LoopStructural的低层级数据输入准备

li

构造地质模型

描述地质对象的3D几何特征是理解和模拟地质过程的基础。地质模型是由界面(interface, surface)边界约束和由断层(surface)切断的一套formations (volumes)。断层不一定约束formation。

3D制图学的一般目标是：（1）几何正确：是由已知的几何特征拟合的；（2）拓扑上一致：不同地质对象的组成之间的关系得到正确反映；（3）地质上是真实的。

当数据点足够且地质界面相对简单时，经典的地质统计学方法是有效的。而数据稀疏和地质体复杂时，需要为各种地质类型发展特殊的方法。一些插值方法，如Discrete Smooth Interpolation, Bezier surface已应用于几何表面模拟。

Lajaunie C., Courrioux G., Manuel L.1997. Foliation fields and 3D cartography in geology: Principles of a method based on potential interpolation, Math. Geol. 29: 571-584.

构造地质模型

Lajaunie, et al. (1997)建立了一种方法：当已知一个或多个界面上的点，以及可获取额外的平面方向(plane orientation)数据。这些方向数据不必属于某个界面，但假设采样来自地质构造(sedimentary plane, foliation, cleavage plane)。地质建模问题就是：构造穿过各界面上已知点的面(surface)，并且与方向数据兼容。

适合条件

（1）假设待模拟的面属于一组近似符合叶理构造(foliation)场的平行面。

（2）假设一些方向数据可以转换为矢量数据。因此，从地质学角度看，必须已知某些位置上构造的polarity。

构造地质模型

基本原理

隐式地质建模的基本原理基于以下思路：空间上定义的标量场，其梯度与方向数据正交。

插值标量场：

(1) 界面的一些点有与另一些点相同的，但未知的标量值；

(2) **相同界面上的另一些点**：已知标量场的梯度，或已知与该梯度正交的方向（角度）。

最终，模型面由插值场的等值面表征。

构造地质模型

隐式建模的优势是：在一些插值算法中，联合使用不同类型的独立的地质信息，包括：界面上的已知点、其他点上的方向数据以及梯度数据。该方法的另一个特性是：可以模拟一组面，同时考虑不同界面上所有的已知点。

缺点是：模型需要是规则的(regularity)。奇异性(singularity)，比如不可微分的断层（不连续体）需要做特殊的修正；还需要做模型（不确定性）评估。

fault_trace.shp

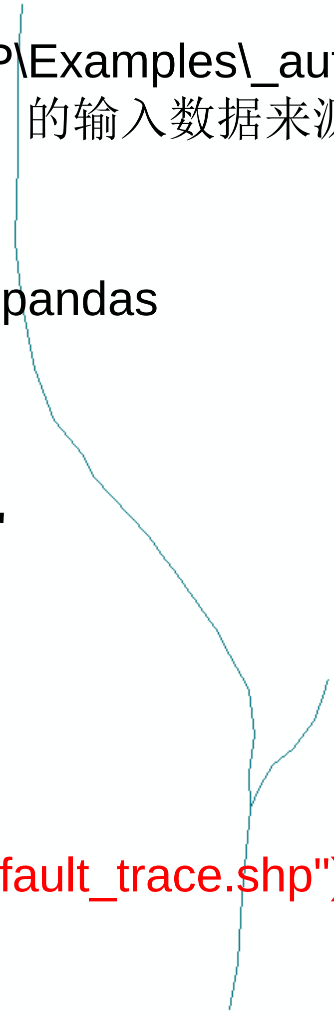
E:\Reservoir_modelling\GeoModelling\LOOP\Examples_auto_examples_python\3_fault\ fault_network.py 的输入数据来源

```
def load_fault_trace():
 """Load the fault trace dataset, requires geopandas
Returns
-------
GeoDataFrame
   dataframe of a shapefile for two faults    """

 import geopandas
 module_path = dirname(__file__)

fault_trace = geopandas.read_file(
    join(module_path, Path("data/fault_trace/fault_trace.shp"))
 )

  return fault_trace
```




Table

fault_trace

FID	Shape	id	fault_name
0	Polyline	0	fault_1
1	Polyline	0	fault_2

1 (0 out of 2 Selected)

fault_trace

fault_trace.shp

构建断层的地质模型，关于断层的最低标准的输入要求包括：

断层的XYZ坐标、断层名称(name)、断层连接关系(名称对)、断层位移

geological_map_data （读取最底层的地质数据）

- bbox.csv
- contacts.csv
- fault_displacement.csv
- fault_edges.txt
- fault_locations.csv
- fault_orientations.csv
- stratigraphic_order.csv
- stratigraphic_orientations.csv
- stratigraphic_thickness.csv

There is a disconnect between the input data required by 3D modelling software and a geological map. In LoopStructural the geological model is a collection of implicit functions that can be mapped to the distribution of stratigraphic units and the location of fault surfaces. Each implicit function is approximated from the observations of the stratigraphy, this requires grouping conformable geological units together as a single implicit function, mapping the different stratigraphic horizons to a value of the implicit function and determining the relationship with geological structures such as faults.

总结，LoopStructural输入就是要准备：

•stratigraphic contacts（地层接触）

•stratigraphic orientations（地层方向）

•stratigraphic thickness（地层厚度）

•stratigraphic order（地层顺序）

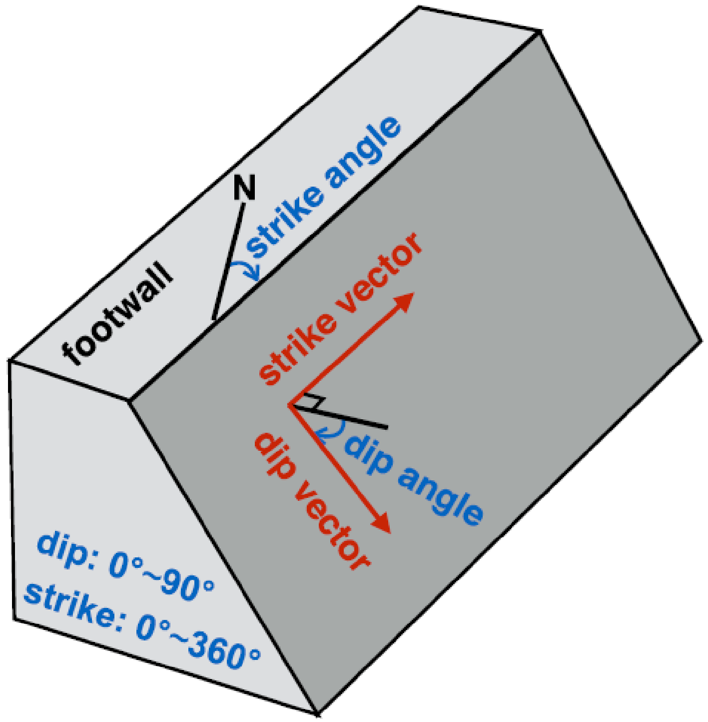
为构建地层平面模型，需要：

•fault locations（断层位置）

•fault orientations（断层方向）

•fault properties（断层属性）

•fault edges（断层边界）



```
################################
#    Read stratigraphy from csv
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
(
contacts,
stratigraphic_orientations,
stratigraphic_thickness,
stratigraphic_order,
bbox,
fault_locations,
fault_orientations,
fault_properties,
fault_edges,)
= load_geological_map_data()              # 见：_base.py

thicknesses = dict(
        zip(
                list(stratigraphic_thickness["name"]),
                list(stratigraphic_thickness["thickness"]),
        )
)
```

通过应用 dict() 函数和 zip() 函数，可将两个列表List转换为对应的字典。

_base.py

```
def load_claudius():
    """Model dataset sampled from 3D seismic data

def load_noddy_single_fold():
    """Model dataset for plunging cylindrical fold

def load_intrusion():
    """Model dataset for a faulted intrusion

def load_unconformity():
    """Model dataset sampled for a model containing an unconformity

def load_geological_map_data():
    """An example dataset to use the processinput data class

def load_fault_trace():
    """Load the fault trace dataset, requires geopandas
```

读取shapefile文件

```
def load_geological_map_data():

 module_path = dirname(__file__)
 contacts = pd.read_csv(join(module_path,
          Path("data/geological_map_data/contacts.csv"))    )
```

bbox.csv | _base.py | contacts.csv

```
X,Y,Z,name
521588.3817254504,7500713.701949331,596.2389787218352,Mount_McRae_Shale_and_Mount_Sylvia_Formation
520790.42998466,7500582.96937573,592.253634860429,Mount_McRae_Shale_and_Mount_Sylvia_Formation
519713.53061227,7500191.47044685,548.481424280350 4,Mount_McRae_Shale_and_Mount_Sylvia_Formation
527989.1321683651,7501093.758504495,546.0052443577555,Mount_McRae_Shale_and_Mount_Sylvia_Formation
526591.88981018,7500682.03115599,545.280018762287 9,Mount_McRae_Shale_and_Mount_Sylvia_Formation
525164.74149011,7500478.01038839,555.861793515468 9,Mount_McRae_Shale_and_Mount_Sylvia_Formation
522249.9582507248,7500072.981502291,542.2788204650827,Wittenoom_Formation
521177.75841614,7500024.97138615,529.7766110744667,Wittenoom_Formation
519637.3397091,7499435.95032536,515.4533338163503,Wittenoom_Formation
524500.3112717137,7500180.372155858,543.2555005725637,Marra_Mamba_Iron_Formation
522960.03946771007,7499706.49144167,556.3080980307724,Marra_Mamba_Iron_Formation
522503.1624246932,7499751.781237598,552.6455814320934,Marra_Mamba_Iron_Formation
522476.74840981024,7499754.399634302,552.7089652819747,Marra_Mamba_Iron_Formation
522349.8299966399,7499766.98093844,552.5575026437795,Marra_Mamba_Iron_Formation
521446.58837318,7499727.47903245,544.7599128787926,Marra_Mamba_Iron_Formation
```

接触面？ 岩石地层单元？ Formation/Group

def load_geological_map_data():

stratigraphic_orientations = pd.read_csv(join(module_path, Path("data/geological_map_data/stratigraphic_orientations.csv")))

fault_orientations.csv

```
,X,Y,Z,gx,gy,gz,coord,feature_name
0,541963.19163038,7493283.13978594,674.3337120305703,0.5637102340673046,0.7000439341002919,0.43837114678907735,0,Fault_2997
1,545755.3624073934,7489723.89,674.3337120305703,0.5637102340673046,0.7000439341002919,0.43837114678907735,0,Fault_2997
2,534479.09627295,7498804.10098736,518.9818313655426,0.5637102340673046,0.7000439341002919,0.43837114678907735,0,Fault_2997
3,525794.77713101,7496712.51283815,473.15194527242295,-0.8407355773657558,-0.5312602839648115,0.10452846326765346,0,Fault_3496
4,530012.248380578,7489723.89,473.15194527242295,-0.8407355773657558,-0.5312602839648115,0.10452846326765346,0,Fault_3496
5,522189.26567998,7502103.99834995,547.9009647774983,-0.8407355773657558,-0.5312602839648115,0.10452846326765346,0,Fault_3496
6,532914.74969844,7501129.60123437,596.9398929098236,0.7158482261571691,0.5305223305892705,0.45399049973954686,0,Fault_3498
7,539513.18738029,7491597.13566214,596.9398929098236,0.7158482261571691,0.5305223305892705,0.45399049973954686,0,Fault_3498
8,530175.96563729,7504196.10320143,574.9156341302787,0.7158482261571691,0.5305223305892705,0.45399049973954686,0,Fault_3498
9,534321.30492983,7491059.07179509,558.9449048693546,0.23786642062354135,0.9523715100858164,0.19080899537654492,0,Fault_7439
10,540588.0629057533,7489723.89,558.9449048693546,0.23786642062354135,0.9523715100858164,0.19080899537654492,0,Fault_7439
11,527475.04337437,7492999.02684152,493.1021139078194,0.23786642062354135,0.9523715100858164,0.19080899537654492,0,Fault_7439
12,524494.08762163,7496605.00666576,535.0082514443101,0.8284417463508296,0.52005480646225,0.20791169081775948,0,Fault_12647
13,520343.17884566,7501712.97896174,535.0082514443101,0.8284417463508296,0.52005480646225,0.20791169081775948,0,Fault_12647
14,527869.336680077,7489723.89,448.2019668469705,0.8284417463508296,0.52005480646225,0.20791169081775948,0,Fault_12647
15,548978.50085605,7508211.38333105,627.0135395186778,0.804019186643728,0.32179674875417574,0.5000000000000001,0,Fault_12658
16,546639.32683299,7512670.78252025,627.0135395186778,0.804019186643728,0.32179674875417574,0.5000000000000001,0,Fault_12658
17,551978.745,7499330.080140844,536.5018389409632,0.804019186643728,0.32179674875417574,0.5000000000000001,0,Fault_12658
```

地层层理参数？

```
def load_geological_map_data():
    stratigraphic_thickness = pd.read_csv(        join(module_path,
            Path("data/geological_map_data/stratigraphic_thickness.csv")),
            skiprows=1,
            names=["name", "thickness"],     )
```

_base.py | stratigraphic_thickness.csv

```
,thickness
Mount_McRae_Shale_and_Mount_Sylvia_Formation,224.5
Marra_Mamba_Iron_Formation,152.0
Boolgeeda_Iron_Formation,166.5
Woongarra_Rhyolite,389.0
Jeerinah_Formation,600.0
Brockman_Iron_Formation,557.0
Wittenoom_Formation,236.0
Weeli_Wolli_Formation,241.5
Turee_Creek_Group,162.0
Fortescue_Group,236.0
Bunjinah_Formation,236.0
Pyradie_Formation,236.0
```

地层厚度

```
def load_geological_map_data():

    stratigraphic_order = pd.read_csv(        join(module_path,
            Path("data/geological_map_data/stratigraphic_order.csv")),
            skiprows=1,
            names=["order", "unit name"],     )
```

_base.py | stratigraphic_order.csv

```
group,index in group,unit name
0,0,Turee_Creek_Group
0,1,Boolgeeda_Iron_Formation
0,2,Woongarra_Rhyolite
0,3,Weeli_Wolli_Formation
0,4,Brockman_Iron_Formation
0,5,Mount_McRae_Shale_and_Mount_Sylvia_Formation
0,6,Wittenoom_Formation
0,7,Marra_Mamba_Iron_Formation
0,8,Jeerinah_Formation
0,9,Fortescue_Group
0,10,Bunjinah_Formation
0,11,Pyradie_Formation
```

地层group，及组内的formation编号及单元名称

def load_geological_map_data():

```
bbox = pd.read_csv(        join(module_path,
            Path("data/geological_map_data/bbox.csv")),
            index_col=0,
            header=None,
            names=["X", "Y", "Z"],    )
```

bbox.csv   _base.py

```
origin, 519572.569,7489723.89,-4800.0
maximum,551978.745,7516341.01,1200.0
```

纵向坐标的正负号表示什么？
地层纵坐标可以是正的？
负数表示深度？

def load_geological_map_data():

```
    fault_properties = pd.read_csv(        join(module_path,
            Path("data/geological_map_data/fault_displacement.csv")),
            index_col=0,     )
```

base.py   fault_displacement.csv

```
Fault,displacement
Fault_2997,90.48575714705343
Fault_3496,84.0
Fault_3498,98.01101917761947
Fault_7439,84.4739703254439
Fault_12647,88.47256501865093
Fault_12658,34.0
```

断层的位移？

```
def load_geological_map_data():
    fault_edges = []    with open(        join(module_path,
            Path("data/geological_map_data/fault_edges.txt")), "r" ) as f:
            for l in f.read().split("\n"):
                    faults = l.split(",")
                     if len(faults) == 2:
                            fault_edges.append((faults[0], faults[1]))
```

ault_edges.txt

```
Fault_7439, Fault_3496
Fault_2997, Fault_3498
```

断层边界？

```
def load_geological_map_data():

    fault_locations = pd.read_csv(     join(module_path,
            Path("data/geological_map_data/fault_locations.csv"))    )
```

ault_edges.txt | fault_locations.csv

```
,X,Y,Z,val,feature_name,coord
0,545755.3624073934,7489723.89,674.3337120305703,0,Fault_2997,0
1,547430.9534941226,7491804.482069679,-4799.0,0,Fault_2997,0
2,544315.01813374,7491692.17202982,608.8423420181291,0,Fault_2997,0
3,545970.5755012535,7493747.885211789,-4799.0,0,Fault_2997,0
4,543212.94172121,7492633.64954097,540.9191728101371,0,Fault_2997,0
5,544847.7214855079,7494663.560042382,-4799.0,0,Fault_2997,0
6,541963.19163038,7493283.13978594,549.9515878490478,0,Fault_2997,0
7,543600.7343979055,7495316.481524772,-4799.0,0,Fault_2997,0
8,540638.12432243,7494130.62800413,502.72256808569006,0,Fault_2997,0
9,542261.2197983414,7496146.028364134,-4799.0,0,Fault_2997,0
10,538598.44607519,7497133.1117222,484.3026384285973,0,Fault_2997,0
11,540215.9069201774,7499141.514711994,-4799.0,0,Fault_2997,0
12,535889.29443311,7498465.10473309,494.1987996506309,0,Fault_2997,0
13,537509.7825000841,7500477.267080548,-4799.0,0,Fault_2997,0
14,534479.09627295,7498804.10098736,518.9818313655426,0,Fault_2997,0
15,536107.1654349385,7500825.677922771,-4799.0,0,Fault_2997,0
```

……

最后1列都是0

```
def load_geological_map_data():

    fault_orientations = pd.read_csv(       join(module_path,
            Path("data/geological_map_data/fault_orientations.csv"))
    )
```

ult_edges.txt stratigraphic_orientations.csv

```
X,Y,Z,azimuth,dip,polarity,formation,source
535257.61161618,7499029.61466975,516.933710639043,360.0,40.0,1,Jeerinah_Formation,observed
548279.32061205,7493304.27929258,547.2018422432042,190.0,50.0,1,Jeerinah_Formation,observed
548279.32061205,7493304.27929258,547.2018422432042,190.0,55.0,1,Jeerinah_Formation,observed
541013.16039186,7493387.43930108,540.2179723136178,150.0,28.0,1,Bunjinah_Formation,observed
536742.23216759,7490698.33941677,500.982204988715,110.0,28.0,1,Bunjinah_Formation,observed
548249.15588261,7493512.48955459,539.491956095125,190.0,50.0,1,Jeerinah_Formation,observed
548249.15588261,7493512.48955459,539.491956095125,190.0,55.0,1,Jeerinah_Formation,observed
523989.54366251,7500823.54711461,634.4154005612631,355.0,63.0,1,Mount_McRae_Shale_and_Mount
537199.09563866,7490607.57099584,515.1002674574747,110.0,28.0,1,Jeerinah_Formation,observed
543234.97739865,7505326.76715363,602.3721861160096,330.0,15.0,1,Jeerinah_Formation,observed
543754.18068029,7504599.12601575,564.4080619548795,330.0,15.0,1,Jeerinah_Formation,observed
522441.04068434,7500496.57253559,585.065125951248,163.0,29.0,1,Mount_McRae_Shale_and_Mount_
521969.86148165,7509447.16831663,573.6155658100872,365.0,52.0,1,Brockman_Iron_Formation,obs
535856.33026169,7512629.90191351,652.0562424889541,383.0,32.0,1,Brockman_Iron_Formation,obs
538825.16687886,7511863.37365568,670.993051781565,371.0,9.0,1,Brockman_Iron_Formation,obse
```

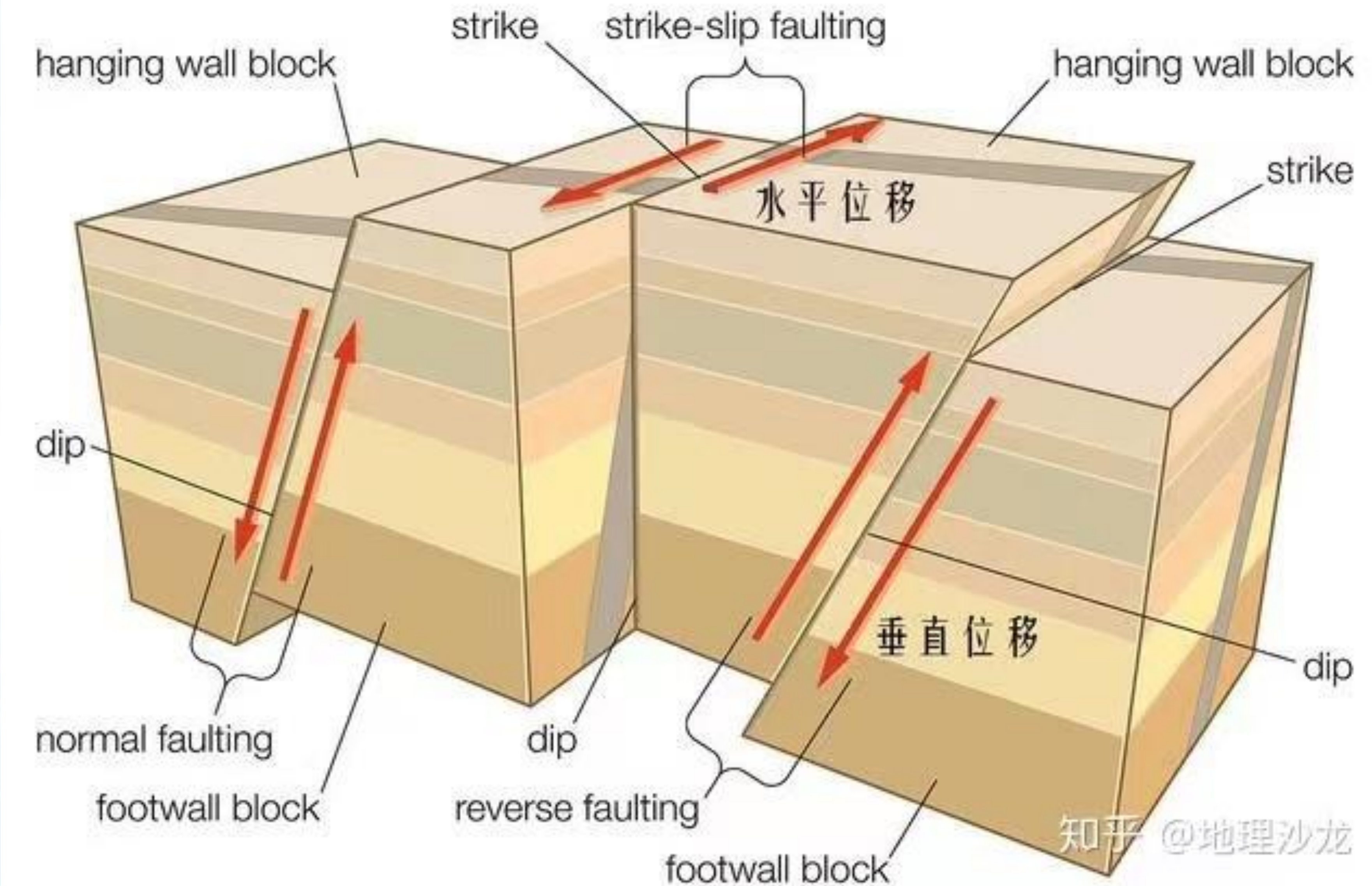
……

断层的方向？

上面的底层的断层参数和地层参数，map2loop可以基于ArcGIS的shapefile文件的自动解译（降低大量的工作耗时, days → hours），但是使用的是澳大利亚有公开的数字地质图。

用于中国区域的地质建模，需要研究李晨阳(2019)的公开数据库的内容。

李晨阳，王新春，何春珍，吴轩，孔昭煜，李晓蕾. 2019. 全国1 : 200 000数字地质图（公开版）空间数据库. 中国地质，46(S1): 1-10.


strike
strike-slip faulting
hanging wall block
hanging wall block
水平位移
strike
dip
垂直位移
dip
normal faulting
dip
footwall block
reverse faulting
footwall block
知乎 @地理沙龙

Petrel解译的断层数据文件输出格式

Charisma 2D interpretation lines (ASCII) (*.*)
Charisma 2D interpretation lines (ASCII) (*.*)
Charisma 3D interpretation lines (ASCII) (*.*)
Charisma fault sticks (ASCII) (*.*)
Charisma Irap 2D interpretation lines (ASCII) (*.*)
IESX 2D interpretation lines (ASCII) (*.*)
IESX 3D interpretation lines (ASCII) (*.*)
IESX fault polygons (ASCII) (*.*)
IESX fault sticks (ASCII) (*.*)
Kingdom 2D interpretation lines (ASCII) (*.*)
Kingdom 3D interpretation lines (ASCII) (*.*)
Kingdom fault sticks (as charisma) (ASCII) (*.*)
Seisworks 3D interpretation (ASCII) (*.*)

第八列是点号，代表一条连续解释的线条，就是一个INLINE剖面里解释了多条断层线 每解释一条就会有一个数字编号 表示一条连续解释的断层线

XYZ坐标、断层的名称(Feature_name)

INLINE-	2147483647	2147483647	15485312.14200	4888034.52000	6345.79500	epthTDQ_2021_Ht_J_F17	75
INLINE-	2147483647	2147483647	15485312.13300	4887875.56300	6416.55500	epthTDQ_2021_Ht_J_F17	75
INLINE-	2147483647	2147483647	15485312.12800	4887789.97100	6479.22200	epthTDQ_2021_Ht_J_F17	75
INLINE-	2147483647	2147483647	15485312.11200	4887508.74000	6822.38900	epthTDQ_2021_Ht_J_F17	75
INLINE-	2147483647	2147483647	15485312.10400	4887374.23800	6979.37200	epthTDQ_2021_Ht_J_F17	75
INLINE-	2147483647	2147483647	15481712.28000	4889251.99900	5777.89800	epthTDQ_2021_Ht_J_F17	76
INLINE-	2147483647	2147483647	15481249.95400	4889251.99900	6017.38700	epthTDQ_2021_Ht_J_F17	76
INLINE-	2147483647	2147483647	15480492.64200	4889251.99900	6257.73700	epthTDQ_2021_Ht_J_F17	76
INLINE-	2147483647	2147483647	15479116.53700	4889251.99900	6634.40200	epthTDQ_2021_Ht_J_F17	76
INLINE-	2147483647	2147483647	15478679.19000	4889251.99900	6815.54600	epthTDQ_2021_Ht_J_F17	76
INLINE-	2147483647	2147483647	15485223.72200	4888051.99500	6370.69900	epthTDQ_2021_Ht_J_F17	77
INLINE-	2147483647	2147483647	15485084.50800	4888051.99500	6428.52800	epthTDQ_2021_Ht_J_F17	77
INLINE-	2147483647	2147483647	15484912.15000	4888051.99500	6485.13500	epthTDQ_2021_Ht_J_F17	77
INLINE-	2147483647	2147483647	15484666.15100	4888051.99500	6496.49500	epthTDQ_2021_Ht_J_F17	77
INLINE-	2147483647	2147483647	15484250.10400	4888051.99500	6489.51300	epthTDQ_2021_Ht_J_F17	77
INLINE-	2147483647	2147483647	15483712.17400	4888051.99500	6491.27700	epthTDQ_2021_Ht_J_F17	77
INLINE-	2147483647	2147483647	15483107.75200	4888051.99500	6541.45600	epthTDQ_2021_Ht_J_F17	77
INLINE-	2147483647	2147483647	15482615.97600	4888051.99500	6622.42200	epthTDQ_2021_Ht_J_F17	77
INLINE-	2147483647	2147483647	15482224.04500	4888051.99500	6657.07200	epthTDQ_2021_Ht_J_F17	77
INLINE-	2147483647	2147483647	15482112.20400	4888051.99500	6700.85900	epthTDQ_2021_Ht_J_F17	77
INLINE-	2147483647	2147483647	15481965.39900	4888051.99500	6766.34900	epthTDQ_2021_Ht_J_F17	77
INLINE-	2147483647	2147483647	15481811.15600	4888051.99500	6934.13300	epthTDQ_2021_Ht_J_F17	77
INLINE-	2147483647	2147483647	15481749.86100	4888051.99500	6992.25000	epthTDQ_2021_Ht_J_F17	77
INLINE-	2147483647	2147483647	15481658.66300	4888051.99500	7053.45900	epthTDQ_2021_Ht_J_F17	77