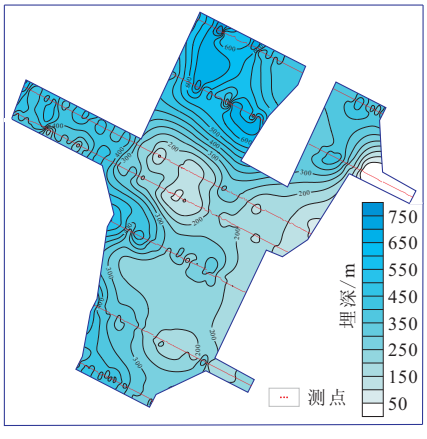

图 8　目标层底板埋深图

Fig. 8　Map of buried depth of interest bed

层埋深数据奠定了基础，而 Python 简洁的语言及其丰富的第三方库，有效优化程序设计方案，极大地缩短了开发周期。

参考文献：

[1] 张勇. 带地形等值线断面图绘制简法[J]. 资源环境与工程，2015，29 (4)：515－518.
ZHANG Y. A simple method of rendering terrain contour profile [J]. Resource Environment & Engineering，2015，29 (4)：515－518. (In Chinese)

[2] 崔年生. 基于 2D Alpha Shapes 自动生成露天矿台阶爆破边界[J]. 采矿技术，2019，19(2)：138－141.
CUI N S. Automatic generation blasting boundary of bench at strip mine based on 2D Alpha Shapes algorithm [J]. Mining Technology，2019，19(2)：138－141. (In Chinese)

[3] 周永章，张良均，张奥多，等. 地球科学大数据挖掘与机器学习[M]. 广州：中山大学出版社，2018.
ZHOU Y Z，ZHANG L J，ZHANG A D，et al. Big data mining & machine learning in geoscience [M]. Guangzhou：SUN YAT－SEN University Press，2018. (In Chinese)

[4] BILL LUBANOVIC. Introducing Python [M]. Sebastopol：O'Reilly Media，Inc，2014.

[5] FRANCISCO J. Blanco－Silva. Mastering SciPy[M]. Birmingham. Packt Publishing Ltd，2015.

[6] The SciPy Community. SciPy Reference Guide (Release 1. 7. 1) [EB/OL]. https://docs. scipy. org/doc/. 2021.

[7] KENNETH E. BELLOCK. Alpha Shape Toolbox [EB/OL]. https://github. com/ bellockk/alphashape，2021.

[8] The matplotlib development team. Matplotlib (Release 3. 4. 3) [EB/OL]. https://matplotlib. org/stable/gallery/index. html/Event handling. 2021.

[9] BENJAMIN V. Root. Interactive applications using matplotlib [M]. Birmingham. Packt Publishing Ltd，2015.

[10] WES MCKINNEY. Python for Data Analysis [M]. Sebastopol：O'Reilly Media，Inc，2018.

[11] 杨金政，邱崇涛，陈鹏. 基于 SVG 格式进行 MapGis 图形转换[J]. 物探与化探，2018，42(5)：1069－1075.
YANG J Z，QIU C T，CHEN P. The graphics conversion of mapGis format based on SVG[J]. Geophysical and Geochemical Exploration，2018，42(5)：1069－1075. (In Chinese)

Programming development of buried depth data extracted of interest bed from interpretational sections based on Python

WANG Huibo[1]，QIU Chongtao[2,3,4]，ZHANG Wei[2,3,4]，NIU Yu[2,3,4]，XIE Minghong[2,3,4]

(1. China Nuclear Resources Development Company Limited，Beijing 100013，China；
2. Airbrone Survey and remote Sensing Center of Nuclear Industry，Shijiazhuang 050061，China；
3. CNNC Key Laboratory for Geophysical Exploration Technology Center of Uranium Resources，Shijiazhuang 050061，China；
4. Hebei Key Laboratory of Airborne Survey and remote Sensing Technology，Shijiazhuang 050061，China)

Abstract：In the electromagnetic prospecting survey，the buried depth of interest bed is one of the main results submitted. Combined with the common geologic situations that appeared in interpretational sections of inverse resistivity，the extracting ways of buried depth data of interest beds under the different geologic situations are analyzed and summarized. Meanwhile，the algorithm of the boundary delineated buried depth. The buried depth data of the interested bed and its boundary is extracted or delineated by robust computation and drawing of Python and its third－party libraries，which provide a base for further data mapping. The case shows that based on the different geologic situations，this programming can extract the correlation data rapidly and precisely by the interactive operation according to the drawings，which has stronger practicability and is valuable.

Keywords：interest bed；Python；interpretational section；interactive extraction；buried depth data；boundary delineation