pydEXP Documentation

Release v0.1

Benjamin Mary

Nov 12, 2022

PYDEXP

pyDEXP is a open-source python package aiming at processing potential field data using the dEXP theory formulated by Fedi et al., 2012. The package largely benefits from the imaging methods module of the Fatiando a terra open-source python library.

References

- Uieda, L., V. C. Oliveira Jr, and V. C. F. Barbosa (2013), Modeling the Earth with Fatiando a Terra, Proceedings of the 12th Python in Science Conference, pp. 91 - 98.
- Fedi, M., and M. Pilkington (2012), Understanding imaging methods for potential field data, Geophysics, 77(1), G13, doi:10.1190/geo2011-0078.1

CHAPTER

ONE

QUICK USE

Clone the gitlab repository:

```
git clone https://github.com/BenjMy/dEXP_imaging.git
```

From the same *dEXP_imaging* directory you can import the module from source using python.

CHAPTER

TWO

CONTRIBUTE AND SUPPORT

- Source Code: https://github.com/BenjMy/dEXP_imaging
- Issue Tracker: https://github.com/BenjMy/dEXP_imaging/issues

CHAPTER

THREE

CITING PYDEXP

If you use the pydEXP code for you work, please cite this paper as:

Paper in preparation

3.1 Getting Started

pyDEXP aims to process Mise-à-la-masse (MALM) datasets for a variety of applications. pyDEXP has been initially developed for plant root imaging.

The simpliest processing can be achieved with the python API. You'll first need to import the pyDEXP package:

```
import lib.dEXP as dEXP
import lib.plot_dEXP as pEXP
```

Then after loading you data make basics operation such as upward continuation (Uieda et al., 2013, 2018):

```
mesh, label_prop = dEXP.upwc(xp, yp, zp, U, shape,
                 zmin=0, zmax=max_elevation, nlayers=nlay,
                 qorder=qorder)
```

Searching for ridges (Fedi et al., 2012):

```
dfI,dfII, dfIII = dEXP.ridges_minmax(xp, yp, mesh, p1, p2,
                                     label=label_prop,
                                     fix_peak_nb=2,
```

and finally plot the results:

```
fig = plt.figure()
ax = plt.gca()
pEXP.plot_xy(mesh, label=label_prop, ax=ax)
pEXP.plot_ridges_harmonic(dfI,dfII,dfIII,ax=ax)
```

More examples are available in the Example gallery section.

References

Uieda, L., V. C. Oliveira Jr, and V. C. F. Barbosa (2013), Modeling the Earth with Fatiando a Terra, Proceedings of the 12th Python in Science Conference, pp. 91 - 98.

Uieda, L. (2018). Verde: Processing and gridding spatial data using Green's functions. Journal of Open Source Software, 3(29), 957. doi:10.21105/joss.00957

Fedi, M., and M. Pilkington (2012), Understanding imaging methods for potential field data, Geophysics, 77(1), G13, doi:10.1190/geo2011-0078.1

3.2 Which Python?

For the moment, pydEXP is only tested on Python 3.7. First, make sure you have all dependencies (see below) installed.

Clone the gitlab repository:

```
git clone https://github.com/BenjMy/dEXP_imaging.git
```

From the same *dEXP_imaging* directory you can import the module from source using python.

3.3 Dependencies

pydEXP requires the following dependencies for running:

- numpy
- scipy
- matplotlib
- pandas
- mpl_axes_aligner

In order to make upward continuation and derivate the field, pyDEXP uses: * fatiando

3.4 Optionnal Third party packages

In order to forward model the geolectrical data, pyDEXP uses:

- Pygimli
- Resipy

You can test the installation running one of the exemple.

CHAPTER

FOUR

GALLERY OF EXAMPLES

4.1 Gallery of examples

Below is a gallery of examples for different physics: gravimetry, magnetic and geoelectrical methods (SP and active ERT).

4.2 Gravimetric potential field data

- estimate of the **depth of the density** anomaly using the dEXP tranformation method

4.3 Magnetic potential field data

Sources properties:

  - radius = 1.5e3
  - inc = 50
  - dec = -30

- Identify 2 depths of sources produces by 2 distincts magnetic sources using the geometrical method
- in prep Identify 2 depths of sources produces by 2 distincts magnetic sources using the dexp ratio method

4.4 Mise-à-la-masse analysis using the dEXP theory

The theory of the dEXP theory is applied here for the first time (to our knowledges) for active geoelectrical methods. The theory behind this choice and required adjustements are explained in the documentation section.

We show here:

- **preprocessing** of MALM for dEXP;
- a sensitivity analysis to **anomaly depth** (3d) of the dEXP theory apply to Mise-à-la-Masse data;
- effect of **noise level**;
- **in prep**: a case study where we used a Mise-à-la-Masse survey to identify a leakage in a landfill;
- **in prep**: effect of borehole data and downward continuation.

4.4.1 Gravimetric potential field data

- estimate of the **depth of the density** anomaly using the dEXP tranformation method

Example of gravimetric potential field data analysis using pyDEXP

This code shows a step-by-step processing of potential field imaging aiming at giving an estimate of the depth of the anomaly depth using the dEXP tranformation method. dEXP method implementation from Fedi et al. 2012. Calculations used *dEXP*, while plotting use the *plot_dEXP* module.

The gravimetric model data was created using geometric objects from `fatiando.mesher`. The forward simulation of the data was done using `fatiando.gravmag` module.

Sources locations:
Center of mass = [,,] # xyz coordinates l.w.h = ,,, # length, width, height (in m)

Sources properties:
density contrast= ?

Implements the DEXP method described in Fedi and Pilkington (2012). Application on a anomaly of density (gravimetry).

Note: This is part of a larger project aiming at inverting current sources density (see more at: https://icsd-dev.readthedocs.io/en/latest/)

References

Uieda, L., V. C. Oliveira Jr, and V. C. F. Barbosa (2013), Modeling the Earth with Fatiando a Terra, Proceedings of the 12th Python in Science Conference, pp. 91 - 98.

Uieda, L. (2018). Verde: Processing and gridding spatial data using Green's functions. Journal of Open Source Software, 3(29), 957. doi:10.21105/joss.00957

Fedi, M., and M. Pilkington (2012), Understanding imaging methods for potential field data, Geophysics, 77(1), G13, doi:10.1190/geo2011-0078.1

```
import os

from fatiando.vis import mpl #, myv
from fatiando import gridder, mesher, utils
from fatiando.gravmag import prism, imaging, transform
from fatiando.vis.mpl import square

# my own functions
import lib.dEXP as dEXP
from lib.dEXP import _fit
import lib.plot_dEXP as pEXP
import lib.set_parameters as para

# exemples
import examples.gravimetry.loadgrav.grav_models as grav

import numpy as np
```

(continues on next page)

(continued from previous page)

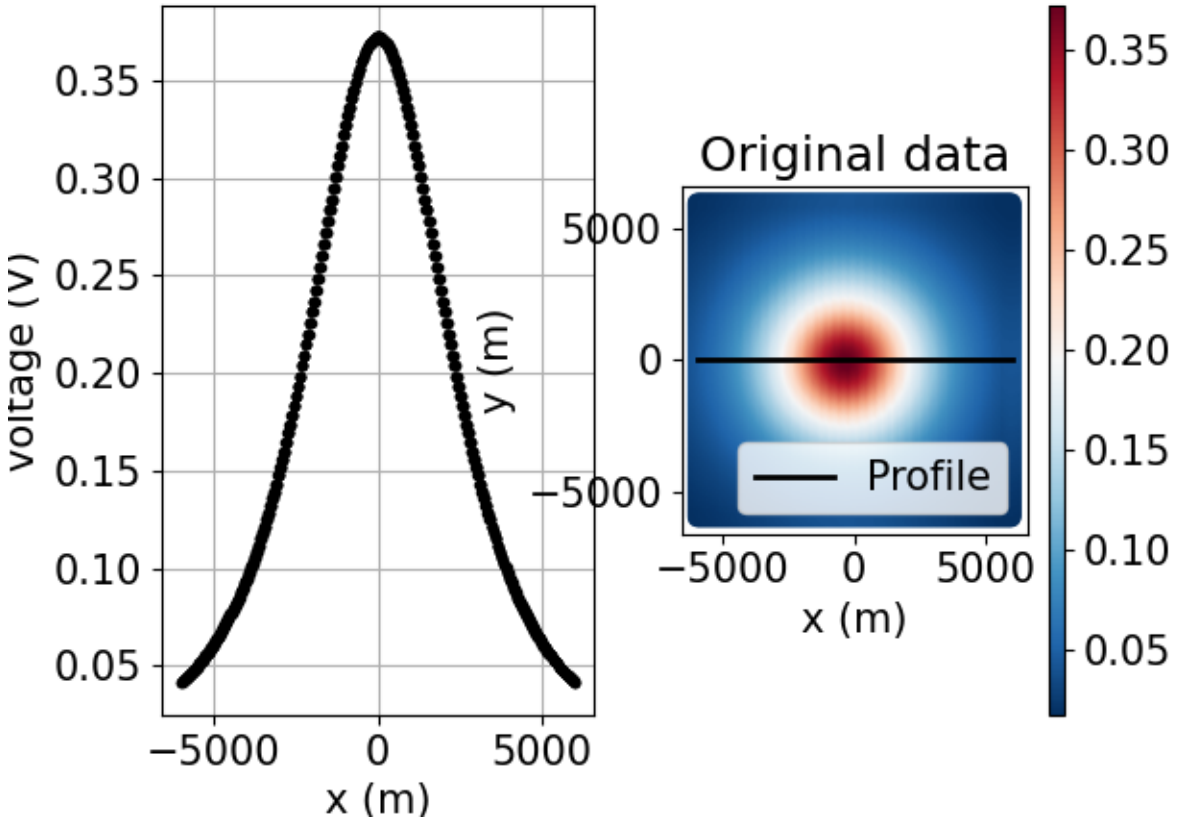
```
import matplotlib.pyplot as plt
from matplotlib import ticker

plt.rcParams['font.size'] = 15
```

load model previously generated using Fatiando a terra package

```
os.getcwd()
data_struct = grav.load_grav_fatiando(name='loadgrav/za3000_zb3500_l500_ofs0_dens1200.pkl
↪')

xp,yp,zp,U = data_struct['xyzg']
shape = data_struct['shape']
model = data_struct['model']
dens  = data_struct['density']
# scaled, SI, zp, qorder, nlay, minAlt_ridge, maxAlt_ridge = para.set_par(shape=shape,
↪max_elevation=max_elevation)

x1, x2, y1, y2, z1, z2 = np.array(model[0].get_bounds())

p1 =[min(yp),0]
p2 =[max(yp),0]

max_elevation=z2*1.2
scaled, SI, zp, qorder, nlay, minAlt_ridge, maxAlt_ridge = para.set_par(shape=shape,max_
↪elevation=max_elevation)
interp = True
qorder = 0

x_axis='y'
```

Plot the data

```
pEXP.plot_line(xp, yp, U,p1,p2, interp=interp,Xaxis=x_axis)
```




```
(array([-6000.        , -5987.98798799, -5975.97597598, -5963.96396396,
       -5951.95195195, -5939.93993994, -5927.92792793, -5915.91591592,
       -5903.9039039 , -5891.89189189, -5879.87987988, -5867.86786787,
       -5855.85585586, -5843.84384384, -5831.83183183, -5819.81981982,
       -5807.80780781, -5795.7957958 , -5783.78378378, -5771.77177177,
       -5759.75975976, -5747.74774775, -5735.73573574, -5723.72372372,
       -5711.71171171, -5699.6996997 , -5687.68768769, -5675.67567568,
       -5663.66366366, -5651.65165165, -5639.63963964, -5627.62762763,
       -5615.61561562, -5603.6036036 , -5591.59159159, -5579.57957958,
       -5567.56756757, -5555.55555556, -5543.54354354, -5531.53153153,
       -5519.51951952, -5507.50750751, -5495.4954955 , -5483.48348348,
       -5471.47147147, -5459.45945946, -5447.44744745, -5435.43543544,
       -5423.42342342, -5411.41141141, -5399.3993994 , -5387.38738739,
       -5375.37537538, -5363.36336336, -5351.35135135, -5339.33933934,
       -5327.32732733, -5315.31531532, -5303.3033033 , -5291.29129129,
       -5279.27927928, -5267.26726727, -5255.25525526, -5243.24324324,
       -5231.23123123, -5219.21921922, -5207.20720721, -5195.1951952 ,
       -5183.18318318, -5171.17117117, -5159.15915916, -5147.14714715,
       -5135.13513514, -5123.12312312, -5111.11111111, -5099.0990991 ,
       -5087.08708709, -5075.07507508, -5063.06306306, -5051.05105105,
       -5039.03903904, -5027.02702703, -5015.01501502, -5003.003003  ,
       -4990.99099099, -4978.97897898, -4966.96696697, -4954.95495495,
       -4942.94294294, -4930.93093093, -4918.91891892, -4906.90690691,
```

(continues on next page)

(continued from previous page)

```
       -4894.89489489, -4882.88288288, -4870.87087087, -4858.85885886,
       -4846.84684685, -4834.83483483, -4822.82282282, -4810.81081081,
       -4798.7987988 , -4786.78678679, -4774.77477477, -4762.76276276,
       -4750.75075075, -4738.73873874, -4726.72672673, -4714.71471471,
       -4702.7027027 , -4690.69069069, -4678.67867868, -4666.66666667,
       -4654.65465465, -4642.64264264, -4630.63063063, -4618.61861862,
       -4606.60660661, -4594.59459459, -4582.58258258, -4570.57057057,
       -4558.55855856, -4546.54654655, -4534.53453453, -4522.52252252,
       -4510.51051051, -4498.4984985 , -4486.48648649, -4474.47447447,
       -4462.46246246, -4450.45045045, -4438.43843844, -4426.42642643,
       -4414.41441441, -4402.4024024 , -4390.39039039, -4378.37837838,
       -4366.36636637, -4354.35435435, -4342.34234234, -4330.33033033,
       -4318.31831832, -4306.30630631, -4294.29429429, -4282.28228228,
       -4270.27027027, -4258.25825826, -4246.24624625, -4234.23423423,
       -4222.22222222, -4210.21021021, -4198.1981982 , -4186.18618619,
       -4174.17417417, -4162.16216216, -4150.15015015, -4138.13813814,
       -4126.12612613, -4114.11411411, -4102.1021021 , -4090.09009009,
       -4078.07807808, -4066.06606607, -4054.05405405, -4042.04204204,
       -4030.03003003, -4018.01801802, -4006.00600601, -3993.99399399,
       -3981.98198198, -3969.96996997, -3957.95795796, -3945.94594595,
       -3933.93393393, -3921.92192192, -3909.90990991, -3897.8978979 ,
       -3885.88588589, -3873.87387387, -3861.86186186, -3849.84984985,
       -3837.83783784, -3825.82582583, -3813.81381381, -3801.8018018 ,
       -3789.78978979, -3777.77777778, -3765.76576577, -3753.75375375,
       -3741.74174174, -3729.72972973, -3717.71771772, -3705.70570571,
       -3693.69369369, -3681.68168168, -3669.66966967, -3657.65765766,
       -3645.64564565, -3633.63363363, -3621.62162162, -3609.60960961,
       -3597.5975976 , -3585.58558559, -3573.57357357, -3561.56156156,
       -3549.54954955, -3537.53753754, -3525.52552553, -3513.51351351,
       -3501.5015015 , -3489.48948949, -3477.47747748, -3465.46546547,
       -3453.45345345, -3441.44144144, -3429.42942943, -3417.41741742,
       -3405.40540541, -3393.39339339, -3381.38138138, -3369.36936937,
       -3357.35735736, -3345.34534535, -3333.33333333, -3321.32132132,
       -3309.30930931, -3297.2972973 , -3285.28528529, -3273.27327327,
       -3261.26126126, -3249.24924925, -3237.23723724, -3225.22522523,
       -3213.21321321, -3201.2012012 , -3189.18918919, -3177.17717718,
       -3165.16516517, -3153.15315315, -3141.14114114, -3129.12912913,
       -3117.11711712, -3105.10510511, -3093.09309309, -3081.08108108,
       -3069.06906907, -3057.05705706, -3045.04504505, -3033.03303303,
       -3021.02102102, -3009.00900901, -2996.996997  , -2984.98498498,
       -2972.97297297, -2960.96096096, -2948.94894895, -2936.93693694,
       -2924.92492492, -2912.91291291, -2900.9009009 , -2888.88888889,
       -2876.87687688, -2864.86486486, -2852.85285285, -2840.84084084,
       -2828.82882883, -2816.81681682, -2804.8048048 , -2792.79279279,
       -2780.78078078, -2768.76876877, -2756.75675676, -2744.74474474,
       -2732.73273273, -2720.72072072, -2708.70870871, -2696.6966967 ,
       -2684.68468468, -2672.67267267, -2660.66066066, -2648.64864865,
       -2636.63663664, -2624.62462462, -2612.61261261, -2600.6006006 ,
       -2588.58858859, -2576.57657658, -2564.56456456, -2552.55255255,
       -2540.54054054, -2528.52852853, -2516.51651652, -2504.5045045 ,
       -2492.49249249, -2480.48048048, -2468.46846847, -2456.45645646,
       -2444.44444444, -2432.43243243, -2420.42042042, -2408.40840841,
```

(continues on next page)

(continued from previous page)

```
       -2396.3963964 , -2384.38438438, -2372.37237237, -2360.36036036,
       -2348.34834835, -2336.33633634, -2324.32432432, -2312.31231231,
       -2300.3003003 , -2288.28828829, -2276.27627628, -2264.26426426,
       -2252.25225225, -2240.24024024, -2228.22822823, -2216.21621622,
       -2204.2042042 , -2192.19219219, -2180.18018018, -2168.16816817,
       -2156.15615616, -2144.14414414, -2132.13213213, -2120.12012012,
       -2108.10810811, -2096.0960961 , -2084.08408408, -2072.07207207,
       -2060.06006006, -2048.04804805, -2036.03603604, -2024.02402402,
       -2012.01201201, -2000.        , -1987.98798799, -1975.97597598,
       -1963.96396396, -1951.95195195, -1939.93993994, -1927.92792793,
       -1915.91591592, -1903.9039039 , -1891.89189189, -1879.87987988,
       -1867.86786787, -1855.85585586, -1843.84384384, -1831.83183183,
       -1819.81981982, -1807.80780781, -1795.7957958 , -1783.78378378,
       -1771.77177177, -1759.75975976, -1747.74774775, -1735.73573574,
       -1723.72372372, -1711.71171171, -1699.6996997 , -1687.68768769,
       -1675.67567568, -1663.66366366, -1651.65165165, -1639.63963964,
       -1627.62762763, -1615.61561562, -1603.6036036 , -1591.59159159,
       -1579.57957958, -1567.56756757, -1555.55555556, -1543.54354354,
       -1531.53153153, -1519.51951952, -1507.50750751, -1495.4954955 ,
       -1483.48348348, -1471.47147147, -1459.45945946, -1447.44744745,
       -1435.43543544, -1423.42342342, -1411.41141141, -1399.3993994 ,
       -1387.38738739, -1375.37537538, -1363.36336336, -1351.35135135,
       -1339.33933934, -1327.32732733, -1315.31531532, -1303.3033033 ,
       -1291.29129129, -1279.27927928, -1267.26726727, -1255.25525526,
       -1243.24324324, -1231.23123123, -1219.21921922, -1207.20720721,
       -1195.1951952 , -1183.18318318, -1171.17117117, -1159.15915916,
       -1147.14714715, -1135.13513514, -1123.12312312, -1111.11111111,
       -1099.0990991 , -1087.08708709, -1075.07507508, -1063.06306306,
       -1051.05105105, -1039.03903904, -1027.02702703, -1015.01501502,
       -1003.003003  ,  -990.99099099,  -978.97897898,  -966.96696697,
        -954.95495495,  -942.94294294,  -930.93093093,  -918.91891892,
        -906.90690691,  -894.89489489,  -882.88288288,  -870.87087087,
        -858.85885886,  -846.84684685,  -834.83483483,  -822.82282282,
        -810.81081081,  -798.7987988 ,  -786.78678679,  -774.77477477,
        -762.76276276,  -750.75075075,  -738.73873874,  -726.72672673,
        -714.71471471,  -702.7027027 ,  -690.69069069,  -678.67867868,
        -666.66666667,  -654.65465465,  -642.64264264,  -630.63063063,
        -618.61861862,  -606.60660661,  -594.59459459,  -582.58258258,
        -570.57057057,  -558.55855856,  -546.54654655,  -534.53453453,
        -522.52252252,  -510.51051051,  -498.4984985 ,  -486.48648649,
        -474.47447447,  -462.46246246,  -450.45045045,  -438.43843844,
        -426.42642643,  -414.41441441,  -402.4024024 ,  -390.39039039,
        -378.37837838,  -366.36636637,  -354.35435435,  -342.34234234,
        -330.33033033,  -318.31831832,  -306.30630631,  -294.29429429,
        -282.28228228,  -270.27027027,  -258.25825826,  -246.24624625,
        -234.23423423,  -222.22222222,  -210.21021021,  -198.1981982 ,
        -186.18618619,  -174.17417417,  -162.16216216,  -150.15015015,
        -138.13813814,  -126.12612613,  -114.11411411,  -102.1021021 ,
         -90.09009009,   -78.07807808,   -66.06606607,   -54.05405405,
         -42.04204204,   -30.03003003,   -18.01801802,    -6.00600601,
           6.00600601,    18.01801802,    30.03003003,    42.04204204,
          54.05405405,    66.06606607,    78.07807808,    90.09009009,
```

(continues on next page)

(continued from previous page)

```
       102.1021021 ,   114.11411411,   126.12612613,   138.13813814,
       150.15015015,   162.16216216,   174.17417417,   186.18618619,
       198.1981982 ,   210.21021021,   222.22222222,   234.23423423,
       246.24624625,   258.25825826,   270.27027027,   282.28228228,
       294.29429429,   306.30630631,   318.31831832,   330.33033033,
       342.34234234,   354.35435435,   366.36636637,   378.37837838,
       390.39039039,   402.4024024 ,   414.41441441,   426.42642643,
       438.43843844,   450.45045045,   462.46246246,   474.47447447,
       486.48648649,   498.4984985 ,   510.51051051,   522.52252252,
       534.53453453,   546.54654655,   558.55855856,   570.57057057,
       582.58258258,   594.59459459,   606.60660661,   618.61861862,
       630.63063063,   642.64264264,   654.65465465,   666.66666667,
       678.67867868,   690.69069069,   702.7027027 ,   714.71471471,
       726.72672673,   738.73873874,   750.75075075,   762.76276276,
       774.77477477,   786.78678679,   798.7987988 ,   810.81081081,
       822.82282282,   834.83483483,   846.84684685,   858.85885886,
       870.87087087,   882.88288288,   894.89489489,   906.90690691,
       918.91891892,   930.93093093,   942.94294294,   954.95495495,
       966.96696697,   978.97897898,   990.99099099,  1003.003003  ,
      1015.01501502,  1027.02702703,  1039.03903904,  1051.05105105,
      1063.06306306,  1075.07507508,  1087.08708709,  1099.0990991 ,
      1111.11111111,  1123.12312312,  1135.13513514,  1147.14714715,
      1159.15915916,  1171.17117117,  1183.18318318,  1195.1951952 ,
      1207.20720721,  1219.21921922,  1231.23123123,  1243.24324324,
      1255.25525526,  1267.26726727,  1279.27927928,  1291.29129129,
      1303.3033033 ,  1315.31531532,  1327.32732733,  1339.33933934,
      1351.35135135,  1363.36336336,  1375.37537538,  1387.38738739,
      1399.3993994 ,  1411.41141141,  1423.42342342,  1435.43543544,
      1447.44744745,  1459.45945946,  1471.47147147,  1483.48348348,
      1495.4954955 ,  1507.50750751,  1519.51951952,  1531.53153153,
      1543.54354354,  1555.55555556,  1567.56756757,  1579.57957958,
      1591.59159159,  1603.6036036 ,  1615.61561562,  1627.62762763,
      1639.63963964,  1651.65165165,  1663.66366366,  1675.67567568,
      1687.68768769,  1699.6996997 ,  1711.71171171,  1723.72372372,
      1735.73573574,  1747.74774775,  1759.75975976,  1771.77177177,
      1783.78378378,  1795.7957958 ,  1807.80780781,  1819.81981982,
      1831.83183183,  1843.84384384,  1855.85585586,  1867.86786787,
      1879.87987988,  1891.89189189,  1903.9039039 ,  1915.91591592,
      1927.92792793,  1939.93993994,  1951.95195195,  1963.96396396,
      1975.97597598,  1987.98798799,  2000.        ,  2012.01201201,
      2024.02402402,  2036.03603604,  2048.04804805,  2060.06006006,
      2072.07207207,  2084.08408408,  2096.0960961 ,  2108.10810811,
      2120.12012012,  2132.13213213,  2144.14414414,  2156.15615616,
      2168.16816817,  2180.18018018,  2192.19219219,  2204.2042042 ,
      2216.21621622,  2228.22822823,  2240.24024024,  2252.25225225,
      2264.26426426,  2276.27627628,  2288.28828829,  2300.3003003 ,
      2312.31231231,  2324.32432432,  2336.33633634,  2348.34834835,
      2360.36036036,  2372.37237237,  2384.38438438,  2396.3963964 ,
      2408.40840841,  2420.42042042,  2432.43243243,  2444.44444444,
      2456.45645646,  2468.46846847,  2480.48048048,  2492.49249249,
      2504.5045045 ,  2516.51651652,  2528.52852853,  2540.54054054,
      2552.55255255,  2564.56456456,  2576.57657658,  2588.58858859,
```

(continues on next page)

(continued from previous page)

```
       2600.6006006 ,  2612.61261261,  2624.62462462,  2636.63663664,
       2648.64864865,  2660.66066066,  2672.67267267,  2684.68468468,
       2696.6966967 ,  2708.70870871,  2720.72072072,  2732.73273273,
       2744.74474474,  2756.75675676,  2768.76876877,  2780.78078078,
       2792.79279279,  2804.8048048 ,  2816.81681682,  2828.82882883,
       2840.84084084,  2852.85285285,  2864.86486486,  2876.87687688,
       2888.88888889,  2900.9009009 ,  2912.91291291,  2924.92492492,
       2936.93693694,  2948.94894895,  2960.96096096,  2972.97297297,
       2984.98498498,  2996.996997  ,  3009.00900901,  3021.02102102,
       3033.03303303,  3045.04504505,  3057.05705706,  3069.06906907,
       3081.08108108,  3093.09309309,  3105.10510511,  3117.11711712,
       3129.12912913,  3141.14114114,  3153.15315315,  3165.16516517,
       3177.17717718,  3189.18918919,  3201.2012012 ,  3213.21321321,
       3225.22522523,  3237.23723724,  3249.24924925,  3261.26126126,
       3273.27327327,  3285.28528529,  3297.2972973 ,  3309.30930931,
       3321.32132132,  3333.33333333,  3345.34534535,  3357.35735736,
       3369.36936937,  3381.38138138,  3393.39339339,  3405.40540541,
       3417.41741742,  3429.42942943,  3441.44144144,  3453.45345345,
       3465.46546547,  3477.47747748,  3489.48948949,  3501.5015015 ,
       3513.51351351,  3525.52552553,  3537.53753754,  3549.54954955,
       3561.56156156,  3573.57357357,  3585.58558559,  3597.5975976 ,
       3609.60960961,  3621.62162162,  3633.63363363,  3645.64564565,
       3657.65765766,  3669.66966967,  3681.68168168,  3693.69369369,
       3705.70570571,  3717.71771772,  3729.72972973,  3741.74174174,
       3753.75375375,  3765.76576577,  3777.77777778,  3789.78978979,
       3801.8018018 ,  3813.81381381,  3825.82582583,  3837.83783784,
       3849.84984985,  3861.86186186,  3873.87387387,  3885.88588589,
       3897.8978979 ,  3909.90990991,  3921.92192192,  3933.93393393,
       3945.94594595,  3957.95795796,  3969.96996997,  3981.98198198,
       3993.99399399,  4006.00600601,  4018.01801802,  4030.03003003,
       4042.04204204,  4054.05405405,  4066.06606607,  4078.07807808,
       4090.09009009,  4102.1021021 ,  4114.11411411,  4126.12612613,
       4138.13813814,  4150.15015015,  4162.16216216,  4174.17417417,
       4186.18618619,  4198.1981982 ,  4210.21021021,  4222.22222222,
       4234.23423423,  4246.24624625,  4258.25825826,  4270.27027027,
       4282.28228228,  4294.29429429,  4306.30630631,  4318.31831832,
       4330.33033033,  4342.34234234,  4354.35435435,  4366.36636637,
       4378.37837838,  4390.39039039,  4402.4024024 ,  4414.41441441,
       4426.42642643,  4438.43843844,  4450.45045045,  4462.46246246,
       4474.47447447,  4486.48648649,  4498.4984985 ,  4510.51051051,
       4522.52252252,  4534.53453453,  4546.54654655,  4558.55855856,
       4570.57057057,  4582.58258258,  4594.59459459,  4606.60660661,
       4618.61861862,  4630.63063063,  4642.64264264,  4654.65465465,
       4666.66666667,  4678.67867868,  4690.69069069,  4702.7027027 ,
       4714.71471471,  4726.72672673,  4738.73873874,  4750.75075075,
       4762.76276276,  4774.77477477,  4786.78678679,  4798.7987988 ,
       4810.81081081,  4822.82282282,  4834.83483483,  4846.84684685,
       4858.85885886,  4870.87087087,  4882.88288288,  4894.89489489,
       4906.90690691,  4918.91891892,  4930.93093093,  4942.94294294,
       4954.95495495,  4966.96696697,  4978.97897898,  4990.99099099,
       5003.003003  ,  5015.01501502,  5027.02702703,  5039.03903904,
       5051.05105105,  5063.06306306,  5075.07507508,  5087.08708709,
```

(continues on next page)

(continued from previous page)

```
        5099.0990991 ,  5111.11111111,  5123.12312312,  5135.13513514,
        5147.14714715,  5159.15915916,  5171.17117117,  5183.18318318,
        5195.1951952 ,  5207.20720721,  5219.21921922,  5231.23123123,
        5243.24324324,  5255.25525526,  5267.26726727,  5279.27927928,
        5291.29129129,  5303.3033033 ,  5315.31531532,  5327.32732733,
        5339.33933934,  5351.35135135,  5363.36336336,  5375.37537538,
        5387.38738739,  5399.3993994 ,  5411.41141141,  5423.42342342,
        5435.43543544,  5447.44744745,  5459.45945946,  5471.47147147,
        5483.48348348,  5495.4954955 ,  5507.50750751,  5519.51951952,
        5531.53153153,  5543.54354354,  5555.55555556,  5567.56756757,
        5579.57957958,  5591.59159159,  5603.6036036 ,  5615.61561562,
        5627.62762763,  5639.63963964,  5651.65165165,  5663.66366366,
        5675.67567568,  5687.68768769,  5699.6996997 ,  5711.71171171,
        5723.72372372,  5735.73573574,  5747.74774775,  5759.75975976,
        5771.77177177,  5783.78378378,  5795.7957958 ,  5807.80780781,
        5819.81981982,  5831.83183183,  5843.84384384,  5855.85585586,
        5867.86786787,  5879.87987988,  5891.89189189,  5903.9039039 ,
        5915.91591592,  5927.92792793,  5939.93993994,  5951.95195195,
        5963.96396396,  5975.97597598,  5987.98798799,  6000.        ]), array([0., 0.,␣
↪0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
```

(continues on next page)

(continued from previous page)

```
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]), array([    0.          , ␣
↪   12.01201201,    24.02402402,    36.03603604,
         48.04804805,    60.06006006,    72.07207207,    84.08408408,
         96.0960961 ,   108.10810811,   120.12012012,   132.13213213,
        144.14414414,   156.15615616,   168.16816817,   180.18018018,
        192.19219219,   204.2042042 ,   216.21621622,   228.22822823,
        240.24024024,   252.25225225,   264.26426426,   276.27627628,
        288.28828829,   300.3003003 ,   312.31231231,   324.32432432,
        336.33633634,   348.34834835,   360.36036036,   372.37237237,
        384.38438438,   396.3963964 ,   408.40840841,   420.42042042,
        432.43243243,   444.44444444,   456.45645646,   468.46846847,
        480.48048048,   492.49249249,   504.5045045 ,   516.51651652,
        528.52852853,   540.54054054,   552.55255255,   564.56456456,
        576.57657658,   588.58858859,   600.6006006 ,   612.61261261,
        624.62462462,   636.63663664,   648.64864865,   660.66066066,
        672.67267267,   684.68468468,   696.6966967 ,   708.70870871,
        720.72072072,   732.73273273,   744.74474474,   756.75675676,
        768.76876877,   780.78078078,   792.79279279,   804.8048048 ,
        816.81681682,   828.82882883,   840.84084084,   852.85285285,
        864.86486486,   876.87687688,   888.88888889,   900.9009009 ,
        912.91291291,   924.92492492,   936.93693694,   948.94894895,
        960.96096096,   972.97297297,   984.98498498,   996.996997  ,
       1009.00900901,  1021.02102102,  1033.03303303,  1045.04504505,
       1057.05705706,  1069.06906907,  1081.08108108,  1093.09309309,
       1105.10510511,  1117.11711712,  1129.12912913,  1141.14114114,
       1153.15315315,  1165.16516517,  1177.17717718,  1189.18918919,
       1201.2012012 ,  1213.21321321,  1225.22522523,  1237.23723724,
```

(continues on next page)

(continued from previous page)

```
       1249.24924925,  1261.26126126,  1273.27327327,  1285.28528529,
       1297.2972973 ,  1309.30930931,  1321.32132132,  1333.33333333,
       1345.34534535,  1357.35735736,  1369.36936937,  1381.38138138,
       1393.39339339,  1405.40540541,  1417.41741742,  1429.42942943,
       1441.44144144,  1453.45345345,  1465.46546547,  1477.47747748,
       1489.48948949,  1501.5015015 ,  1513.51351351,  1525.52552553,
       1537.53753754,  1549.54954955,  1561.56156156,  1573.57357357,
       1585.58558559,  1597.5975976 ,  1609.60960961,  1621.62162162,
       1633.63363363,  1645.64564565,  1657.65765766,  1669.66966967,
       1681.68168168,  1693.69369369,  1705.70570571,  1717.71771772,
       1729.72972973,  1741.74174174,  1753.75375375,  1765.76576577,
       1777.77777778,  1789.78978979,  1801.8018018 ,  1813.81381381,
       1825.82582583,  1837.83783784,  1849.84984985,  1861.86186186,
       1873.87387387,  1885.88588589,  1897.8978979 ,  1909.90990991,
       1921.92192192,  1933.93393393,  1945.94594595,  1957.95795796,
       1969.96996997,  1981.98198198,  1993.99399399,  2006.00600601,
       2018.01801802,  2030.03003003,  2042.04204204,  2054.05405405,
       2066.06606607,  2078.07807808,  2090.09009009,  2102.1021021 ,
       2114.11411411,  2126.12612613,  2138.13813814,  2150.15015015,
       2162.16216216,  2174.17417417,  2186.18618619,  2198.1981982 ,
       2210.21021021,  2222.22222222,  2234.23423423,  2246.24624625,
       2258.25825826,  2270.27027027,  2282.28228228,  2294.29429429,
       2306.30630631,  2318.31831832,  2330.33033033,  2342.34234234,
       2354.35435435,  2366.36636637,  2378.37837838,  2390.39039039,
       2402.4024024 ,  2414.41441441,  2426.42642643,  2438.43843844,
       2450.45045045,  2462.46246246,  2474.47447447,  2486.48648649,
       2498.4984985 ,  2510.51051051,  2522.52252252,  2534.53453453,
       2546.54654655,  2558.55855856,  2570.57057057,  2582.58258258,
       2594.59459459,  2606.60660661,  2618.61861862,  2630.63063063,
       2642.64264264,  2654.65465465,  2666.66666667,  2678.67867868,
       2690.69069069,  2702.7027027 ,  2714.71471471,  2726.72672673,
       2738.73873874,  2750.75075075,  2762.76276276,  2774.77477477,
       2786.78678679,  2798.7987988 ,  2810.81081081,  2822.82282282,
       2834.83483483,  2846.84684685,  2858.85885886,  2870.87087087,
       2882.88288288,  2894.89489489,  2906.90690691,  2918.91891892,
       2930.93093093,  2942.94294294,  2954.95495495,  2966.96696697,
       2978.97897898,  2990.99099099,  3003.003003  ,  3015.01501502,
       3027.02702703,  3039.03903904,  3051.05105105,  3063.06306306,
       3075.07507508,  3087.08708709,  3099.0990991 ,  3111.11111111,
       3123.12312312,  3135.13513514,  3147.14714715,  3159.15915916,
       3171.17117117,  3183.18318318,  3195.1951952 ,  3207.20720721,
       3219.21921922,  3231.23123123,  3243.24324324,  3255.25525526,
       3267.26726727,  3279.27927928,  3291.29129129,  3303.3033033 ,
       3315.31531532,  3327.32732733,  3339.33933934,  3351.35135135,
       3363.36336336,  3375.37537538,  3387.38738739,  3399.3993994 ,
       3411.41141141,  3423.42342342,  3435.43543544,  3447.44744745,
       3459.45945946,  3471.47147147,  3483.48348348,  3495.4954955 ,
       3507.50750751,  3519.51951952,  3531.53153153,  3543.54354354,
       3555.55555556,  3567.56756757,  3579.57957958,  3591.59159159,
       3603.6036036 ,  3615.61561562,  3627.62762763,  3639.63963964,
       3651.65165165,  3663.66366366,  3675.67567568,  3687.68768769,
       3699.6996997 ,  3711.71171171,  3723.72372372,  3735.73573574,
```

(continues on next page)

(continued from previous page)

```
       3747.74774775,  3759.75975976,  3771.77177177,  3783.78378378,
       3795.7957958 ,  3807.80780781,  3819.81981982,  3831.83183183,
       3843.84384384,  3855.85585586,  3867.86786787,  3879.87987988,
       3891.89189189,  3903.9039039 ,  3915.91591592,  3927.92792793,
       3939.93993994,  3951.95195195,  3963.96396396,  3975.97597598,
       3987.98798799,  4000.        ,  4012.01201201,  4024.02402402,
       4036.03603604,  4048.04804805,  4060.06006006,  4072.07207207,
       4084.08408408,  4096.0960961 ,  4108.10810811,  4120.12012012,
       4132.13213213,  4144.14414414,  4156.15615616,  4168.16816817,
       4180.18018018,  4192.19219219,  4204.2042042 ,  4216.21621622,
       4228.22822823,  4240.24024024,  4252.25225225,  4264.26426426,
       4276.27627628,  4288.28828829,  4300.3003003 ,  4312.31231231,
       4324.32432432,  4336.33633634,  4348.34834835,  4360.36036036,
       4372.37237237,  4384.38438438,  4396.3963964 ,  4408.40840841,
       4420.42042042,  4432.43243243,  4444.44444444,  4456.45645646,
       4468.46846847,  4480.48048048,  4492.49249249,  4504.5045045 ,
       4516.51651652,  4528.52852853,  4540.54054054,  4552.55255255,
       4564.56456456,  4576.57657658,  4588.58858859,  4600.6006006 ,
       4612.61261261,  4624.62462462,  4636.63663664,  4648.64864865,
       4660.66066066,  4672.67267267,  4684.68468468,  4696.6966967 ,
       4708.70870871,  4720.72072072,  4732.73273273,  4744.74474474,
       4756.75675676,  4768.76876877,  4780.78078078,  4792.79279279,
       4804.8048048 ,  4816.81681682,  4828.82882883,  4840.84084084,
       4852.85285285,  4864.86486486,  4876.87687688,  4888.88888889,
       4900.9009009 ,  4912.91291291,  4924.92492492,  4936.93693694,
       4948.94894895,  4960.96096096,  4972.97297297,  4984.98498498,
       4996.996997  ,  5009.00900901,  5021.02102102,  5033.03303303,
       5045.04504505,  5057.05705706,  5069.06906907,  5081.08108108,
       5093.09309309,  5105.10510511,  5117.11711712,  5129.12912913,
       5141.14114114,  5153.15315315,  5165.16516517,  5177.17717718,
       5189.18918919,  5201.2012012 ,  5213.21321321,  5225.22522523,
       5237.23723724,  5249.24924925,  5261.26126126,  5273.27327327,
       5285.28528529,  5297.2972973 ,  5309.30930931,  5321.32132132,
       5333.33333333,  5345.34534535,  5357.35735736,  5369.36936937,
       5381.38138138,  5393.39339339,  5405.40540541,  5417.41741742,
       5429.42942943,  5441.44144144,  5453.45345345,  5465.46546547,
       5477.47747748,  5489.48948949,  5501.5015015 ,  5513.51351351,
       5525.52552553,  5537.53753754,  5549.54954955,  5561.56156156,
       5573.57357357,  5585.58558559,  5597.5975976 ,  5609.60960961,
       5621.62162162,  5633.63363363,  5645.64564565,  5657.65765766,
       5669.66966967,  5681.68168168,  5693.69369369,  5705.70570571,
       5717.71771772,  5729.72972973,  5741.74174174,  5753.75375375,
       5765.76576577,  5777.77777778,  5789.78978979,  5801.8018018 ,
       5813.81381381,  5825.82582583,  5837.83783784,  5849.84984985,
       5861.86186186,  5873.87387387,  5885.88588589,  5897.8978979 ,
       5909.90990991,  5921.92192192,  5933.93393393,  5945.94594595,
       5957.95795796,  5969.96996997,  5981.98198198,  5993.99399399,
       6006.00600601,  6018.01801802,  6030.03003003,  6042.04204204,
       6054.05405405,  6066.06606607,  6078.07807808,  6090.09009009,
       6102.1021021 ,  6114.11411411,  6126.12612613,  6138.13813814,
       6150.15015015,  6162.16216216,  6174.17417417,  6186.18618619,
       6198.1981982 ,  6210.21021021,  6222.22222222,  6234.23423423,
```

(continues on next page)

(continued from previous page)

```
       6246.24624625,  6258.25825826,  6270.27027027,  6282.28228228,
       6294.29429429,  6306.30630631,  6318.31831832,  6330.33033033,
       6342.34234234,  6354.35435435,  6366.36636637,  6378.37837838,
       6390.39039039,  6402.4024024 ,  6414.41441441,  6426.42642643,
       6438.43843844,  6450.45045045,  6462.46246246,  6474.47447447,
       6486.48648649,  6498.4984985 ,  6510.51051051,  6522.52252252,
       6534.53453453,  6546.54654655,  6558.55855856,  6570.57057057,
       6582.58258258,  6594.59459459,  6606.60660661,  6618.61861862,
       6630.63063063,  6642.64264264,  6654.65465465,  6666.66666667,
       6678.67867868,  6690.69069069,  6702.7027027 ,  6714.71471471,
       6726.72672673,  6738.73873874,  6750.75075075,  6762.76276276,
       6774.77477477,  6786.78678679,  6798.7987988 ,  6810.81081081,
       6822.82282282,  6834.83483483,  6846.84684685,  6858.85885886,
       6870.87087087,  6882.88288288,  6894.89489489,  6906.90690691,
       6918.91891892,  6930.93093093,  6942.94294294,  6954.95495495,
       6966.96696697,  6978.97897898,  6990.99099099,  7003.003003  ,
       7015.01501502,  7027.02702703,  7039.03903904,  7051.05105105,
       7063.06306306,  7075.07507508,  7087.08708709,  7099.0990991 ,
       7111.11111111,  7123.12312312,  7135.13513514,  7147.14714715,
       7159.15915916,  7171.17117117,  7183.18318318,  7195.1951952 ,
       7207.20720721,  7219.21921922,  7231.23123123,  7243.24324324,
       7255.25525526,  7267.26726727,  7279.27927928,  7291.29129129,
       7303.3033033 ,  7315.31531532,  7327.32732733,  7339.33933934,
       7351.35135135,  7363.36336336,  7375.37537538,  7387.38738739,
       7399.3993994 ,  7411.41141141,  7423.42342342,  7435.43543544,
       7447.44744745,  7459.45945946,  7471.47147147,  7483.48348348,
       7495.4954955 ,  7507.50750751,  7519.51951952,  7531.53153153,
       7543.54354354,  7555.55555556,  7567.56756757,  7579.57957958,
       7591.59159159,  7603.6036036 ,  7615.61561562,  7627.62762763,
       7639.63963964,  7651.65165165,  7663.66366366,  7675.67567568,
       7687.68768769,  7699.6996997 ,  7711.71171171,  7723.72372372,
       7735.73573574,  7747.74774775,  7759.75975976,  7771.77177177,
       7783.78378378,  7795.7957958 ,  7807.80780781,  7819.81981982,
       7831.83183183,  7843.84384384,  7855.85585586,  7867.86786787,
       7879.87987988,  7891.89189189,  7903.9039039 ,  7915.91591592,
       7927.92792793,  7939.93993994,  7951.95195195,  7963.96396396,
       7975.97597598,  7987.98798799,  8000.        ,  8012.01201201,
       8024.02402402,  8036.03603604,  8048.04804805,  8060.06006006,
       8072.07207207,  8084.08408408,  8096.0960961 ,  8108.10810811,
       8120.12012012,  8132.13213213,  8144.14414414,  8156.15615616,
       8168.16816817,  8180.18018018,  8192.19219219,  8204.2042042 ,
       8216.21621622,  8228.22822823,  8240.24024024,  8252.25225225,
       8264.26426426,  8276.27627628,  8288.28828829,  8300.3003003 ,
       8312.31231231,  8324.32432432,  8336.33633634,  8348.34834835,
       8360.36036036,  8372.37237237,  8384.38438438,  8396.3963964 ,
       8408.40840841,  8420.42042042,  8432.43243243,  8444.44444444,
       8456.45645646,  8468.46846847,  8480.48048048,  8492.49249249,
       8504.5045045 ,  8516.51651652,  8528.52852853,  8540.54054054,
       8552.55255255,  8564.56456456,  8576.57657658,  8588.58858859,
       8600.6006006 ,  8612.61261261,  8624.62462462,  8636.63663664,
       8648.64864865,  8660.66066066,  8672.67267267,  8684.68468468,
       8696.6966967 ,  8708.70870871,  8720.72072072,  8732.73273273,
```

(continues on next page)

(continued from previous page)

```
 8744.74474474,  8756.75675676,  8768.76876877,  8780.78078078,
 8792.79279279,  8804.8048048 ,  8816.81681682,  8828.82882883,
 8840.84084084,  8852.85285285,  8864.86486486,  8876.87687688,
 8888.88888889,  8900.9009009 ,  8912.91291291,  8924.92492492,
 8936.93693694,  8948.94894895,  8960.96096096,  8972.97297297,
 8984.98498498,  8996.996997  ,  9009.00900901,  9021.02102102,
 9033.03303303,  9045.04504505,  9057.05705706,  9069.06906907,
 9081.08108108,  9093.09309309,  9105.10510511,  9117.11711712,
 9129.12912913,  9141.14114114,  9153.15315315,  9165.16516517,
 9177.17717718,  9189.18918919,  9201.2012012 ,  9213.21321321,
 9225.22522523,  9237.23723724,  9249.24924925,  9261.26126126,
 9273.27327327,  9285.28528529,  9297.2972973 ,  9309.30930931,
 9321.32132132,  9333.33333333,  9345.34534535,  9357.35735736,
 9369.36936937,  9381.38138138,  9393.39339339,  9405.40540541,
 9417.41741742,  9429.42942943,  9441.44144144,  9453.45345345,
 9465.46546547,  9477.47747748,  9489.48948949,  9501.5015015 ,
 9513.51351351,  9525.52552553,  9537.53753754,  9549.54954955,
 9561.56156156,  9573.57357357,  9585.58558559,  9597.5975976 ,
 9609.60960961,  9621.62162162,  9633.63363363,  9645.64564565,
 9657.65765766,  9669.66966967,  9681.68168168,  9693.69369369,
 9705.70570571,  9717.71771772,  9729.72972973,  9741.74174174,
 9753.75375375,  9765.76576577,  9777.77777778,  9789.78978979,
 9801.8018018 ,  9813.81381381,  9825.82582583,  9837.83783784,
 9849.84984985,  9861.86186186,  9873.87387387,  9885.88588589,
 9897.8978979 ,  9909.90990991,  9921.92192192,  9933.93393393,
 9945.94594595,  9957.95795796,  9969.96996997,  9981.98198198,
 9993.99399399, 10006.00600601, 10018.01801802, 10030.03003003,
10042.04204204, 10054.05405405, 10066.06606607, 10078.07807808,
10090.09009009, 10102.1021021 , 10114.11411411, 10126.12612613,
10138.13813814, 10150.15015015, 10162.16216216, 10174.17417417,
10186.18618619, 10198.1981982 , 10210.21021021, 10222.22222222,
10234.23423423, 10246.24624625, 10258.25825826, 10270.27027027,
10282.28228228, 10294.29429429, 10306.30630631, 10318.31831832,
10330.33033033, 10342.34234234, 10354.35435435, 10366.36636637,
10378.37837838, 10390.39039039, 10402.4024024 , 10414.41441441,
10426.42642643, 10438.43843844, 10450.45045045, 10462.46246246,
10474.47447447, 10486.48648649, 10498.4984985 , 10510.51051051,
10522.52252252, 10534.53453453, 10546.54654655, 10558.55855856,
10570.57057057, 10582.58258258, 10594.59459459, 10606.60660661,
10618.61861862, 10630.63063063, 10642.64264264, 10654.65465465,
10666.66666667, 10678.67867868, 10690.69069069, 10702.7027027 ,
10714.71471471, 10726.72672673, 10738.73873874, 10750.75075075,
10762.76276276, 10774.77477477, 10786.78678679, 10798.7987988 ,
10810.81081081, 10822.82282282, 10834.83483483, 10846.84684685,
10858.85885886, 10870.87087087, 10882.88288288, 10894.89489489,
10906.90690691, 10918.91891892, 10930.93093093, 10942.94294294,
10954.95495495, 10966.96696697, 10978.97897898, 10990.99099099,
11003.003003  , 11015.01501502, 11027.02702703, 11039.03903904,
11051.05105105, 11063.06306306, 11075.07507508, 11087.08708709,
11099.0990991 , 11111.11111111, 11123.12312312, 11135.13513514,
11147.14714715, 11159.15915916, 11171.17117117, 11183.18318318,
11195.1951952 , 11207.20720721, 11219.21921922, 11231.23123123,
```

(continues on next page)

(continued from previous page)

```
       11243.24324324, 11255.25525526, 11267.26726727, 11279.27927928,
       11291.29129129, 11303.3033033 , 11315.31531532, 11327.32732733,
       11339.33933934, 11351.35135135, 11363.36336336, 11375.37537538,
       11387.38738739, 11399.3993994 , 11411.41141141, 11423.42342342,
       11435.43543544, 11447.44744745, 11459.45945946, 11471.47147147,
       11483.48348348, 11495.4954955 , 11507.50750751, 11519.51951952,
       11531.53153153, 11543.54354354, 11555.55555556, 11567.56756757,
       11579.57957958, 11591.59159159, 11603.6036036 , 11615.61561562,
       11627.62762763, 11639.63963964, 11651.65165165, 11663.66366366,
       11675.67567568, 11687.68768769, 11699.6996997 , 11711.71171171,
       11723.72372372, 11735.73573574, 11747.74774775, 11759.75975976,
       11771.77177177, 11783.78378378, 11795.7957958 , 11807.80780781,
       11819.81981982, 11831.83183183, 11843.84384384, 11855.85585586,
       11867.86786787, 11879.87987988, 11891.89189189, 11903.9039039 ,
       11915.91591592, 11927.92792793, 11939.93993994, 11951.95195195,
       11963.96396396, 11975.97597598, 11987.98798799, 12000.        ]), array([0.
↪04110633, 0.04110633, 0.04110633, 0.04207942, 0.04207942,
       0.04207942, 0.04207942, 0.04207942, 0.04308104, 0.04308104,
       0.04308104, 0.04308104, 0.04308104, 0.04411214, 0.04411214,
       0.04411214, 0.04411214, 0.04411214, 0.04517369, 0.04517369,
       0.04517369, 0.04517369, 0.04517369, 0.04626672, 0.04626672,
       0.04626672, 0.04626672, 0.04626672, 0.04739227, 0.04739227,
       0.04739227, 0.04739227, 0.04739227, 0.04855141, 0.04855141,
       0.04855141, 0.04855141, 0.04855141, 0.04974526, 0.04974526,
       0.04974526, 0.04974526, 0.04974526, 0.05097498, 0.05097498,
       0.05097498, 0.05097498, 0.05097498, 0.05224176, 0.05224176,
       0.05224176, 0.05224176, 0.05224176, 0.05354682, 0.05354682,
       0.05354682, 0.05354682, 0.05354682, 0.05489144, 0.05489144,
       0.05489144, 0.05489144, 0.05489144, 0.05627692, 0.05627692,
       0.05627692, 0.05627692, 0.05627692, 0.05770461, 0.05770461,
       0.05770461, 0.05770461, 0.05770461, 0.05917591, 0.05917591,
       0.05917591, 0.05917591, 0.05917591, 0.06069223, 0.06069223,
       0.06069223, 0.06069223, 0.06069223, 0.06225506, 0.06225506,
       0.06225506, 0.06225506, 0.06225506, 0.06386591, 0.06386591,
       0.06386591, 0.06386591, 0.06386591, 0.06552634, 0.06552634,
       0.06552634, 0.06552634, 0.06552634, 0.06723795, 0.06723795,
       0.06723795, 0.06723795, 0.06723795, 0.06900238, 0.06900238,
       0.06900238, 0.06900238, 0.06900238, 0.07082131, 0.07082131,
       0.07082131, 0.07082131, 0.07082131, 0.07269648, 0.07269648,
       0.07269648, 0.07269648, 0.07269648, 0.07462966, 0.07462966,
       0.07462966, 0.07462966, 0.07462966, 0.07662264, 0.07662264,
       0.07662264, 0.07662264, 0.07662264, 0.07662264, 0.07867727,
       0.07867727, 0.07867727, 0.07867727, 0.07867727, 0.08079545,
       0.08079545, 0.08079545, 0.08079545, 0.08079545, 0.08297909,
       0.08297909, 0.08297909, 0.08297909, 0.08297909, 0.08523013,
       0.08523013, 0.08523013, 0.08523013, 0.08523013, 0.08755057,
       0.08755057, 0.08755057, 0.08755057, 0.08755057, 0.08994242,
       0.08994242, 0.08994242, 0.08994242, 0.08994242, 0.09240769,
       0.09240769, 0.09240769, 0.09240769, 0.09240769, 0.09494846,
       0.09494846, 0.09494846, 0.09494846, 0.09494846, 0.09756677,
       0.09756677, 0.09756677, 0.09756677, 0.09756677, 0.10026471,
       0.10026471, 0.10026471, 0.10026471, 0.10026471, 0.10304434,
```

(continues on next page)

(continued from previous page)

```
       0.10304434, 0.10304434, 0.10304434, 0.10304434, 0.10590775,
       0.10590775, 0.10590775, 0.10590775, 0.10590775, 0.10885697,
       0.10885697, 0.10885697, 0.10885697, 0.10885697, 0.11189405,
       0.11189405, 0.11189405, 0.11189405, 0.11189405, 0.11502099,
       0.11502099, 0.11502099, 0.11502099, 0.11502099, 0.11823975,
       0.11823975, 0.11823975, 0.11823975, 0.11823975, 0.12155222,
       0.12155222, 0.12155222, 0.12155222, 0.12155222, 0.12496024,
       0.12496024, 0.12496024, 0.12496024, 0.12496024, 0.12846555,
       0.12846555, 0.12846555, 0.12846555, 0.12846555, 0.13206981,
       0.13206981, 0.13206981, 0.13206981, 0.13206981, 0.13577455,
       0.13577455, 0.13577455, 0.13577455, 0.13577455, 0.13958115,
       0.13958115, 0.13958115, 0.13958115, 0.13958115, 0.14349086,
       0.14349086, 0.14349086, 0.14349086, 0.14349086, 0.14750474,
       0.14750474, 0.14750474, 0.14750474, 0.14750474, 0.15162364,
       0.15162364, 0.15162364, 0.15162364, 0.15162364, 0.15584819,
       0.15584819, 0.15584819, 0.15584819, 0.15584819, 0.16017878,
       0.16017878, 0.16017878, 0.16017878, 0.16017878, 0.16461548,
       0.16461548, 0.16461548, 0.16461548, 0.16461548, 0.16915809,
       0.16915809, 0.16915809, 0.16915809, 0.16915809, 0.17380603,
       0.17380603, 0.17380603, 0.17380603, 0.17380603, 0.17855838,
       0.17855838, 0.17855838, 0.17855838, 0.17855838, 0.18341379,
       0.18341379, 0.18341379, 0.18341379, 0.18341379, 0.18837046,
       0.18837046, 0.18837046, 0.18837046, 0.18837046, 0.19342613,
       0.19342613, 0.19342613, 0.19342613, 0.19342613, 0.19857801,
       0.19857801, 0.19857801, 0.19857801, 0.19857801, 0.20382276,
       0.20382276, 0.20382276, 0.20382276, 0.20382276, 0.20915647,
       0.20915647, 0.20915647, 0.20915647, 0.20915647, 0.21457458,
       0.21457458, 0.21457458, 0.21457458, 0.21457458, 0.22007189,
       0.22007189, 0.22007189, 0.22007189, 0.22007189, 0.22564251,
       0.22564251, 0.22564251, 0.22564251, 0.22564251, 0.23127983,
       0.23127983, 0.23127983, 0.23127983, 0.23127983, 0.2369765 ,
       0.2369765 , 0.2369765 , 0.2369765 , 0.2369765 , 0.24272438,
       0.24272438, 0.24272438, 0.24272438, 0.24272438, 0.24851456,
       0.24851456, 0.24851456, 0.24851456, 0.24851456, 0.25433732,
       0.25433732, 0.25433732, 0.25433732, 0.25433732, 0.26018213,
       0.26018213, 0.26018213, 0.26018213, 0.26018213, 0.26603763,
       0.26603763, 0.26603763, 0.26603763, 0.26603763, 0.27189168,
       0.27189168, 0.27189168, 0.27189168, 0.27189168, 0.27773131,
       0.27773131, 0.27773131, 0.27773131, 0.27773131, 0.28354282,
       0.28354282, 0.28354282, 0.28354282, 0.28354282, 0.28354282,
       0.28931173, 0.28931173, 0.28931173, 0.28931173, 0.28931173,
       0.29502289, 0.29502289, 0.29502289, 0.29502289, 0.29502289,
       0.30066049, 0.30066049, 0.30066049, 0.30066049, 0.30066049,
       0.30620816, 0.30620816, 0.30620816, 0.30620816, 0.30620816,
       0.311649  , 0.311649  , 0.311649  , 0.311649  , 0.311649  ,
       0.31696571, 0.31696571, 0.31696571, 0.31696571, 0.31696571,
       0.32214066, 0.32214066, 0.32214066, 0.32214066, 0.32214066,
       0.32715601, 0.32715601, 0.32715601, 0.32715601, 0.32715601,
       0.33199381, 0.33199381, 0.33199381, 0.33199381, 0.33199381,
       0.33663615, 0.33663615, 0.33663615, 0.33663615, 0.33663615,
       0.34106527, 0.34106527, 0.34106527, 0.34106527, 0.34106527,
       0.34526371, 0.34526371, 0.34526371, 0.34526371, 0.34526371,
```

(continues on next page)

(continued from previous page)

```
       0.34921444, 0.34921444, 0.34921444, 0.34921444, 0.34921444,
       0.35290103, 0.35290103, 0.35290103, 0.35290103, 0.35290103,
       0.35630778, 0.35630778, 0.35630778, 0.35630778, 0.35630778,
       0.35941985, 0.35941985, 0.35941985, 0.35941985, 0.35941985,
       0.36222343, 0.36222343, 0.36222343, 0.36222343, 0.36222343,
       0.36470585, 0.36470585, 0.36470585, 0.36470585, 0.36470585,
       0.36685574, 0.36685574, 0.36685574, 0.36685574, 0.36685574,
       0.36866309, 0.36866309, 0.36866309, 0.36866309, 0.36866309,
       0.37011942, 0.37011942, 0.37011942, 0.37011942, 0.37011942,
       0.37121781, 0.37121781, 0.37121781, 0.37121781, 0.37121781,
       0.37195301, 0.37195301, 0.37195301, 0.37195301, 0.37195301,
       0.3723215 , 0.3723215 , 0.3723215 , 0.3723215 , 0.3723215 ,
       0.3723215 , 0.3723215 , 0.3723215 , 0.3723215 , 0.3723215 ,
       0.37195301, 0.37195301, 0.37195301, 0.37195301, 0.37195301,
       0.37121781, 0.37121781, 0.37121781, 0.37121781, 0.37121781,
       0.37011942, 0.37011942, 0.37011942, 0.37011942, 0.37011942,
       0.36866309, 0.36866309, 0.36866309, 0.36866309, 0.36866309,
       0.36685574, 0.36685574, 0.36685574, 0.36685574, 0.36685574,
       0.36470585, 0.36470585, 0.36470585, 0.36470585, 0.36470585,
       0.36222343, 0.36222343, 0.36222343, 0.36222343, 0.36222343,
       0.35941985, 0.35941985, 0.35941985, 0.35941985, 0.35941985,
       0.35630778, 0.35630778, 0.35630778, 0.35630778, 0.35630778,
       0.35290103, 0.35290103, 0.35290103, 0.35290103, 0.35290103,
       0.34921444, 0.34921444, 0.34921444, 0.34921444, 0.34921444,
       0.34526371, 0.34526371, 0.34526371, 0.34526371, 0.34526371,
       0.34106527, 0.34106527, 0.34106527, 0.34106527, 0.34106527,
       0.33663615, 0.33663615, 0.33663615, 0.33663615, 0.33663615,
       0.33199381, 0.33199381, 0.33199381, 0.33199381, 0.33199381,
       0.32715601, 0.32715601, 0.32715601, 0.32715601, 0.32715601,
       0.32214066, 0.32214066, 0.32214066, 0.32214066, 0.32214066,
       0.31696571, 0.31696571, 0.31696571, 0.31696571, 0.31696571,
       0.311649  , 0.311649  , 0.311649  , 0.311649  , 0.311649  ,
       0.30620816, 0.30620816, 0.30620816, 0.30620816, 0.30620816,
       0.30066049, 0.30066049, 0.30066049, 0.30066049, 0.30066049,
       0.29502289, 0.29502289, 0.29502289, 0.29502289, 0.29502289,
       0.28931173, 0.28931173, 0.28931173, 0.28931173, 0.28931173,
       0.28354282, 0.28354282, 0.28354282, 0.28354282, 0.28354282,
       0.28354282, 0.27773131, 0.27773131, 0.27773131, 0.27773131,
       0.27773131, 0.27189168, 0.27189168, 0.27189168, 0.27189168,
       0.27189168, 0.26603763, 0.26603763, 0.26603763, 0.26603763,
       0.26603763, 0.26018213, 0.26018213, 0.26018213, 0.26018213,
       0.26018213, 0.25433732, 0.25433732, 0.25433732, 0.25433732,
       0.25433732, 0.24851456, 0.24851456, 0.24851456, 0.24851456,
       0.24851456, 0.24272438, 0.24272438, 0.24272438, 0.24272438,
       0.24272438, 0.2369765 , 0.2369765 , 0.2369765 , 0.2369765 ,
       0.2369765 , 0.23127983, 0.23127983, 0.23127983, 0.23127983,
       0.23127983, 0.22564251, 0.22564251, 0.22564251, 0.22564251,
       0.22564251, 0.22007189, 0.22007189, 0.22007189, 0.22007189,
       0.22007189, 0.21457458, 0.21457458, 0.21457458, 0.21457458,
       0.21457458, 0.20915647, 0.20915647, 0.20915647, 0.20915647,
       0.20915647, 0.20382276, 0.20382276, 0.20382276, 0.20382276,
       0.20382276, 0.19857801, 0.19857801, 0.19857801, 0.19857801,
```

(continues on next page)

(continued from previous page)

```
       0.19857801, 0.19342613, 0.19342613, 0.19342613, 0.19342613,
       0.19342613, 0.18837046, 0.18837046, 0.18837046, 0.18837046,
       0.18837046, 0.18341379, 0.18341379, 0.18341379, 0.18341379,
       0.18341379, 0.17855838, 0.17855838, 0.17855838, 0.17855838,
       0.17855838, 0.17380603, 0.17380603, 0.17380603, 0.17380603,
       0.17380603, 0.16915809, 0.16915809, 0.16915809, 0.16915809,
       0.16915809, 0.16461548, 0.16461548, 0.16461548, 0.16461548,
       0.16461548, 0.16017878, 0.16017878, 0.16017878, 0.16017878,
       0.16017878, 0.15584819, 0.15584819, 0.15584819, 0.15584819,
       0.15584819, 0.15162364, 0.15162364, 0.15162364, 0.15162364,
       0.15162364, 0.14750474, 0.14750474, 0.14750474, 0.14750474,
       0.14750474, 0.14349086, 0.14349086, 0.14349086, 0.14349086,
       0.14349086, 0.13958115, 0.13958115, 0.13958115, 0.13958115,
       0.13958115, 0.13577455, 0.13577455, 0.13577455, 0.13577455,
       0.13577455, 0.13206981, 0.13206981, 0.13206981, 0.13206981,
       0.13206981, 0.12846555, 0.12846555, 0.12846555, 0.12846555,
       0.12846555, 0.12496024, 0.12496024, 0.12496024, 0.12496024,
       0.12496024, 0.12155222, 0.12155222, 0.12155222, 0.12155222,
       0.12155222, 0.11823975, 0.11823975, 0.11823975, 0.11823975,
       0.11823975, 0.11502099, 0.11502099, 0.11502099, 0.11502099,
       0.11502099, 0.11189405, 0.11189405, 0.11189405, 0.11189405,
       0.11189405, 0.10885697, 0.10885697, 0.10885697, 0.10885697,
       0.10885697, 0.10590775, 0.10590775, 0.10590775, 0.10590775,
       0.10590775, 0.10304434, 0.10304434, 0.10304434, 0.10304434,
       0.10304434, 0.10026471, 0.10026471, 0.10026471, 0.10026471,
       0.10026471, 0.09756677, 0.09756677, 0.09756677, 0.09756677,
       0.09756677, 0.09494846, 0.09494846, 0.09494846, 0.09494846,
       0.09494846, 0.09240769, 0.09240769, 0.09240769, 0.09240769,
       0.09240769, 0.08994242, 0.08994242, 0.08994242, 0.08994242,
       0.08994242, 0.08755057, 0.08755057, 0.08755057, 0.08755057,
       0.08755057, 0.08523013, 0.08523013, 0.08523013, 0.08523013,
       0.08523013, 0.08297909, 0.08297909, 0.08297909, 0.08297909,
       0.08297909, 0.08079545, 0.08079545, 0.08079545, 0.08079545,
       0.08079545, 0.07867727, 0.07867727, 0.07867727, 0.07867727,
       0.07867727, 0.07662264, 0.07662264, 0.07662264, 0.07662264,
       0.07662264, 0.07662264, 0.07462966, 0.07462966, 0.07462966,
       0.07462966, 0.07462966, 0.07269648, 0.07269648, 0.07269648,
       0.07269648, 0.07269648, 0.07082131, 0.07082131, 0.07082131,
       0.07082131, 0.07082131, 0.06900238, 0.06900238, 0.06900238,
       0.06900238, 0.06900238, 0.06723795, 0.06723795, 0.06723795,
       0.06723795, 0.06723795, 0.06552634, 0.06552634, 0.06552634,
       0.06552634, 0.06552634, 0.06386591, 0.06386591, 0.06386591,
       0.06386591, 0.06386591, 0.06225506, 0.06225506, 0.06225506,
       0.06225506, 0.06225506, 0.06069223, 0.06069223, 0.06069223,
       0.06069223, 0.06069223, 0.05917591, 0.05917591, 0.05917591,
       0.05917591, 0.05917591, 0.05770461, 0.05770461, 0.05770461,
       0.05770461, 0.05770461, 0.05627692, 0.05627692, 0.05627692,
       0.05627692, 0.05627692, 0.05489144, 0.05489144, 0.05489144,
       0.05489144, 0.05489144, 0.05354682, 0.05354682, 0.05354682,
       0.05354682, 0.05354682, 0.05224176, 0.05224176, 0.05224176,
       0.05224176, 0.05224176, 0.05097498, 0.05097498, 0.05097498,
       0.05097498, 0.05097498, 0.04974526, 0.04974526, 0.04974526,
```

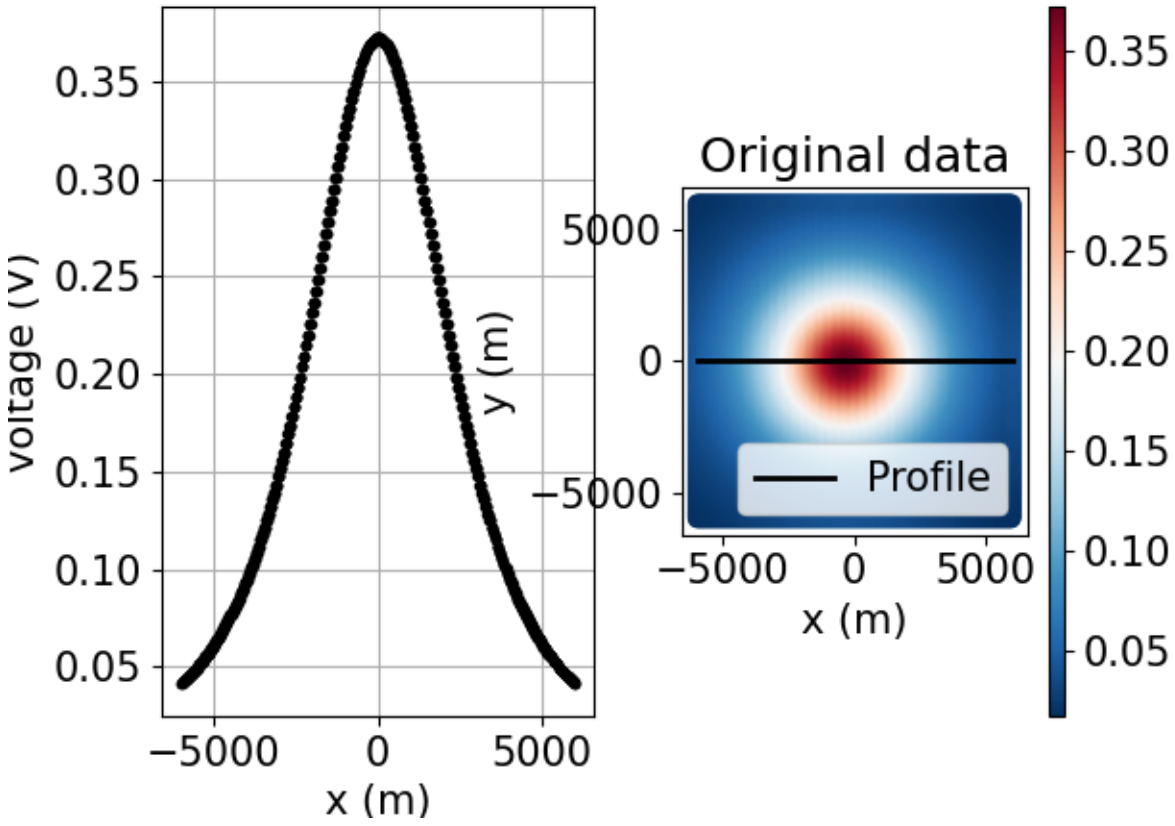
(continues on next page)

(continued from previous page)

```
       0.04974526, 0.04974526, 0.04855141, 0.04855141, 0.04855141,
       0.04855141, 0.04855141, 0.04739227, 0.04739227, 0.04739227,
       0.04739227, 0.04739227, 0.04626672, 0.04626672, 0.04626672,
       0.04626672, 0.04626672, 0.04517369, 0.04517369, 0.04517369,
       0.04517369, 0.04517369, 0.04411214, 0.04411214, 0.04411214,
       0.04411214, 0.04411214, 0.04308104, 0.04308104, 0.04308104,
       0.04308104, 0.04308104, 0.04207942, 0.04207942, 0.04207942,
       0.04207942, 0.04207942, 0.04110633, 0.04110633, 0.04110633]), None, <module
↪'matplotlib.pyplot' from '/home/docs/checkouts/readthedocs.org/user_builds/dexp-
↪imaging/envs/latest/lib/python3.7/site-packages/matplotlib/pyplot.py'>)
```

Pad the edges of grids (if necessary)

```
# xp,yp,U, shape = dEXP.pad_edges(xp,yp,U,shape,pad_type=0) # reflexion=5
# p1 =[min(yp),0]
# p2 =[max(yp),0]

xx, yy, distance, profile, ax, plt = pEXP.plot_line(xp, yp,U,p1,p2, interp=interp,
↪Xaxis=x_axis)
```

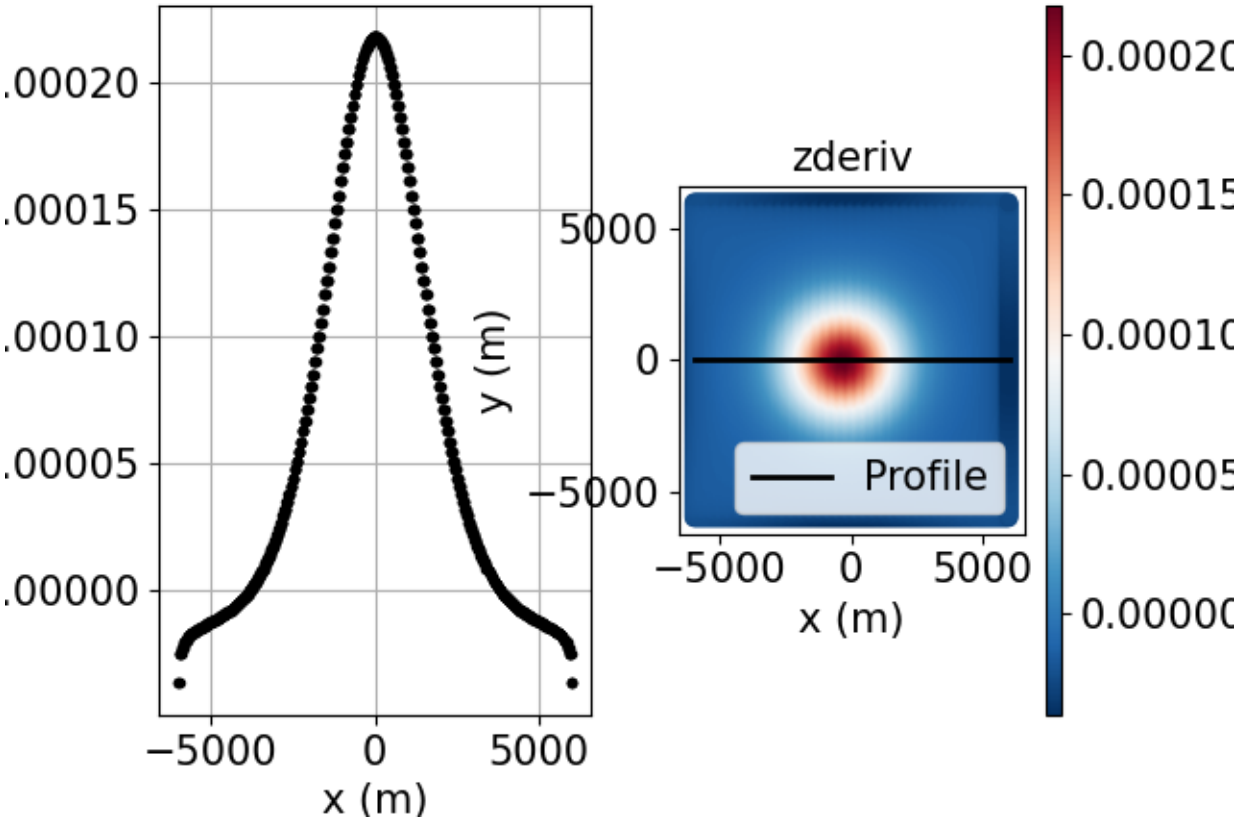



```
# p1 =[0,-6000]
# p2 =[0,6000]
```

(continues on next page)

(continued from previous page)

```
zderiv = transform.derivz(xp, yp, U, shape,order=1)
xx, yy, distance, dz, ax, plt = pEXP.plot_line(xp, yp, zderiv, p1,p2, interp=True, title=
↪'zderiv',Xaxis=x_axis)
```

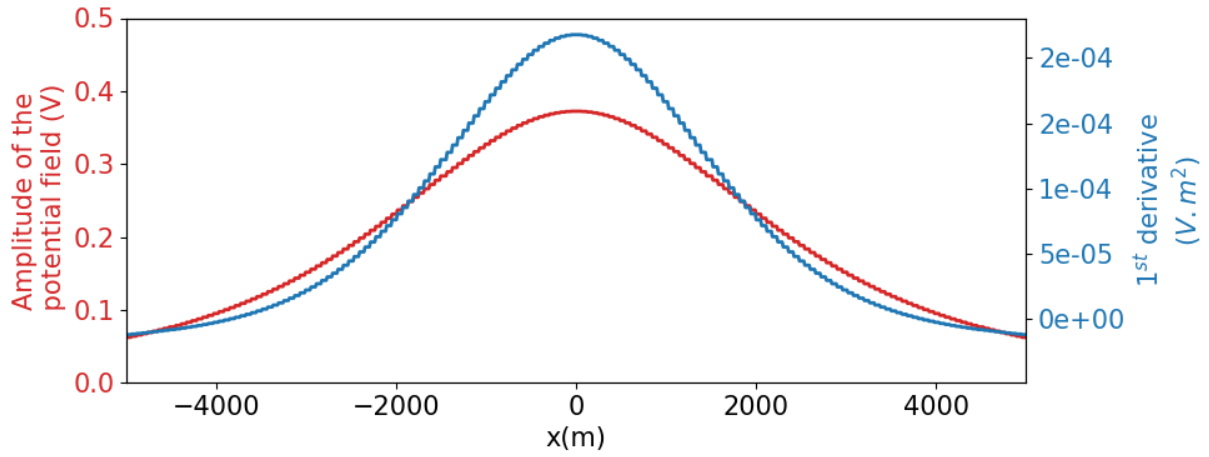



```
# Plot field against its 1st vertical derivative

fig, ax1 = plt.subplots(figsize=(10,4))

color = 'tab:red'
ax1.set_xlabel('x(m)')
ax1.set_ylabel('Amplitude of the\n potential field (V)', color=color)
ax1.plot(xx, profile, color=color, linewidth=2)
ax1.tick_params(axis='y', labelcolor=color)
ax1.set_ylim([0,.5])

ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis

color = 'tab:blue'
ax2.set_ylabel('$1^{st}$ derivative\n($V.m^2$)', color=color)  # we already handled the␣
↪x-label with ax1
ax2.plot(xx, dz, color=color, linewidth=2)
```

(continues on next page)

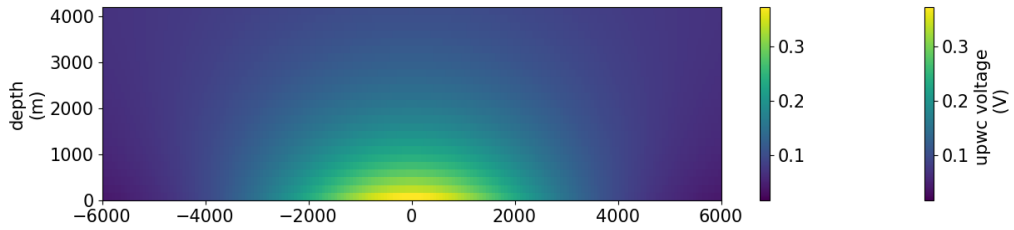
(continued from previous page)

```
ax2.tick_params(axis='y', labelcolor=color)
# ax2.set_ylim([-0.0001,3e-4])
ax2.yaxis.set_major_formatter(ticker.FormatStrFormatter('%1.0e'))
ax2.set_xlim([-5000,5000])

fig.tight_layout()  # otherwise the right y-label is slightly clipped
# ax2.set_aspect(aspect=1e-2)
```



Upward continuation of the field data

```
mesh, label_prop = dEXP.upwc(xp, yp, zp, U, shape,
                  zmin=0, zmax=max_elevation, nlayers=nlay,
                  qorder=qorder)

plt, cmap = pEXP.plot_xy(mesh, label=label_prop,Xaxis=x_axis)
plt.colorbar(cmap)
```

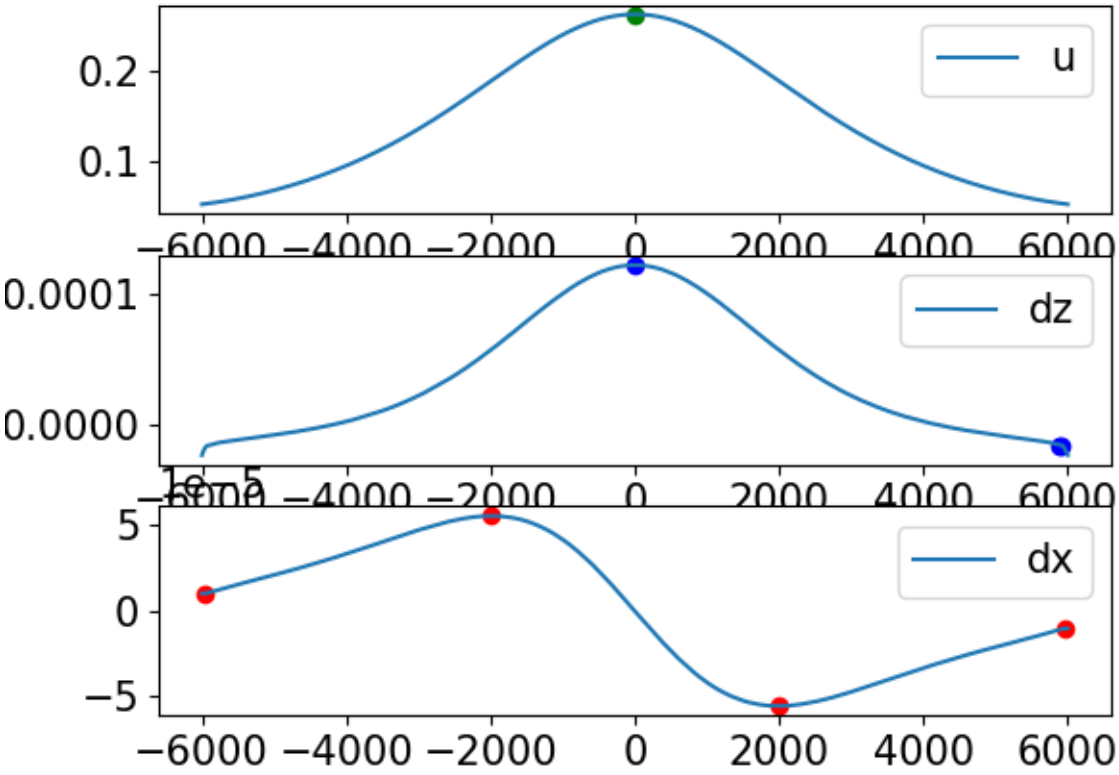



```
/home/docs/checkouts/readthedocs.org/user_builds/dexp-imaging/checkouts/latest/fatiando/
↪gravmag/transform.py:182: UserWarning: Using 'height' <= 0 means downward continuation,
↪ which is known to be unstable.
  warnings.warn("Using 'height' <= 0 means downward continuation, " +

<matplotlib.colorbar.Colorbar object at 0x7f7cecc1ff10>
```

ridges identification dEXP.ridges_minmax_plot(xp, yp, mesh, p1, p2,

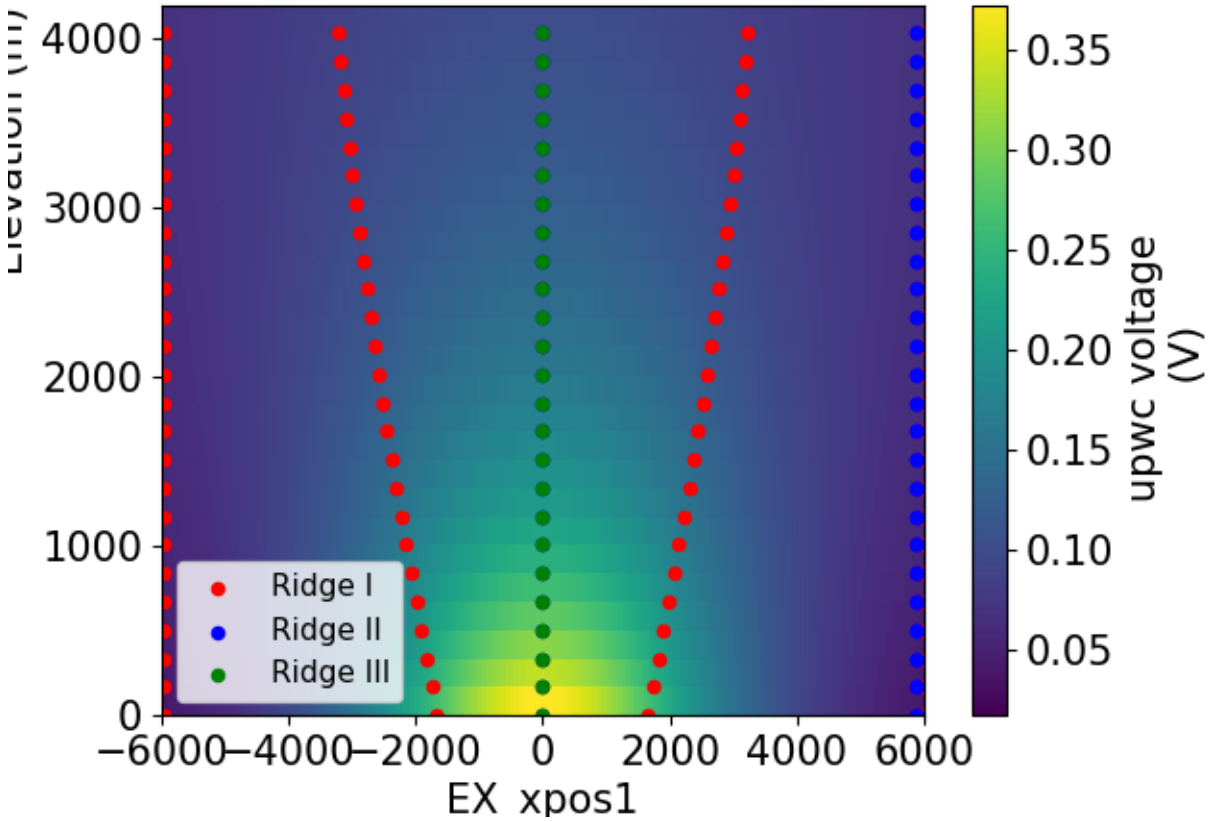
label=label_prop, fix_peak_nb=2, method_peak='find_peaks')

```
# or  find_peaks or peakdet or spline_roots
dfI,dfII, dfIII, _  = dEXP.ridges_minmax(xp, yp, mesh, p1, p2,
                                         label=label_prop,
                                         fix_peak_nb=2,
                                         method_peak='find_peaks',
                                         showfig=True,
                                         Xaxis=x_axis)
```



Plot ridges over continuated section

```
fig = plt.figure()
ax = plt.gca()
pEXP.plot_xy(mesh, label=label_prop, ax=ax, Xaxis=x_axis)
pEXP.plot_ridges_harmonic(dfI,dfII,dfIII,ax=ax)
```




```
<AxesSubplot:xlabel='EX_xpos1', ylabel='Elevation (m)'>
```

Filter ridges regionally constrainsted)

```
dfI_f,dfII_f, dfIII_f = dEXP.filter_ridges(dfI,dfII,dfIII,
                                           minDepth=1000,
                                           maxDepth=3000,
                                           minlength=3,rmvNaN=True)
df_f = dfI_f, dfII_f, dfIII_f
# df_f = dfI, dfII, dfIII
```

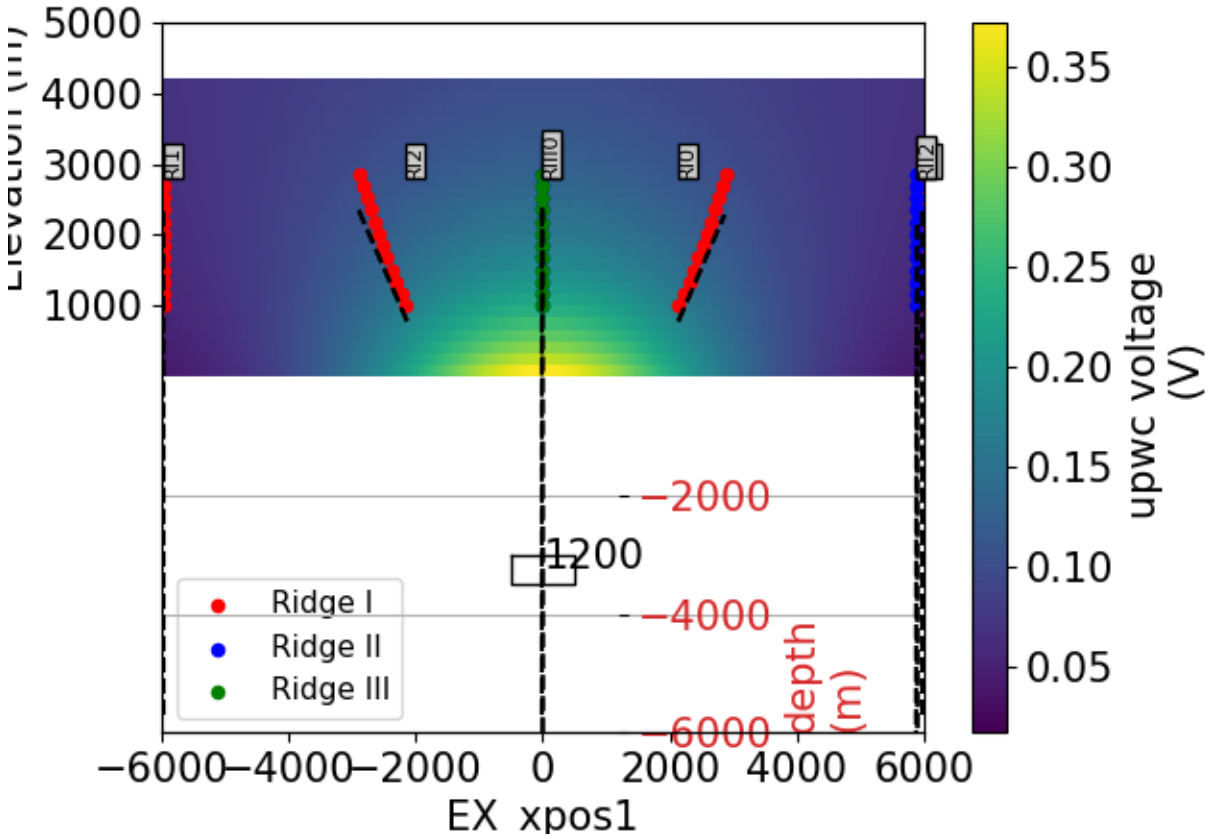
Plot ridges fitted over continuated section

```
fig = plt.figure()
ax = plt.gca()

pEXP.plot_xy(mesh, label=label_prop, ax=ax) #, ldg=)
pEXP.plot_ridges_harmonic(dfI_f,dfII_f,dfIII_f,ax=ax,label=True)

df_fit = dEXP.fit_ridges(df_f, rmvOutliers=True) # fit ridges on filtered data

# pEXP.plot_ridges_sources(df_fit, ax=ax, z_max_source=-max_elevation*1.2,
#                          ridge_type=[0,1,2],ridge_nb=None)
pEXP.plot_ridges_sources(df_fit, ax=ax, z_max_source=-6000,
```

(continues on next page)

(continued from previous page)

```
                          ridge_type=[0,1,2],ridge_nb=None)
square([x1, x2, -z1, -z2])
plt.annotate(dens,[(x1 + x2)/2, -(z1+z2)/2])
```



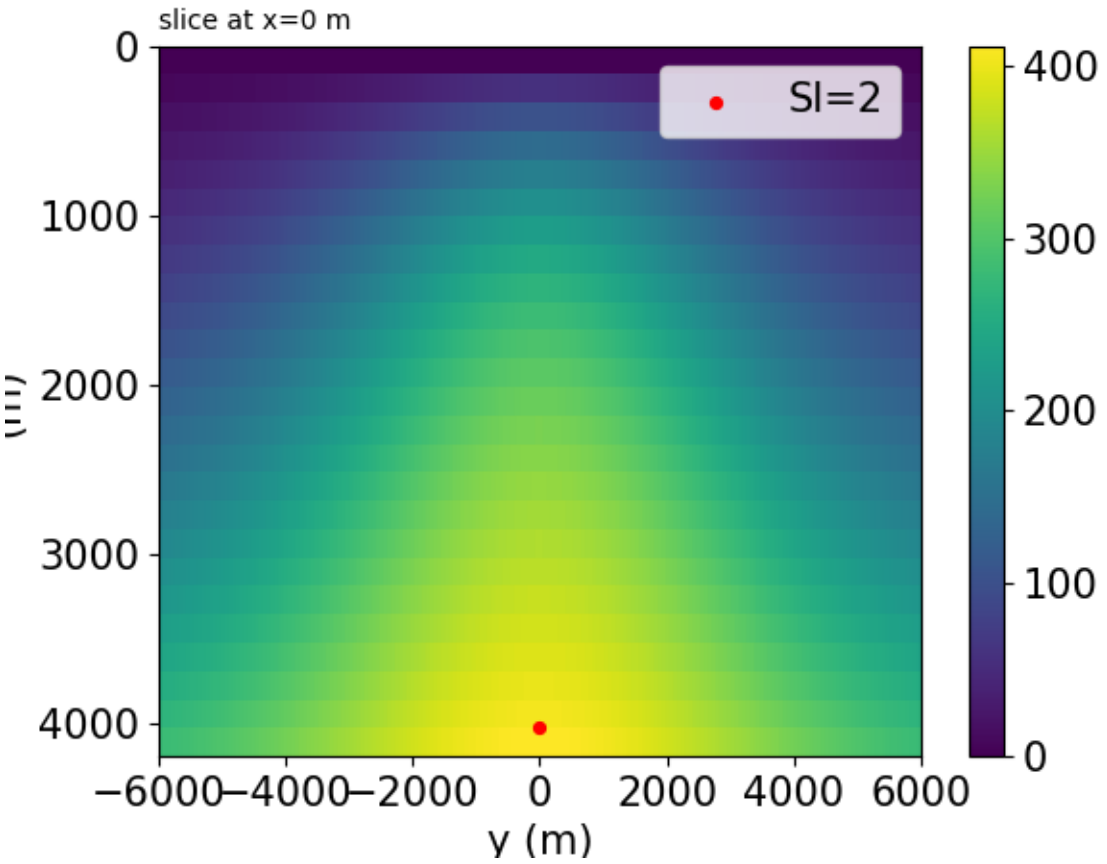

```
/home/docs/checkouts/readthedocs.org/user_builds/dexp-imaging/envs/latest/lib/python3.7/
↪site-packages/scipy/optimize/minpack.py:834: OptimizeWarning: Covariance of the␣
↪parameters could not be estimated
  category=OptimizeWarning)

Text(0.0, -3250.0, '1200')
```

ridges analysis

```
#z0 = -2000
#points, fit, SI, EXTnb = dEXP.scalFUN(dfI_f,EXTnb=[1],z0=z0)
#pEXP.plot_scalFUN(points, fit, z0=z0)


# z0 = -2000
# points, fit, SI, EXTnb = dEXP.scalFUN(dfI_f,EXTnb=[3],z0=z0)
# pEXP.plot_scalFUN(points, fit, z0=z0)
```

ridges analysis

```
mesh_dexp, label_dexp = dEXP.dEXP(xp, yp, zp, U, shape,
                  zmin=0, zmax=max_elevation, nlayers=nlay,
                  qorder=qorder,
                  SI=SI)

fig = plt.figure()
ax = plt.gca()

pEXP.plot_xy(mesh_dexp, label=label_dexp,markerMax=True, SI=SI,
             p1p2=np.array([p1,p2]), ax=ax) #, ldg=)
square([x1, x2, -z1, -z2])
plt.annotate(dens,[(x1 + x2)/2, -(z1+z2)/2])
```




```
/home/docs/checkouts/readthedocs.org/user_builds/dexp-imaging/checkouts/latest/fatiando/
↪gravmag/transform.py:182: UserWarning: Using 'height' <= 0 means downward continuation,
↪ which is known to be unstable.
  warnings.warn("Using 'height' <= 0 means downward continuation, " +
need to rotate first?
dexp_q0doesnt match any label unit
Markermax_z=4032.0
Markermax_x=0.0

Text(0.0, -3250.0, '1200')
```

Total running time of the script: (1 minutes 14.221 seconds)

4.4.2 Magnetic potential field data

Sources properties:

- radius = 1.5e3
- inc = 50
- dec = -30

- Identify 2 depths of sources produces by 2 distincts magnetic sources using the geometrical method
- in prep Identify 2 depths of sources produces by 2 distincts magnetic sources using the dexp ratio method

Magnetic field data analysis using pyDEXP: a 2-sources case

This code shows a step-by-step processing of potential field imaging aiming at giving an estimate of magnetic sources positions and depth using the dEXP tranformation method. dEXP method implementation from Fedi et al. 2012. Calculations used *dEXP*, while plotting use the *plot_dEXP* module.

The model data was created using geometric objects from `fatiando.mesher`. The forward simulation of the data was done using `fatiando.gravmag` module.

Sources locations:

- S_{A} = [10e3,10e3,2e3] # xyz coordinates
- S_{B} = [25e3,10e3,1e3]

Sources properties:

- radius = 1.5e3
- inc = 50
- dec = -30

Note: This is part of a larger project aiming at inverting current sources density (see more at: https://icsd-dev.readthedocs.io/en/latest/)

References

Uieda, L., V. C. Oliveira Jr, and V. C. F. Barbosa (2013), Modeling the Earth with Fatiando a Terra, Proceedings of the 12th Python in Science Conference, pp. 91 - 98.

Uieda, L. (2018). Verde: Processing and gridding spatial data using Green's functions. Journal of Open Source Software, 3(29), 957. doi:10.21105/joss.00957

Fedi, M., and M. Pilkington (2012), Understanding imaging methods for potential field data, Geophysics, 77(1), G13, doi:10.1190/geo2011-0078.1

```
import matplotlib.pyplot as plt
import numpy as np
import lib.dEXP as dEXP
from lib.dEXP import _fit
import lib.plot_dEXP as pEXP
```

(continues on next page)

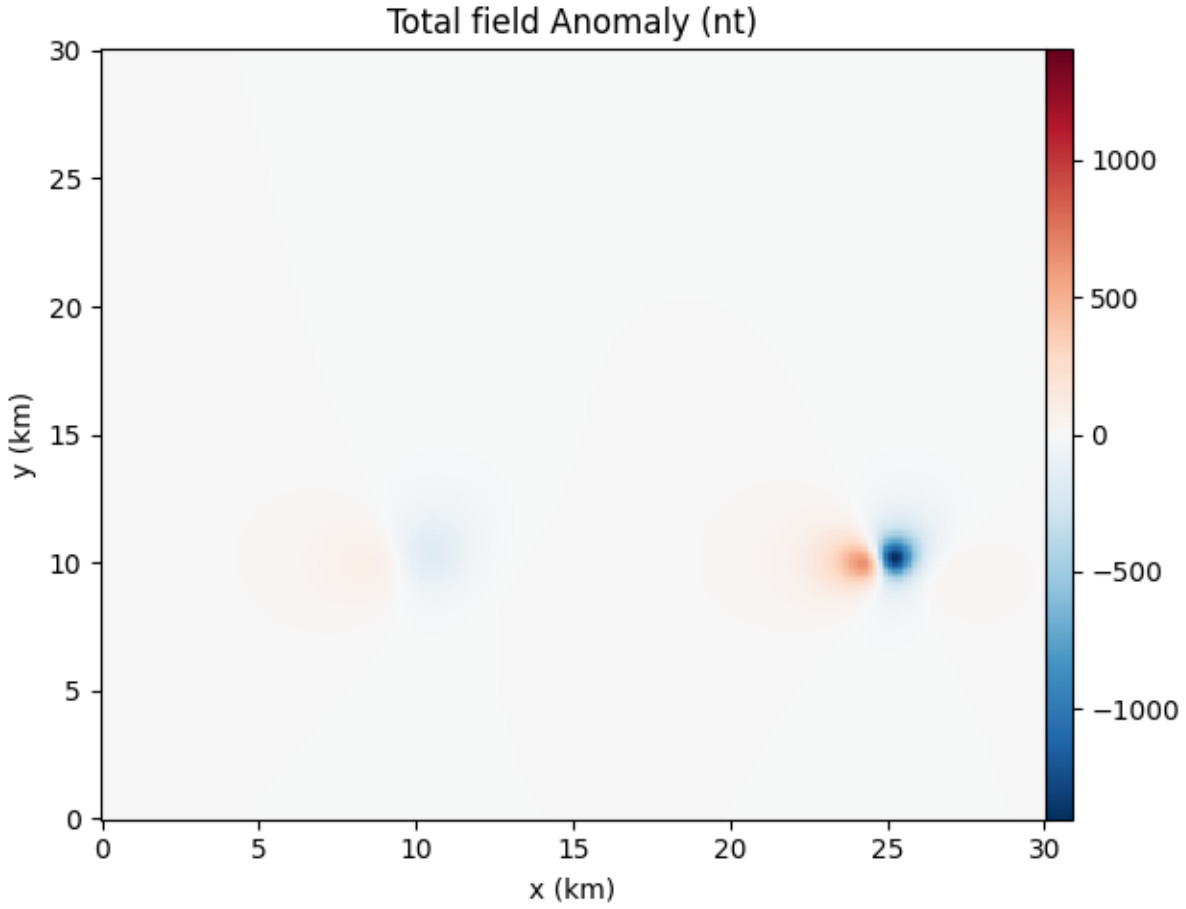
(continued from previous page)

```
import lib.set_parameters as para
import examples.magnetic.fwdmag.fwd_mag_sphere as magfwd
```

```
Failed to import duecredit due to No module named 'duecredit'
```

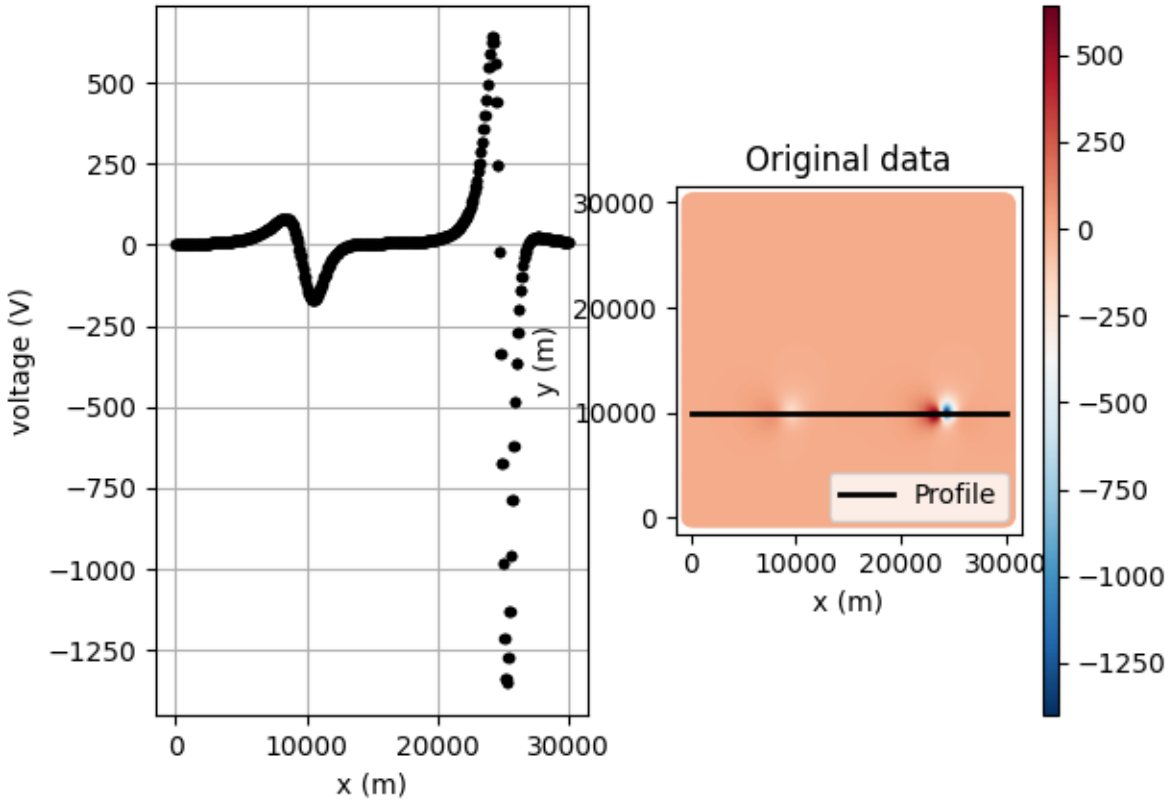
Create a model using geometric objects from fatiando.mesher

```
xp, yp, zp, U, shape, p1, p2, coord= magfwd.load_mag_synthetic()
max_elevation=2*max(coord[:,2])
scaled, SI, zp, qorder, nlay, minAlt_ridge, maxAlt_ridge = para.set_par(shape=shape,max_
↪elevation=max_elevation)
interp = True
x_axis='y'
```



Plot field data over a 2d line crossing the anomalies

```
pEXP.plot_line(xp, yp, U,p1,p2, interp=interp,Xaxis=x_axis)
```




```
(array([    0.        ,    30.03003003,    60.06006006,    90.09009009,
         120.12012012,   150.15015015,   180.18018018,   210.21021021,
         240.24024024,   270.27027027,   300.3003003 ,   330.33033033,
         360.36036036,   390.39039039,   420.42042042,   450.45045045,
         480.48048048,   510.51051051,   540.54054054,   570.57057057,
         600.6006006 ,   630.63063063,   660.66066066,   690.69069069,
         720.72072072,   750.75075075,   780.78078078,   810.81081081,
         840.84084084,   870.87087087,   900.9009009 ,   930.93093093,
         960.96096096,   990.99099099,  1021.02102102,  1051.05105105,
        1081.08108108,  1111.11111111,  1141.14114114,  1171.17117117,
        1201.2012012 ,  1231.23123123,  1261.26126126,  1291.29129129,
        1321.32132132,  1351.35135135,  1381.38138138,  1411.41141141,
        1441.44144144,  1471.47147147,  1501.5015015 ,  1531.53153153,
        1561.56156156,  1591.59159159,  1621.62162162,  1651.65165165,
        1681.68168168,  1711.71171171,  1741.74174174,  1771.77177177,
        1801.8018018 ,  1831.83183183,  1861.86186186,  1891.89189189,
        1921.92192192,  1951.95195195,  1981.98198198,  2012.01201201,
        2042.04204204,  2072.07207207,  2102.1021021 ,  2132.13213213,
        2162.16216216,  2192.19219219,  2222.22222222,  2252.25225225,
        2282.28228228,  2312.31231231,  2342.34234234,  2372.37237237,
        2402.4024024 ,  2432.43243243,  2462.46246246,  2492.49249249,
        2522.52252252,  2552.55255255,  2582.58258258,  2612.61261261,
        2642.64264264,  2672.67267267,  2702.7027027 ,  2732.73273273,
```

(continues on next page)

(continued from previous page)

```
        2762.76276276,  2792.79279279,  2822.82282282,  2852.85285285,
        2882.88288288,  2912.91291291,  2942.94294294,  2972.97297297,
        3003.003003  ,  3033.03303303,  3063.06306306,  3093.09309309,
        3123.12312312,  3153.15315315,  3183.18318318,  3213.21321321,
        3243.24324324,  3273.27327327,  3303.3033033 ,  3333.33333333,
        3363.36336336,  3393.39339339,  3423.42342342,  3453.45345345,
        3483.48348348,  3513.51351351,  3543.54354354,  3573.57357357,
        3603.6036036 ,  3633.63363363,  3663.66366366,  3693.69369369,
        3723.72372372,  3753.75375375,  3783.78378378,  3813.81381381,
        3843.84384384,  3873.87387387,  3903.9039039 ,  3933.93393393,
        3963.96396396,  3993.99399399,  4024.02402402,  4054.05405405,
        4084.08408408,  4114.11411411,  4144.14414414,  4174.17417417,
        4204.2042042 ,  4234.23423423,  4264.26426426,  4294.29429429,
        4324.32432432,  4354.35435435,  4384.38438438,  4414.41441441,
        4444.44444444,  4474.47447447,  4504.5045045 ,  4534.53453453,
        4564.56456456,  4594.59459459,  4624.62462462,  4654.65465465,
        4684.68468468,  4714.71471471,  4744.74474474,  4774.77477477,
        4804.8048048 ,  4834.83483483,  4864.86486486,  4894.89489489,
        4924.92492492,  4954.95495495,  4984.98498498,  5015.01501502,
        5045.04504505,  5075.07507508,  5105.10510511,  5135.13513514,
        5165.16516517,  5195.1951952 ,  5225.22522523,  5255.25525526,
        5285.28528529,  5315.31531532,  5345.34534535,  5375.37537538,
        5405.40540541,  5435.43543544,  5465.46546547,  5495.4954955 ,
        5525.52552553,  5555.55555556,  5585.58558559,  5615.61561562,
        5645.64564565,  5675.67567568,  5705.70570571,  5735.73573574,
        5765.76576577,  5795.7957958 ,  5825.82582583,  5855.85585586,
        5885.88588589,  5915.91591592,  5945.94594595,  5975.97597598,
        6006.00600601,  6036.03603604,  6066.06606607,  6096.0960961 ,
        6126.12612613,  6156.15615616,  6186.18618619,  6216.21621622,
        6246.24624625,  6276.27627628,  6306.30630631,  6336.33633634,
        6366.36636637,  6396.3963964 ,  6426.42642643,  6456.45645646,
        6486.48648649,  6516.51651652,  6546.54654655,  6576.57657658,
        6606.60660661,  6636.63663664,  6666.66666667,  6696.6966967 ,
        6726.72672673,  6756.75675676,  6786.78678679,  6816.81681682,
        6846.84684685,  6876.87687688,  6906.90690691,  6936.93693694,
        6966.96696697,  6996.996997  ,  7027.02702703,  7057.05705706,
        7087.08708709,  7117.11711712,  7147.14714715,  7177.17717718,
        7207.20720721,  7237.23723724,  7267.26726727,  7297.2972973 ,
        7327.32732733,  7357.35735736,  7387.38738739,  7417.41741742,
        7447.44744745,  7477.47747748,  7507.50750751,  7537.53753754,
        7567.56756757,  7597.5975976 ,  7627.62762763,  7657.65765766,
        7687.68768769,  7717.71771772,  7747.74774775,  7777.77777778,
        7807.80780781,  7837.83783784,  7867.86786787,  7897.8978979 ,
        7927.92792793,  7957.95795796,  7987.98798799,  8018.01801802,
        8048.04804805,  8078.07807808,  8108.10810811,  8138.13813814,
        8168.16816817,  8198.1981982 ,  8228.22822823,  8258.25825826,
        8288.28828829,  8318.31831832,  8348.34834835,  8378.37837838,
        8408.40840841,  8438.43843844,  8468.46846847,  8498.4984985 ,
        8528.52852853,  8558.55855856,  8588.58858859,  8618.61861862,
        8648.64864865,  8678.67867868,  8708.70870871,  8738.73873874,
        8768.76876877,  8798.7987988 ,  8828.82882883,  8858.85885886,
        8888.88888889,  8918.91891892,  8948.94894895,  8978.97897898,
```

(continues on next page)

(continued from previous page)

```
 9009.00900901,  9039.03903904,  9069.06906907,  9099.0990991 ,
 9129.12912913,  9159.15915916,  9189.18918919,  9219.21921922,
 9249.24924925,  9279.27927928,  9309.30930931,  9339.33933934,
 9369.36936937,  9399.3993994 ,  9429.42942943,  9459.45945946,
 9489.48948949,  9519.51951952,  9549.54954955,  9579.57957958,
 9609.60960961,  9639.63963964,  9669.66966967,  9699.6996997 ,
 9729.72972973,  9759.75975976,  9789.78978979,  9819.81981982,
 9849.84984985,  9879.87987988,  9909.90990991,  9939.93993994,
 9969.96996997, 10000.        , 10030.03003003, 10060.06006006,
10090.09009009, 10120.12012012, 10150.15015015, 10180.18018018,
10210.21021021, 10240.24024024, 10270.27027027, 10300.3003003 ,
10330.33033033, 10360.36036036, 10390.39039039, 10420.42042042,
10450.45045045, 10480.48048048, 10510.51051051, 10540.54054054,
10570.57057057, 10600.6006006 , 10630.63063063, 10660.66066066,
10690.69069069, 10720.72072072, 10750.75075075, 10780.78078078,
10810.81081081, 10840.84084084, 10870.87087087, 10900.9009009 ,
10930.93093093, 10960.96096096, 10990.99099099, 11021.02102102,
11051.05105105, 11081.08108108, 11111.11111111, 11141.14114114,
11171.17117117, 11201.2012012 , 11231.23123123, 11261.26126126,
11291.29129129, 11321.32132132, 11351.35135135, 11381.38138138,
11411.41141141, 11441.44144144, 11471.47147147, 11501.5015015 ,
11531.53153153, 11561.56156156, 11591.59159159, 11621.62162162,
11651.65165165, 11681.68168168, 11711.71171171, 11741.74174174,
11771.77177177, 11801.8018018 , 11831.83183183, 11861.86186186,
11891.89189189, 11921.92192192, 11951.95195195, 11981.98198198,
12012.01201201, 12042.04204204, 12072.07207207, 12102.1021021 ,
12132.13213213, 12162.16216216, 12192.19219219, 12222.22222222,
12252.25225225, 12282.28228228, 12312.31231231, 12342.34234234,
12372.37237237, 12402.4024024 , 12432.43243243, 12462.46246246,
12492.49249249, 12522.52252252, 12552.55255255, 12582.58258258,
12612.61261261, 12642.64264264, 12672.67267267, 12702.7027027 ,
12732.73273273, 12762.76276276, 12792.79279279, 12822.82282282,
12852.85285285, 12882.88288288, 12912.91291291, 12942.94294294,
12972.97297297, 13003.003003  , 13033.03303303, 13063.06306306,
13093.09309309, 13123.12312312, 13153.15315315, 13183.18318318,
13213.21321321, 13243.24324324, 13273.27327327, 13303.3033033 ,
13333.33333333, 13363.36336336, 13393.39339339, 13423.42342342,
13453.45345345, 13483.48348348, 13513.51351351, 13543.54354354,
13573.57357357, 13603.6036036 , 13633.63363363, 13663.66366366,
13693.69369369, 13723.72372372, 13753.75375375, 13783.78378378,
13813.81381381, 13843.84384384, 13873.87387387, 13903.9039039 ,
13933.93393393, 13963.96396396, 13993.99399399, 14024.02402402,
14054.05405405, 14084.08408408, 14114.11411411, 14144.14414414,
14174.17417417, 14204.2042042 , 14234.23423423, 14264.26426426,
14294.29429429, 14324.32432432, 14354.35435435, 14384.38438438,
14414.41441441, 14444.44444444, 14474.47447447, 14504.5045045 ,
14534.53453453, 14564.56456456, 14594.59459459, 14624.62462462,
14654.65465465, 14684.68468468, 14714.71471471, 14744.74474474,
14774.77477477, 14804.8048048 , 14834.83483483, 14864.86486486,
14894.89489489, 14924.92492492, 14954.95495495, 14984.98498498,
15015.01501502, 15045.04504505, 15075.07507508, 15105.10510511,
15135.13513514, 15165.16516517, 15195.1951952 , 15225.22522523,
```

(continues on next page)

(continued from previous page)

```
       15255.25525526, 15285.28528529, 15315.31531532, 15345.34534535,
       15375.37537538, 15405.40540541, 15435.43543544, 15465.46546547,
       15495.4954955 , 15525.52552553, 15555.55555556, 15585.58558559,
       15615.61561562, 15645.64564565, 15675.67567568, 15705.70570571,
       15735.73573574, 15765.76576577, 15795.7957958 , 15825.82582583,
       15855.85585586, 15885.88588589, 15915.91591592, 15945.94594595,
       15975.97597598, 16006.00600601, 16036.03603604, 16066.06606607,
       16096.0960961 , 16126.12612613, 16156.15615616, 16186.18618619,
       16216.21621622, 16246.24624625, 16276.27627628, 16306.30630631,
       16336.33633634, 16366.36636637, 16396.3963964 , 16426.42642643,
       16456.45645646, 16486.48648649, 16516.51651652, 16546.54654655,
       16576.57657658, 16606.60660661, 16636.63663664, 16666.66666667,
       16696.6966967 , 16726.72672673, 16756.75675676, 16786.78678679,
       16816.81681682, 16846.84684685, 16876.87687688, 16906.90690691,
       16936.93693694, 16966.96696697, 16996.996997  , 17027.02702703,
       17057.05705706, 17087.08708709, 17117.11711712, 17147.14714715,
       17177.17717718, 17207.20720721, 17237.23723724, 17267.26726727,
       17297.2972973 , 17327.32732733, 17357.35735736, 17387.38738739,
       17417.41741742, 17447.44744745, 17477.47747748, 17507.50750751,
       17537.53753754, 17567.56756757, 17597.5975976 , 17627.62762763,
       17657.65765766, 17687.68768769, 17717.71771772, 17747.74774775,
       17777.77777778, 17807.80780781, 17837.83783784, 17867.86786787,
       17897.8978979 , 17927.92792793, 17957.95795796, 17987.98798799,
       18018.01801802, 18048.04804805, 18078.07807808, 18108.10810811,
       18138.13813814, 18168.16816817, 18198.1981982 , 18228.22822823,
       18258.25825826, 18288.28828829, 18318.31831832, 18348.34834835,
       18378.37837838, 18408.40840841, 18438.43843844, 18468.46846847,
       18498.4984985 , 18528.52852853, 18558.55855856, 18588.58858859,
       18618.61861862, 18648.64864865, 18678.67867868, 18708.70870871,
       18738.73873874, 18768.76876877, 18798.7987988 , 18828.82882883,
       18858.85885886, 18888.88888889, 18918.91891892, 18948.94894895,
       18978.97897898, 19009.00900901, 19039.03903904, 19069.06906907,
       19099.0990991 , 19129.12912913, 19159.15915916, 19189.18918919,
       19219.21921922, 19249.24924925, 19279.27927928, 19309.30930931,
       19339.33933934, 19369.36936937, 19399.3993994 , 19429.42942943,
       19459.45945946, 19489.48948949, 19519.51951952, 19549.54954955,
       19579.57957958, 19609.60960961, 19639.63963964, 19669.66966967,
       19699.6996997 , 19729.72972973, 19759.75975976, 19789.78978979,
       19819.81981982, 19849.84984985, 19879.87987988, 19909.90990991,
       19939.93993994, 19969.96996997, 20000.        , 20030.03003003,
       20060.06006006, 20090.09009009, 20120.12012012, 20150.15015015,
       20180.18018018, 20210.21021021, 20240.24024024, 20270.27027027,
       20300.3003003 , 20330.33033033, 20360.36036036, 20390.39039039,
       20420.42042042, 20450.45045045, 20480.48048048, 20510.51051051,
       20540.54054054, 20570.57057057, 20600.6006006 , 20630.63063063,
       20660.66066066, 20690.69069069, 20720.72072072, 20750.75075075,
       20780.78078078, 20810.81081081, 20840.84084084, 20870.87087087,
       20900.9009009 , 20930.93093093, 20960.96096096, 20990.99099099,
       21021.02102102, 21051.05105105, 21081.08108108, 21111.11111111,
       21141.14114114, 21171.17117117, 21201.2012012 , 21231.23123123,
       21261.26126126, 21291.29129129, 21321.32132132, 21351.35135135,
       21381.38138138, 21411.41141141, 21441.44144144, 21471.47147147,
```

(continues on next page)

(continued from previous page)

```
       21501.5015015 , 21531.53153153, 21561.56156156, 21591.59159159,
       21621.62162162, 21651.65165165, 21681.68168168, 21711.71171171,
       21741.74174174, 21771.77177177, 21801.8018018 , 21831.83183183,
       21861.86186186, 21891.89189189, 21921.92192192, 21951.95195195,
       21981.98198198, 22012.01201201, 22042.04204204, 22072.07207207,
       22102.1021021 , 22132.13213213, 22162.16216216, 22192.19219219,
       22222.22222222, 22252.25225225, 22282.28228228, 22312.31231231,
       22342.34234234, 22372.37237237, 22402.4024024 , 22432.43243243,
       22462.46246246, 22492.49249249, 22522.52252252, 22552.55255255,
       22582.58258258, 22612.61261261, 22642.64264264, 22672.67267267,
       22702.7027027 , 22732.73273273, 22762.76276276, 22792.79279279,
       22822.82282282, 22852.85285285, 22882.88288288, 22912.91291291,
       22942.94294294, 22972.97297297, 23003.003003  , 23033.03303303,
       23063.06306306, 23093.09309309, 23123.12312312, 23153.15315315,
       23183.18318318, 23213.21321321, 23243.24324324, 23273.27327327,
       23303.3033033 , 23333.33333333, 23363.36336336, 23393.39339339,
       23423.42342342, 23453.45345345, 23483.48348348, 23513.51351351,
       23543.54354354, 23573.57357357, 23603.6036036 , 23633.63363363,
       23663.66366366, 23693.69369369, 23723.72372372, 23753.75375375,
       23783.78378378, 23813.81381381, 23843.84384384, 23873.87387387,
       23903.9039039 , 23933.93393393, 23963.96396396, 23993.99399399,
       24024.02402402, 24054.05405405, 24084.08408408, 24114.11411411,
       24144.14414414, 24174.17417417, 24204.2042042 , 24234.23423423,
       24264.26426426, 24294.29429429, 24324.32432432, 24354.35435435,
       24384.38438438, 24414.41441441, 24444.44444444, 24474.47447447,
       24504.5045045 , 24534.53453453, 24564.56456456, 24594.59459459,
       24624.62462462, 24654.65465465, 24684.68468468, 24714.71471471,
       24744.74474474, 24774.77477477, 24804.8048048 , 24834.83483483,
       24864.86486486, 24894.89489489, 24924.92492492, 24954.95495495,
       24984.98498498, 25015.01501502, 25045.04504505, 25075.07507508,
       25105.10510511, 25135.13513514, 25165.16516517, 25195.1951952 ,
       25225.22522523, 25255.25525526, 25285.28528529, 25315.31531532,
       25345.34534535, 25375.37537538, 25405.40540541, 25435.43543544,
       25465.46546547, 25495.4954955 , 25525.52552553, 25555.55555556,
       25585.58558559, 25615.61561562, 25645.64564565, 25675.67567568,
       25705.70570571, 25735.73573574, 25765.76576577, 25795.7957958 ,
       25825.82582583, 25855.85585586, 25885.88588589, 25915.91591592,
       25945.94594595, 25975.97597598, 26006.00600601, 26036.03603604,
       26066.06606607, 26096.0960961 , 26126.12612613, 26156.15615616,
       26186.18618619, 26216.21621622, 26246.24624625, 26276.27627628,
       26306.30630631, 26336.33633634, 26366.36636637, 26396.3963964 ,
       26426.42642643, 26456.45645646, 26486.48648649, 26516.51651652,
       26546.54654655, 26576.57657658, 26606.60660661, 26636.63663664,
       26666.66666667, 26696.6966967 , 26726.72672673, 26756.75675676,
       26786.78678679, 26816.81681682, 26846.84684685, 26876.87687688,
       26906.90690691, 26936.93693694, 26966.96696697, 26996.996997  ,
       27027.02702703, 27057.05705706, 27087.08708709, 27117.11711712,
       27147.14714715, 27177.17717718, 27207.20720721, 27237.23723724,
       27267.26726727, 27297.2972973 , 27327.32732733, 27357.35735736,
       27387.38738739, 27417.41741742, 27447.44744745, 27477.47747748,
       27507.50750751, 27537.53753754, 27567.56756757, 27597.5975976 ,
       27627.62762763, 27657.65765766, 27687.68768769, 27717.71771772,
```

(continues on next page)

(continued from previous page)

```
       27747.74774775, 27777.77777778, 27807.80780781, 27837.83783784,
       27867.86786787, 27897.8978979 , 27927.92792793, 27957.95795796,
       27987.98798799, 28018.01801802, 28048.04804805, 28078.07807808,
       28108.10810811, 28138.13813814, 28168.16816817, 28198.1981982 ,
       28228.22822823, 28258.25825826, 28288.28828829, 28318.31831832,
       28348.34834835, 28378.37837838, 28408.40840841, 28438.43843844,
       28468.46846847, 28498.4984985 , 28528.52852853, 28558.55855856,
       28588.58858859, 28618.61861862, 28648.64864865, 28678.67867868,
       28708.70870871, 28738.73873874, 28768.76876877, 28798.7987988 ,
       28828.82882883, 28858.85885886, 28888.88888889, 28918.91891892,
       28948.94894895, 28978.97897898, 29009.00900901, 29039.03903904,
       29069.06906907, 29099.0990991 , 29129.12912913, 29159.15915916,
       29189.18918919, 29219.21921922, 29249.24924925, 29279.27927928,
       29309.30930931, 29339.33933934, 29369.36936937, 29399.3993994 ,
       29429.42942943, 29459.45945946, 29489.48948949, 29519.51951952,
       29549.54954955, 29579.57957958, 29609.60960961, 29639.63963964,
       29669.66966967, 29699.6996997 , 29729.72972973, 29759.75975976,
       29789.78978979, 29819.81981982, 29849.84984985, 29879.87987988,
       29909.90990991, 29939.93993994, 29969.96996997, 30000.        ]), array([10000.,
↪10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
```

(continues on next page)

(continued from previous page)

```
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
       10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
```

(continues on next page)

(continued from previous page)

```
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.,
        10000., 10000., 10000., 10000., 10000., 10000., 10000., 10000.]), array([    0.  ␣
↪      ,    30.03003003,    60.06006006,    90.09009009,
          120.12012012,   150.15015015,   180.18018018,   210.21021021,
          240.24024024,   270.27027027,   300.3003003 ,   330.33033033,
          360.36036036,   390.39039039,   420.42042042,   450.45045045,
          480.48048048,   510.51051051,   540.54054054,   570.57057057,
          600.6006006 ,   630.63063063,   660.66066066,   690.69069069,
          720.72072072,   750.75075075,   780.78078078,   810.81081081,
          840.84084084,   870.87087087,   900.9009009 ,   930.93093093,
          960.96096096,   990.99099099,  1021.02102102,  1051.05105105,
         1081.08108108,  1111.11111111,  1141.14114114,  1171.17117117,
         1201.2012012 ,  1231.23123123,  1261.26126126,  1291.29129129,
         1321.32132132,  1351.35135135,  1381.38138138,  1411.41141141,
```

(continues on next page)

(continued from previous page)

```
       1441.44144144,  1471.47147147,  1501.5015015 ,  1531.53153153,
       1561.56156156,  1591.59159159,  1621.62162162,  1651.65165165,
       1681.68168168,  1711.71171171,  1741.74174174,  1771.77177177,
       1801.8018018 ,  1831.83183183,  1861.86186186,  1891.89189189,
       1921.92192192,  1951.95195195,  1981.98198198,  2012.01201201,
       2042.04204204,  2072.07207207,  2102.1021021 ,  2132.13213213,
       2162.16216216,  2192.19219219,  2222.22222222,  2252.25225225,
       2282.28228228,  2312.31231231,  2342.34234234,  2372.37237237,
       2402.4024024 ,  2432.43243243,  2462.46246246,  2492.49249249,
       2522.52252252,  2552.55255255,  2582.58258258,  2612.61261261,
       2642.64264264,  2672.67267267,  2702.7027027 ,  2732.73273273,
       2762.76276276,  2792.79279279,  2822.82282282,  2852.85285285,
       2882.88288288,  2912.91291291,  2942.94294294,  2972.97297297,
       3003.003003  ,  3033.03303303,  3063.06306306,  3093.09309309,
       3123.12312312,  3153.15315315,  3183.18318318,  3213.21321321,
       3243.24324324,  3273.27327327,  3303.3033033 ,  3333.33333333,
       3363.36336336,  3393.39339339,  3423.42342342,  3453.45345345,
       3483.48348348,  3513.51351351,  3543.54354354,  3573.57357357,
       3603.6036036 ,  3633.63363363,  3663.66366366,  3693.69369369,
       3723.72372372,  3753.75375375,  3783.78378378,  3813.81381381,
       3843.84384384,  3873.87387387,  3903.9039039 ,  3933.93393393,
       3963.96396396,  3993.99399399,  4024.02402402,  4054.05405405,
       4084.08408408,  4114.11411411,  4144.14414414,  4174.17417417,
       4204.2042042 ,  4234.23423423,  4264.26426426,  4294.29429429,
       4324.32432432,  4354.35435435,  4384.38438438,  4414.41441441,
       4444.44444444,  4474.47447447,  4504.5045045 ,  4534.53453453,
       4564.56456456,  4594.59459459,  4624.62462462,  4654.65465465,
       4684.68468468,  4714.71471471,  4744.74474474,  4774.77477477,
       4804.8048048 ,  4834.83483483,  4864.86486486,  4894.89489489,
       4924.92492492,  4954.95495495,  4984.98498498,  5015.01501502,
       5045.04504505,  5075.07507508,  5105.10510511,  5135.13513514,
       5165.16516517,  5195.1951952 ,  5225.22522523,  5255.25525526,
       5285.28528529,  5315.31531532,  5345.34534535,  5375.37537538,
       5405.40540541,  5435.43543544,  5465.46546547,  5495.4954955 ,
       5525.52552553,  5555.55555556,  5585.58558559,  5615.61561562,
       5645.64564565,  5675.67567568,  5705.70570571,  5735.73573574,
       5765.76576577,  5795.7957958 ,  5825.82582583,  5855.85585586,
       5885.88588589,  5915.91591592,  5945.94594595,  5975.97597598,
       6006.00600601,  6036.03603604,  6066.06606607,  6096.0960961 ,
       6126.12612613,  6156.15615616,  6186.18618619,  6216.21621622,
       6246.24624625,  6276.27627628,  6306.30630631,  6336.33633634,
       6366.36636637,  6396.3963964 ,  6426.42642643,  6456.45645646,
       6486.48648649,  6516.51651652,  6546.54654655,  6576.57657658,
       6606.60660661,  6636.63663664,  6666.66666667,  6696.6966967 ,
       6726.72672673,  6756.75675676,  6786.78678679,  6816.81681682,
       6846.84684685,  6876.87687688,  6906.90690691,  6936.93693694,
       6966.96696697,  6996.996997  ,  7027.02702703,  7057.05705706,
       7087.08708709,  7117.11711712,  7147.14714715,  7177.17717718,
       7207.20720721,  7237.23723724,  7267.26726727,  7297.2972973 ,
       7327.32732733,  7357.35735736,  7387.38738739,  7417.41741742,
       7447.44744745,  7477.47747748,  7507.50750751,  7537.53753754,
       7567.56756757,  7597.5975976 ,  7627.62762763,  7657.65765766,
```

(continues on next page)

(continued from previous page)

```
       7687.68768769,  7717.71771772,  7747.74774775,  7777.77777778,
       7807.80780781,  7837.83783784,  7867.86786787,  7897.8978979 ,
       7927.92792793,  7957.95795796,  7987.98798799,  8018.01801802,
       8048.04804805,  8078.07807808,  8108.10810811,  8138.13813814,
       8168.16816817,  8198.1981982 ,  8228.22822823,  8258.25825826,
       8288.28828829,  8318.31831832,  8348.34834835,  8378.37837838,
       8408.40840841,  8438.43843844,  8468.46846847,  8498.4984985 ,
       8528.52852853,  8558.55855856,  8588.58858859,  8618.61861862,
       8648.64864865,  8678.67867868,  8708.70870871,  8738.73873874,
       8768.76876877,  8798.7987988 ,  8828.82882883,  8858.85885886,
       8888.88888889,  8918.91891892,  8948.94894895,  8978.97897898,
       9009.00900901,  9039.03903904,  9069.06906907,  9099.0990991 ,
       9129.12912913,  9159.15915916,  9189.18918919,  9219.21921922,
       9249.24924925,  9279.27927928,  9309.30930931,  9339.33933934,
       9369.36936937,  9399.3993994 ,  9429.42942943,  9459.45945946,
       9489.48948949,  9519.51951952,  9549.54954955,  9579.57957958,
       9609.60960961,  9639.63963964,  9669.66966967,  9699.6996997 ,
       9729.72972973,  9759.75975976,  9789.78978979,  9819.81981982,
       9849.84984985,  9879.87987988,  9909.90990991,  9939.93993994,
       9969.96996997, 10000.        , 10030.03003003, 10060.06006006,
      10090.09009009, 10120.12012012, 10150.15015015, 10180.18018018,
      10210.21021021, 10240.24024024, 10270.27027027, 10300.3003003 ,
      10330.33033033, 10360.36036036, 10390.39039039, 10420.42042042,
      10450.45045045, 10480.48048048, 10510.51051051, 10540.54054054,
      10570.57057057, 10600.6006006 , 10630.63063063, 10660.66066066,
      10690.69069069, 10720.72072072, 10750.75075075, 10780.78078078,
      10810.81081081, 10840.84084084, 10870.87087087, 10900.9009009 ,
      10930.93093093, 10960.96096096, 10990.99099099, 11021.02102102,
      11051.05105105, 11081.08108108, 11111.11111111, 11141.14114114,
      11171.17117117, 11201.2012012 , 11231.23123123, 11261.26126126,
      11291.29129129, 11321.32132132, 11351.35135135, 11381.38138138,
      11411.41141141, 11441.44144144, 11471.47147147, 11501.5015015 ,
      11531.53153153, 11561.56156156, 11591.59159159, 11621.62162162,
      11651.65165165, 11681.68168168, 11711.71171171, 11741.74174174,
      11771.77177177, 11801.8018018 , 11831.83183183, 11861.86186186,
      11891.89189189, 11921.92192192, 11951.95195195, 11981.98198198,
      12012.01201201, 12042.04204204, 12072.07207207, 12102.1021021 ,
      12132.13213213, 12162.16216216, 12192.19219219, 12222.22222222,
      12252.25225225, 12282.28228228, 12312.31231231, 12342.34234234,
      12372.37237237, 12402.4024024 , 12432.43243243, 12462.46246246,
      12492.49249249, 12522.52252252, 12552.55255255, 12582.58258258,
      12612.61261261, 12642.64264264, 12672.67267267, 12702.7027027 ,
      12732.73273273, 12762.76276276, 12792.79279279, 12822.82282282,
      12852.85285285, 12882.88288288, 12912.91291291, 12942.94294294,
      12972.97297297, 13003.003003  , 13033.03303303, 13063.06306306,
      13093.09309309, 13123.12312312, 13153.15315315, 13183.18318318,
      13213.21321321, 13243.24324324, 13273.27327327, 13303.3033033 ,
      13333.33333333, 13363.36336336, 13393.39339339, 13423.42342342,
      13453.45345345, 13483.48348348, 13513.51351351, 13543.54354354,
      13573.57357357, 13603.6036036 , 13633.63363363, 13663.66366366,
      13693.69369369, 13723.72372372, 13753.75375375, 13783.78378378,
      13813.81381381, 13843.84384384, 13873.87387387, 13903.9039039 ,
```

(continues on next page)

(continued from previous page)

```
       13933.93393393, 13963.96396396, 13993.99399399, 14024.02402402,
       14054.05405405, 14084.08408408, 14114.11411411, 14144.14414414,
       14174.17417417, 14204.2042042 , 14234.23423423, 14264.26426426,
       14294.29429429, 14324.32432432, 14354.35435435, 14384.38438438,
       14414.41441441, 14444.44444444, 14474.47447447, 14504.5045045 ,
       14534.53453453, 14564.56456456, 14594.59459459, 14624.62462462,
       14654.65465465, 14684.68468468, 14714.71471471, 14744.74474474,
       14774.77477477, 14804.8048048 , 14834.83483483, 14864.86486486,
       14894.89489489, 14924.92492492, 14954.95495495, 14984.98498498,
       15015.01501502, 15045.04504505, 15075.07507508, 15105.10510511,
       15135.13513514, 15165.16516517, 15195.1951952 , 15225.22522523,
       15255.25525526, 15285.28528529, 15315.31531532, 15345.34534535,
       15375.37537538, 15405.40540541, 15435.43543544, 15465.46546547,
       15495.4954955 , 15525.52552553, 15555.55555556, 15585.58558559,
       15615.61561562, 15645.64564565, 15675.67567568, 15705.70570571,
       15735.73573574, 15765.76576577, 15795.7957958 , 15825.82582583,
       15855.85585586, 15885.88588589, 15915.91591592, 15945.94594595,
       15975.97597598, 16006.00600601, 16036.03603604, 16066.06606607,
       16096.0960961 , 16126.12612613, 16156.15615616, 16186.18618619,
       16216.21621622, 16246.24624625, 16276.27627628, 16306.30630631,
       16336.33633634, 16366.36636637, 16396.3963964 , 16426.42642643,
       16456.45645646, 16486.48648649, 16516.51651652, 16546.54654655,
       16576.57657658, 16606.60660661, 16636.63663664, 16666.66666667,
       16696.6966967 , 16726.72672673, 16756.75675676, 16786.78678679,
       16816.81681682, 16846.84684685, 16876.87687688, 16906.90690691,
       16936.93693694, 16966.96696697, 16996.996997  , 17027.02702703,
       17057.05705706, 17087.08708709, 17117.11711712, 17147.14714715,
       17177.17717718, 17207.20720721, 17237.23723724, 17267.26726727,
       17297.2972973 , 17327.32732733, 17357.35735736, 17387.38738739,
       17417.41741742, 17447.44744745, 17477.47747748, 17507.50750751,
       17537.53753754, 17567.56756757, 17597.5975976 , 17627.62762763,
       17657.65765766, 17687.68768769, 17717.71771772, 17747.74774775,
       17777.77777778, 17807.80780781, 17837.83783784, 17867.86786787,
       17897.8978979 , 17927.92792793, 17957.95795796, 17987.98798799,
       18018.01801802, 18048.04804805, 18078.07807808, 18108.10810811,
       18138.13813814, 18168.16816817, 18198.1981982 , 18228.22822823,
       18258.25825826, 18288.28828829, 18318.31831832, 18348.34834835,
       18378.37837838, 18408.40840841, 18438.43843844, 18468.46846847,
       18498.4984985 , 18528.52852853, 18558.55855856, 18588.58858859,
       18618.61861862, 18648.64864865, 18678.67867868, 18708.70870871,
       18738.73873874, 18768.76876877, 18798.7987988 , 18828.82882883,
       18858.85885886, 18888.88888889, 18918.91891892, 18948.94894895,
       18978.97897898, 19009.00900901, 19039.03903904, 19069.06906907,
       19099.0990991 , 19129.12912913, 19159.15915916, 19189.18918919,
       19219.21921922, 19249.24924925, 19279.27927928, 19309.30930931,
       19339.33933934, 19369.36936937, 19399.3993994 , 19429.42942943,
       19459.45945946, 19489.48948949, 19519.51951952, 19549.54954955,
       19579.57957958, 19609.60960961, 19639.63963964, 19669.66966967,
       19699.6996997 , 19729.72972973, 19759.75975976, 19789.78978979,
       19819.81981982, 19849.84984985, 19879.87987988, 19909.90990991,
       19939.93993994, 19969.96996997, 20000.        , 20030.03003003,
       20060.06006006, 20090.09009009, 20120.12012012, 20150.15015015,
```

(continues on next page)

(continued from previous page)

```
       20180.18018018, 20210.21021021, 20240.24024024, 20270.27027027,
       20300.3003003 , 20330.33033033, 20360.36036036, 20390.39039039,
       20420.42042042, 20450.45045045, 20480.48048048, 20510.51051051,
       20540.54054054, 20570.57057057, 20600.6006006 , 20630.63063063,
       20660.66066066, 20690.69069069, 20720.72072072, 20750.75075075,
       20780.78078078, 20810.81081081, 20840.84084084, 20870.87087087,
       20900.9009009 , 20930.93093093, 20960.96096096, 20990.99099099,
       21021.02102102, 21051.05105105, 21081.08108108, 21111.11111111,
       21141.14114114, 21171.17117117, 21201.2012012 , 21231.23123123,
       21261.26126126, 21291.29129129, 21321.32132132, 21351.35135135,
       21381.38138138, 21411.41141141, 21441.44144144, 21471.47147147,
       21501.5015015 , 21531.53153153, 21561.56156156, 21591.59159159,
       21621.62162162, 21651.65165165, 21681.68168168, 21711.71171171,
       21741.74174174, 21771.77177177, 21801.8018018 , 21831.83183183,
       21861.86186186, 21891.89189189, 21921.92192192, 21951.95195195,
       21981.98198198, 22012.01201201, 22042.04204204, 22072.07207207,
       22102.1021021 , 22132.13213213, 22162.16216216, 22192.19219219,
       22222.22222222, 22252.25225225, 22282.28228228, 22312.31231231,
       22342.34234234, 22372.37237237, 22402.4024024 , 22432.43243243,
       22462.46246246, 22492.49249249, 22522.52252252, 22552.55255255,
       22582.58258258, 22612.61261261, 22642.64264264, 22672.67267267,
       22702.7027027 , 22732.73273273, 22762.76276276, 22792.79279279,
       22822.82282282, 22852.85285285, 22882.88288288, 22912.91291291,
       22942.94294294, 22972.97297297, 23003.003003  , 23033.03303303,
       23063.06306306, 23093.09309309, 23123.12312312, 23153.15315315,
       23183.18318318, 23213.21321321, 23243.24324324, 23273.27327327,
       23303.3033033 , 23333.33333333, 23363.36336336, 23393.39339339,
       23423.42342342, 23453.45345345, 23483.48348348, 23513.51351351,
       23543.54354354, 23573.57357357, 23603.6036036 , 23633.63363363,
       23663.66366366, 23693.69369369, 23723.72372372, 23753.75375375,
       23783.78378378, 23813.81381381, 23843.84384384, 23873.87387387,
       23903.9039039 , 23933.93393393, 23963.96396396, 23993.99399399,
       24024.02402402, 24054.05405405, 24084.08408408, 24114.11411411,
       24144.14414414, 24174.17417417, 24204.2042042 , 24234.23423423,
       24264.26426426, 24294.29429429, 24324.32432432, 24354.35435435,
       24384.38438438, 24414.41441441, 24444.44444444, 24474.47447447,
       24504.5045045 , 24534.53453453, 24564.56456456, 24594.59459459,
       24624.62462462, 24654.65465465, 24684.68468468, 24714.71471471,
       24744.74474474, 24774.77477477, 24804.8048048 , 24834.83483483,
       24864.86486486, 24894.89489489, 24924.92492492, 24954.95495495,
       24984.98498498, 25015.01501502, 25045.04504505, 25075.07507508,
       25105.10510511, 25135.13513514, 25165.16516517, 25195.1951952 ,
       25225.22522523, 25255.25525526, 25285.28528529, 25315.31531532,
       25345.34534535, 25375.37537538, 25405.40540541, 25435.43543544,
       25465.46546547, 25495.4954955 , 25525.52552553, 25555.55555556,
       25585.58558559, 25615.61561562, 25645.64564565, 25675.67567568,
       25705.70570571, 25735.73573574, 25765.76576577, 25795.7957958 ,
       25825.82582583, 25855.85585586, 25885.88588589, 25915.91591592,
       25945.94594595, 25975.97597598, 26006.00600601, 26036.03603604,
       26066.06606607, 26096.0960961 , 26126.12612613, 26156.15615616,
       26186.18618619, 26216.21621622, 26246.24624625, 26276.27627628,
       26306.30630631, 26336.33633634, 26366.36636637, 26396.3963964 ,
```

(continues on next page)

(continued from previous page)

```
       26426.42642643, 26456.45645646, 26486.48648649, 26516.51651652,
       26546.54654655, 26576.57657658, 26606.60660661, 26636.63663664,
       26666.66666667, 26696.6966967 , 26726.72672673, 26756.75675676,
       26786.78678679, 26816.81681682, 26846.84684685, 26876.87687688,
       26906.90690691, 26936.93693694, 26966.96696697, 26996.996997  ,
       27027.02702703, 27057.05705706, 27087.08708709, 27117.11711712,
       27147.14714715, 27177.17717718, 27207.20720721, 27237.23723724,
       27267.26726727, 27297.2972973 , 27327.32732733, 27357.35735736,
       27387.38738739, 27417.41741742, 27447.44744745, 27477.47747748,
       27507.50750751, 27537.53753754, 27567.56756757, 27597.5975976 ,
       27627.62762763, 27657.65765766, 27687.68768769, 27717.71771772,
       27747.74774775, 27777.77777778, 27807.80780781, 27837.83783784,
       27867.86786787, 27897.8978979 , 27927.92792793, 27957.95795796,
       27987.98798799, 28018.01801802, 28048.04804805, 28078.07807808,
       28108.10810811, 28138.13813814, 28168.16816817, 28198.1981982 ,
       28228.22822823, 28258.25825826, 28288.28828829, 28318.31831832,
       28348.34834835, 28378.37837838, 28408.40840841, 28438.43843844,
       28468.46846847, 28498.4984985 , 28528.52852853, 28558.55855856,
       28588.58858859, 28618.61861862, 28648.64864865, 28678.67867868,
       28708.70870871, 28738.73873874, 28768.76876877, 28798.7987988 ,
       28828.82882883, 28858.85885886, 28888.88888889, 28918.91891892,
       28948.94894895, 28978.97897898, 29009.00900901, 29039.03903904,
       29069.06906907, 29099.0990991 , 29129.12912913, 29159.15915916,
       29189.18918919, 29219.21921922, 29249.24924925, 29279.27927928,
       29309.30930931, 29339.33933934, 29369.36936937, 29399.3993994 ,
       29429.42942943, 29459.45945946, 29489.48948949, 29519.51951952,
       29549.54954955, 29579.57957958, 29609.60960961, 29639.63963964,
       29669.66966967, 29699.6996997 , 29729.72972973, 29759.75975976,
       29789.78978979, 29819.81981982, 29849.84984985, 29879.87987988,
       29909.90990991, 29939.93993994, 29969.96996997, 30000.        ]), array([ 2.
↪20795914e+00,  2.20795914e+00,  2.27365315e+00,  2.27365315e+00,
        2.27365315e+00,  2.27365315e+00,  2.34197276e+00,  2.34197276e+00,
        2.34197276e+00,  2.41304847e+00,  2.41304847e+00,  2.41304847e+00,
        2.48701852e+00,  2.48701852e+00,  2.48701852e+00,  2.48701852e+00,
        2.56402934e+00,  2.56402934e+00,  2.56402934e+00,  2.64423616e+00,
        2.64423616e+00,  2.64423616e+00,  2.72780360e+00,  2.72780360e+00,
        2.72780360e+00,  2.72780360e+00,  2.81490629e+00,  2.81490629e+00,
        2.81490629e+00,  2.90572963e+00,  2.90572963e+00,  2.90572963e+00,
        3.00047049e+00,  3.00047049e+00,  3.00047049e+00,  3.00047049e+00,
        3.09933805e+00,  3.09933805e+00,  3.09933805e+00,  3.20255469e+00,
        3.20255469e+00,  3.20255469e+00,  3.31035689e+00,  3.31035689e+00,
        3.31035689e+00,  3.31035689e+00,  3.42299633e+00,  3.42299633e+00,
        3.42299633e+00,  3.54074093e+00,  3.54074093e+00,  3.54074093e+00,
        3.66387606e+00,  3.66387606e+00,  3.66387606e+00,  3.66387606e+00,
        3.79270585e+00,  3.79270585e+00,  3.79270585e+00,  3.92755458e+00,
        3.92755458e+00,  3.92755458e+00,  4.06876818e+00,  4.06876818e+00,
        4.06876818e+00,  4.06876818e+00,  4.21671589e+00,  4.21671589e+00,
        4.21671589e+00,  4.37179201e+00,  4.37179201e+00,  4.37179201e+00,
        4.53441780e+00,  4.53441780e+00,  4.53441780e+00,  4.53441780e+00,
        4.70504361e+00,  4.70504361e+00,  4.70504361e+00,  4.88415111e+00,
        4.88415111e+00,  4.88415111e+00,  5.07225569e+00,  5.07225569e+00,
        5.07225569e+00,  5.07225569e+00,  5.26990915e+00,  5.26990915e+00,
```

(continues on next page)

(continued from previous page)

```
        5.26990915e+00,  5.47770258e+00,  5.47770258e+00,  5.47770258e+00,
        5.69626941e+00,  5.69626941e+00,  5.69626941e+00,  5.69626941e+00,
        5.92628882e+00,  5.92628882e+00,  5.92628882e+00,  6.16848937e+00,
        6.16848937e+00,  6.16848937e+00,  6.42365301e+00,  6.42365301e+00,
        6.42365301e+00,  6.42365301e+00,  6.69261927e+00,  6.69261927e+00,
        6.69261927e+00,  6.97629000e+00,  6.97629000e+00,  6.97629000e+00,
        7.27563436e+00,  7.27563436e+00,  7.27563436e+00,  7.27563436e+00,
        7.59169423e+00,  7.59169423e+00,  7.59169423e+00,  7.92559015e+00,
        7.92559015e+00,  7.92559015e+00,  8.27852761e+00,  8.27852761e+00,
        8.27852761e+00,  8.27852761e+00,  8.65180393e+00,  8.65180393e+00,
        8.65180393e+00,  9.04681564e+00,  9.04681564e+00,  9.04681564e+00,
        9.46506636e+00,  9.46506636e+00,  9.46506636e+00,  9.46506636e+00,
        9.90817534e+00,  9.90817534e+00,  9.90817534e+00,  1.03778865e+01,
        1.03778865e+01,  1.03778865e+01,  1.08760783e+01,  1.08760783e+01,
        1.08760783e+01,  1.08760783e+01,  1.14047735e+01,  1.14047735e+01,
        1.14047735e+01,  1.19661509e+01,  1.19661509e+01,  1.19661509e+01,
        1.19661509e+01,  1.25625556e+01,  1.25625556e+01,  1.25625556e+01,
        1.31965121e+01,  1.31965121e+01,  1.31965121e+01,  1.38707353e+01,
        1.38707353e+01,  1.38707353e+01,  1.38707353e+01,  1.45881440e+01,
        1.45881440e+01,  1.45881440e+01,  1.53518728e+01,  1.53518728e+01,
        1.53518728e+01,  1.61652842e+01,  1.61652842e+01,  1.61652842e+01,
        1.61652842e+01,  1.70319807e+01,  1.70319807e+01,  1.70319807e+01,
        1.79558145e+01,  1.79558145e+01,  1.79558145e+01,  1.89408966e+01,
        1.89408966e+01,  1.89408966e+01,  1.89408966e+01,  1.99916022e+01,
        1.99916022e+01,  1.99916022e+01,  2.11125730e+01,  2.11125730e+01,
        2.11125730e+01,  2.23087139e+01,  2.23087139e+01,  2.23087139e+01,
        2.23087139e+01,  2.35851821e+01,  2.35851821e+01,  2.35851821e+01,
        2.49473683e+01,  2.49473683e+01,  2.49473683e+01,  2.64008634e+01,
        2.64008634e+01,  2.64008634e+01,  2.64008634e+01,  2.79514109e+01,
        2.79514109e+01,  2.79514109e+01,  2.96048378e+01,  2.96048378e+01,
        2.96048378e+01,  3.13669590e+01,  3.13669590e+01,  3.13669590e+01,
        3.13669590e+01,  3.32434484e+01,  3.32434484e+01,  3.32434484e+01,
        3.52396680e+01,  3.52396680e+01,  3.52396680e+01,  3.73604433e+01,
        3.73604433e+01,  3.73604433e+01,  3.73604433e+01,  3.96097741e+01,
        3.96097741e+01,  3.96097741e+01,  4.19904633e+01,  4.19904633e+01,
        4.19904633e+01,  4.45036470e+01,  4.45036470e+01,  4.45036470e+01,
        4.45036470e+01,  4.71482044e+01,  4.71482044e+01,  4.71482044e+01,
        4.99200236e+01,  4.99200236e+01,  4.99200236e+01,  5.28110963e+01,
        5.28110963e+01,  5.28110963e+01,  5.28110963e+01,  5.58084147e+01,
        5.58084147e+01,  5.58084147e+01,  5.88926414e+01,  5.88926414e+01,
        5.88926414e+01,  6.20365279e+01,  6.20365279e+01,  6.20365279e+01,
        6.20365279e+01,  6.52030648e+01,  6.52030648e+01,  6.52030648e+01,
        6.83433607e+01,  6.83433607e+01,  6.83433607e+01,  7.13942705e+01,
        7.13942705e+01,  7.13942705e+01,  7.13942705e+01,  7.42758334e+01,
        7.42758334e+01,  7.42758334e+01,  7.68886339e+01,  7.68886339e+01,
        7.68886339e+01,  7.91112758e+01,  7.91112758e+01,  7.91112758e+01,
        7.91112758e+01,  8.07982594e+01,  8.07982594e+01,  8.07982594e+01,
        8.17786774e+01,  8.17786774e+01,  8.17786774e+01,  8.18562858e+01,
        8.18562858e+01,  8.18562858e+01,  8.18562858e+01,  8.08116536e+01,
        8.08116536e+01,  8.08116536e+01,  7.84072159e+01,  7.84072159e+01,
        7.84072159e+01,  7.84072159e+01,  7.43961070e+01,  7.43961070e+01,
        7.43961070e+01,  6.85355679e+01,  6.85355679e+01,  6.85355679e+01,
```

(continues on next page)

(continued from previous page)

```
 6.06054297e+01,  6.06054297e+01,  6.06054297e+01,  6.06054297e+01,
 5.04315998e+01,  5.04315998e+01,  5.04315998e+01,  3.79135727e+01,
 3.79135727e+01,  3.79135727e+01,  2.30537849e+01,  2.30537849e+01,
 2.30537849e+01,  2.30537849e+01,  5.98528142e+00,  5.98528142e+00,
 5.98528142e+00, -1.30070428e+01, -1.30070428e+01, -1.30070428e+01,
-3.34769957e+01, -3.34769957e+01, -3.34769957e+01, -3.34769957e+01,
-5.48243924e+01, -5.48243924e+01, -5.48243924e+01, -7.63222855e+01,
-7.63222855e+01, -7.63222855e+01, -9.71641484e+01, -9.71641484e+01,
-9.71641484e+01, -9.71641484e+01, -1.16527036e+02, -1.16527036e+02,
-1.16527036e+02, -1.33642861e+02, -1.33642861e+02, -1.33642861e+02,
-1.47867327e+02, -1.47867327e+02, -1.47867327e+02, -1.47867327e+02,
-1.58735754e+02, -1.58735754e+02, -1.58735754e+02, -1.65997224e+02,
-1.65997224e+02, -1.65997224e+02, -1.69622754e+02, -1.69622754e+02,
-1.69622754e+02, -1.69622754e+02, -1.69788221e+02, -1.69788221e+02,
-1.69788221e+02, -1.66837210e+02, -1.66837210e+02, -1.66837210e+02,
-1.61231723e+02, -1.61231723e+02, -1.61231723e+02, -1.61231723e+02,
-1.53499326e+02, -1.53499326e+02, -1.53499326e+02, -1.44184121e+02,
-1.44184121e+02, -1.44184121e+02, -1.33806648e+02, -1.33806648e+02,
-1.33806648e+02, -1.33806648e+02, -1.22835154e+02, -1.22835154e+02,
-1.22835154e+02, -1.11668418e+02, -1.11668418e+02, -1.11668418e+02,
-1.00628692e+02, -1.00628692e+02, -1.00628692e+02, -1.00628692e+02,
-8.99624763e+01, -8.99624763e+01, -8.99624763e+01, -7.98466478e+01,
-7.98466478e+01, -7.98466478e+01, -7.03976788e+01, -7.03976788e+01,
-7.03976788e+01, -7.03976788e+01, -6.16821361e+01, -6.16821361e+01,
-6.16821361e+01, -5.37271629e+01, -5.37271629e+01, -5.37271629e+01,
-4.65301212e+01, -4.65301212e+01, -4.65301212e+01, -4.65301212e+01,
-4.00669558e+01, -4.00669558e+01, -4.00669558e+01, -3.42991206e+01,
-3.42991206e+01, -3.42991206e+01, -2.91790911e+01, -2.91790911e+01,
-2.91790911e+01, -2.91790911e+01, -2.46545948e+01, -2.46545948e+01,
-2.46545948e+01, -2.06717415e+01, -2.06717415e+01, -2.06717415e+01,
-1.71772486e+01, -1.71772486e+01, -1.71772486e+01, -1.71772486e+01,
-1.41199498e+01, -1.41199498e+01, -1.41199498e+01, -1.14517486e+01,
-1.14517486e+01, -1.14517486e+01, -9.12816073e+00, -9.12816073e+00,
-9.12816073e+00, -9.12816073e+00, -7.10855501e+00, -7.10855501e+00,
-7.10855501e+00, -5.35618438e+00, -5.35618438e+00, -5.35618438e+00,
-5.35618438e+00, -3.83807488e+00, -3.83807488e+00, -3.83807488e+00,
-2.52482422e+00, -2.52482422e+00, -2.52482422e+00, -1.39034774e+00,
-1.39034774e+00, -1.39034774e+00, -1.39034774e+00, -4.11598676e-01,
-4.11598676e-01, -4.11598676e-01,  4.31718569e-01,  4.31718569e-01,
 4.31718569e-01,  1.15742968e+00,  1.15742968e+00,  1.15742968e+00,
 1.15742968e+00,  1.78115813e+00,  1.78115813e+00,  1.78115813e+00,
 2.31657472e+00,  2.31657472e+00,  2.31657472e+00,  2.77562658e+00,
 2.77562658e+00,  2.77562658e+00,  2.77562658e+00,  3.16874470e+00,
 3.16874470e+00,  3.16874470e+00,  3.50503048e+00,  3.50503048e+00,
 3.50503048e+00,  3.79242203e+00,  3.79242203e+00,  3.79242203e+00,
 3.79242203e+00,  4.03784168e+00,  4.03784168e+00,  4.03784168e+00,
 4.24732620e+00,  4.24732620e+00,  4.24732620e+00,  4.42614133e+00,
 4.42614133e+00,  4.42614133e+00,  4.42614133e+00,  4.57888223e+00,
 4.57888223e+00,  4.57888223e+00,  4.70956146e+00,  4.70956146e+00,
 4.70956146e+00,  4.82168578e+00,  4.82168578e+00,  4.82168578e+00,
 4.82168578e+00,  4.91832322e+00,  4.91832322e+00,  4.91832322e+00,
 5.00216160e+00,  5.00216160e+00,  5.00216160e+00,  5.07555945e+00,
```

(continues on next page)

(continued from previous page)

```
        5.07555945e+00,  5.07555945e+00,  5.07555945e+00,  5.14059046e+00,
        5.14059046e+00,  5.14059046e+00,  5.19908227e+00,  5.19908227e+00,
        5.19908227e+00,  5.25265015e+00,  5.25265015e+00,  5.25265015e+00,
        5.25265015e+00,  5.30272661e+00,  5.30272661e+00,  5.30272661e+00,
        5.35058709e+00,  5.35058709e+00,  5.35058709e+00,  5.39737255e+00,
        5.39737255e+00,  5.39737255e+00,  5.39737255e+00,  5.44410929e+00,
        5.44410929e+00,  5.44410929e+00,  5.49172630e+00,  5.49172630e+00,
        5.49172630e+00,  5.54107067e+00,  5.54107067e+00,  5.54107067e+00,
        5.54107067e+00,  5.59292112e+00,  5.59292112e+00,  5.59292112e+00,
        5.64800011e+00,  5.64800011e+00,  5.64800011e+00,  5.70698465e+00,
        5.70698465e+00,  5.70698465e+00,  5.70698465e+00,  5.77051600e+00,
        5.77051600e+00,  5.77051600e+00,  5.83920846e+00,  5.83920846e+00,
        5.83920846e+00,  5.91365745e+00,  5.91365745e+00,  5.91365745e+00,
        5.91365745e+00,  5.99444691e+00,  5.99444691e+00,  5.99444691e+00,
        6.08215626e+00,  6.08215626e+00,  6.08215626e+00,  6.17736690e+00,
        6.17736690e+00,  6.17736690e+00,  6.17736690e+00,  6.28066851e+00,
        6.28066851e+00,  6.28066851e+00,  6.39266511e+00,  6.39266511e+00,
        6.39266511e+00,  6.39266511e+00,  6.51398103e+00,  6.51398103e+00,
        6.51398103e+00,  6.64526690e+00,  6.64526690e+00,  6.64526690e+00,
        6.78720566e+00,  6.78720566e+00,  6.78720566e+00,  6.78720566e+00,
        6.94051873e+00,  6.94051873e+00,  6.94051873e+00,  7.10597249e+00,
        7.10597249e+00,  7.10597249e+00,  7.28438499e+00,  7.28438499e+00,
        7.28438499e+00,  7.28438499e+00,  7.47663315e+00,  7.47663315e+00,
        7.47663315e+00,  7.68366037e+00,  7.68366037e+00,  7.68366037e+00,
        7.90648489e+00,  7.90648489e+00,  7.90648489e+00,  7.90648489e+00,
        8.14620876e+00,  8.14620876e+00,  8.14620876e+00,  8.40402769e+00,
        8.40402769e+00,  8.40402769e+00,  8.68124193e+00,  8.68124193e+00,
        8.68124193e+00,  8.68124193e+00,  8.97926822e+00,  8.97926822e+00,
        8.97926822e+00,  9.29965310e+00,  9.29965310e+00,  9.29965310e+00,
        9.64408773e+00,  9.64408773e+00,  9.64408773e+00,  9.64408773e+00,
        1.00144244e+01,  1.00144244e+01,  1.00144244e+01,  1.04126950e+01,
        1.04126950e+01,  1.04126950e+01,  1.08411320e+01,  1.08411320e+01,
        1.08411320e+01,  1.08411320e+01,  1.13021918e+01,  1.13021918e+01,
        1.13021918e+01,  1.17985809e+01,  1.17985809e+01,  1.17985809e+01,
        1.23332863e+01,  1.23332863e+01,  1.23332863e+01,  1.23332863e+01,
        1.29096090e+01,  1.29096090e+01,  1.29096090e+01,  1.35312027e+01,
        1.35312027e+01,  1.35312027e+01,  1.42021174e+01,  1.42021174e+01,
        1.42021174e+01,  1.42021174e+01,  1.49268493e+01,  1.49268493e+01,
        1.49268493e+01,  1.57103982e+01,  1.57103982e+01,  1.57103982e+01,
        1.65583323e+01,  1.65583323e+01,  1.65583323e+01,  1.65583323e+01,
        1.74768630e+01,  1.74768630e+01,  1.74768630e+01,  1.84729314e+01,
        1.84729314e+01,  1.84729314e+01,  1.95543066e+01,  1.95543066e+01,
        1.95543066e+01,  1.95543066e+01,  2.07296999e+01,  2.07296999e+01,
        2.07296999e+01,  2.20088968e+01,  2.20088968e+01,  2.20088968e+01,
        2.34029094e+01,  2.34029094e+01,  2.34029094e+01,  2.34029094e+01,
        2.49241532e+01,  2.49241532e+01,  2.49241532e+01,  2.65866527e+01,
        2.65866527e+01,  2.65866527e+01,  2.84062802e+01,  2.84062802e+01,
        2.84062802e+01,  2.84062802e+01,  3.04010352e+01,  3.04010352e+01,
        3.04010352e+01,  3.25913691e+01,  3.25913691e+01,  3.25913691e+01,
        3.50005662e+01,  3.50005662e+01,  3.50005662e+01,  3.50005662e+01,
        3.76551891e+01,  3.76551891e+01,  3.76551891e+01,  4.05856015e+01,
        4.05856015e+01,  4.05856015e+01,  4.38265802e+01,  4.38265802e+01,
```

(continues on next page)

(continued from previous page)

```
        4.38265802e+01,  4.38265802e+01,  4.74180354e+01,  4.74180354e+01,
        4.74180354e+01,  5.14058545e+01,  5.14058545e+01,  5.14058545e+01,
        5.14058545e+01,  5.58428934e+01,  5.58428934e+01,  5.58428934e+01,
        6.07901374e+01,  6.07901374e+01,  6.07901374e+01,  6.63180592e+01,
        6.63180592e+01,  6.63180592e+01,  6.63180592e+01,  7.25082002e+01,
        7.25082002e+01,  7.25082002e+01,  7.94550032e+01,  7.94550032e+01,
        7.94550032e+01,  8.72679182e+01,  8.72679182e+01,  8.72679182e+01,
        8.72679182e+01,  9.60737931e+01,  9.60737931e+01,  9.60737931e+01,
        1.06019541e+02,  1.06019541e+02,  1.06019541e+02,  1.17275035e+02,
        1.17275035e+02,  1.17275035e+02,  1.17275035e+02,  1.30036118e+02,
        1.30036118e+02,  1.30036118e+02,  1.44527492e+02,  1.44527492e+02,
        1.44527492e+02,  1.61005089e+02,  1.61005089e+02,  1.61005089e+02,
        1.61005089e+02,  1.79757202e+02,  1.79757202e+02,  1.79757202e+02,
        2.01103205e+02,  2.01103205e+02,  2.01103205e+02,  2.25387946e+02,
        2.25387946e+02,  2.25387946e+02,  2.25387946e+02,  2.52968792e+02,
        2.52968792e+02,  2.52968792e+02,  2.84190566e+02,  2.84190566e+02,
        2.84190566e+02,  3.19341217e+02,  3.19341217e+02,  3.19341217e+02,
        3.19341217e+02,  3.58577513e+02,  3.58577513e+02,  3.58577513e+02,
        4.01805582e+02,  4.01805582e+02,  4.01805582e+02,  4.48495839e+02,
        4.48495839e+02,  4.48495839e+02,  4.48495839e+02,  4.97407427e+02,
        4.97407427e+02,  4.97407427e+02,  5.46197634e+02,  5.46197634e+02,
        5.46197634e+02,  5.90905300e+02,  5.90905300e+02,  5.90905300e+02,
        5.90905300e+02,  6.25339716e+02,  6.25339716e+02,  6.25339716e+02,
        6.40501281e+02,  6.40501281e+02,  6.40501281e+02,  6.24330492e+02,
        6.24330492e+02,  6.24330492e+02,  6.24330492e+02,  5.62320349e+02,
        5.62320349e+02,  5.62320349e+02,  4.39728797e+02,  4.39728797e+02,
        4.39728797e+02,  2.46007149e+02,  2.46007149e+02,  2.46007149e+02,
        2.46007149e+02, -1.88201973e+01, -1.88201973e+01, -1.88201973e+01,
       -3.37928210e+02, -3.37928210e+02, -3.37928210e+02, -6.75671481e+02,
       -6.75671481e+02, -6.75671481e+02, -6.75671481e+02, -9.83672910e+02,
       -9.83672910e+02, -9.83672910e+02, -1.21484057e+03, -1.21484057e+03,
       -1.21484057e+03, -1.33905227e+03, -1.33905227e+03, -1.33905227e+03,
       -1.33905227e+03, -1.35182654e+03, -1.35182654e+03, -1.35182654e+03,
       -1.27178763e+03, -1.27178763e+03, -1.27178763e+03, -1.13015834e+03,
       -1.13015834e+03, -1.13015834e+03, -1.13015834e+03, -9.59189960e+02,
       -9.59189960e+02, -9.59189960e+02, -7.84571386e+02, -7.84571386e+02,
       -7.84571386e+02, -7.84571386e+02, -6.22870512e+02, -6.22870512e+02,
       -6.22870512e+02, -4.82446396e+02, -4.82446396e+02, -4.82446396e+02,
       -3.65782904e+02, -3.65782904e+02, -3.65782904e+02, -3.65782904e+02,
       -2.71853063e+02, -2.71853063e+02, -2.71853063e+02, -1.97919489e+02,
       -1.97919489e+02, -1.97919489e+02, -1.40687146e+02, -1.40687146e+02,
       -1.40687146e+02, -1.40687146e+02, -9.69406300e+01, -9.69406300e+01,
       -9.69406300e+01, -6.38390100e+01, -6.38390100e+01, -6.38390100e+01,
       -3.90104305e+01, -3.90104305e+01, -3.90104305e+01, -3.90104305e+01,
       -2.05430365e+01, -2.05430365e+01, -2.05430365e+01, -6.93037105e+00,
       -6.93037105e+00, -6.93037105e+00,  2.99698700e+00,  2.99698700e+00,
        2.99698700e+00,  2.99698700e+00,  1.01379439e+01,  1.01379439e+01,
        1.01379439e+01,  1.51790639e+01,  1.51790639e+01,  1.51790639e+01,
        1.86426080e+01,  1.86426080e+01,  1.86426080e+01,  1.86426080e+01,
        2.09248726e+01,  2.09248726e+01,  2.09248726e+01,  2.23261384e+01,
        2.23261384e+01,  2.23261384e+01,  2.30737484e+01,  2.30737484e+01,
        2.30737484e+01,  2.30737484e+01,  2.33397371e+01,  2.33397371e+01,
```

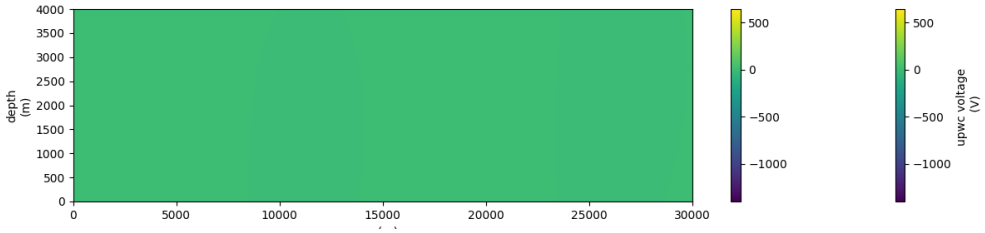
(continues on next page)

(continued from previous page)

```
        2.33397371e+01,  2.32542261e+01,  2.32542261e+01,  2.32542261e+01,
        2.29155721e+01,  2.29155721e+01,  2.29155721e+01,  2.29155721e+01,
        2.23980444e+01,  2.23980444e+01,  2.23980444e+01,  2.17576301e+01,
        2.17576301e+01,  2.17576301e+01,  2.10364286e+01,  2.10364286e+01,
        2.10364286e+01,  2.10364286e+01,  2.02659792e+01,  2.02659792e+01,
        2.02659792e+01,  1.94697863e+01,  1.94697863e+01,  1.94697863e+01,
        1.86652373e+01,  1.86652373e+01,  1.86652373e+01,  1.86652373e+01,
        1.78650630e+01,  1.78650630e+01,  1.78650630e+01,  1.70784506e+01,
        1.70784506e+01,  1.70784506e+01,  1.63118942e+01,  1.63118942e+01,
        1.63118942e+01,  1.63118942e+01,  1.55698447e+01,  1.55698447e+01,
        1.55698447e+01,  1.48552075e+01,  1.48552075e+01,  1.48552075e+01,
        1.41697246e+01,  1.41697246e+01,  1.41697246e+01,  1.41697246e+01,
        1.35142665e+01,  1.35142665e+01,  1.35142665e+01,  1.28890576e+01,
        1.28890576e+01,  1.28890576e+01,  1.22938474e+01,  1.22938474e+01,
        1.22938474e+01,  1.22938474e+01,  1.17280430e+01,  1.17280430e+01,
        1.17280430e+01,  1.11908100e+01,  1.11908100e+01,  1.11908100e+01,
        1.06811488e+01,  1.06811488e+01,  1.06811488e+01,  1.06811488e+01,
        1.01979540e+01,  1.01979540e+01,  1.01979540e+01,  9.74005780e+00,
        9.74005780e+00,  9.74005780e+00,  9.30626363e+00,  9.30626363e+00,
        9.30626363e+00,  9.30626363e+00,  8.89537079e+00,  8.89537079e+00]), None,
↪<module 'matplotlib.pyplot' from '/home/docs/checkouts/readthedocs.org/user_builds/
↪dexp-imaging/envs/latest/lib/python3.7/site-packages/matplotlib/pyplot.py'>)
```

Upward continuation of the field data with discretisation in altitude controlled by the number of layers (nlay) and the maximum elevation desired (max_elevation)

```
mesh, label_prop = dEXP.upwc(xp, yp, zp, U, shape,
                 zmin=0, zmax=max_elevation, nlayers=nlay,
                 qorder=qorder)

plt, cmap = pEXP.plot_xy(mesh, label=label_prop,Xaxis=x_axis)
plt.colorbar(cmap)
```




```
/home/docs/checkouts/readthedocs.org/user_builds/dexp-imaging/checkouts/latest/fatiando/
↪gravmag/transform.py:182: UserWarning: Using 'height' <= 0 means downward continuation,
↪ which is known to be unstable.
  warnings.warn("Using 'height' <= 0 means downward continuation, " +

<matplotlib.colorbar.Colorbar object at 0x7f7ced004e90>
```

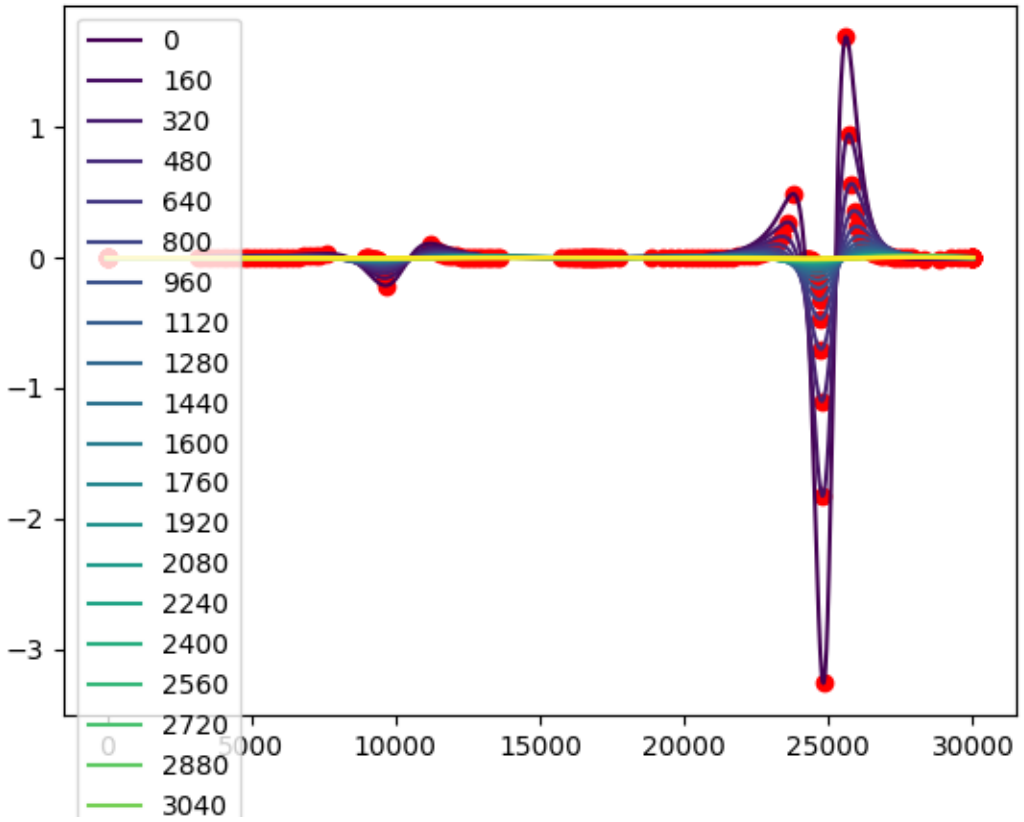
Ridges identification: plot all extremas obtained via find_peaks function (numpy) for a given altitude

```
dEXP.ridges_minmax_plot(xp, yp, mesh, p1, p2,
                                    label=label_prop,
```

(continues on next page)

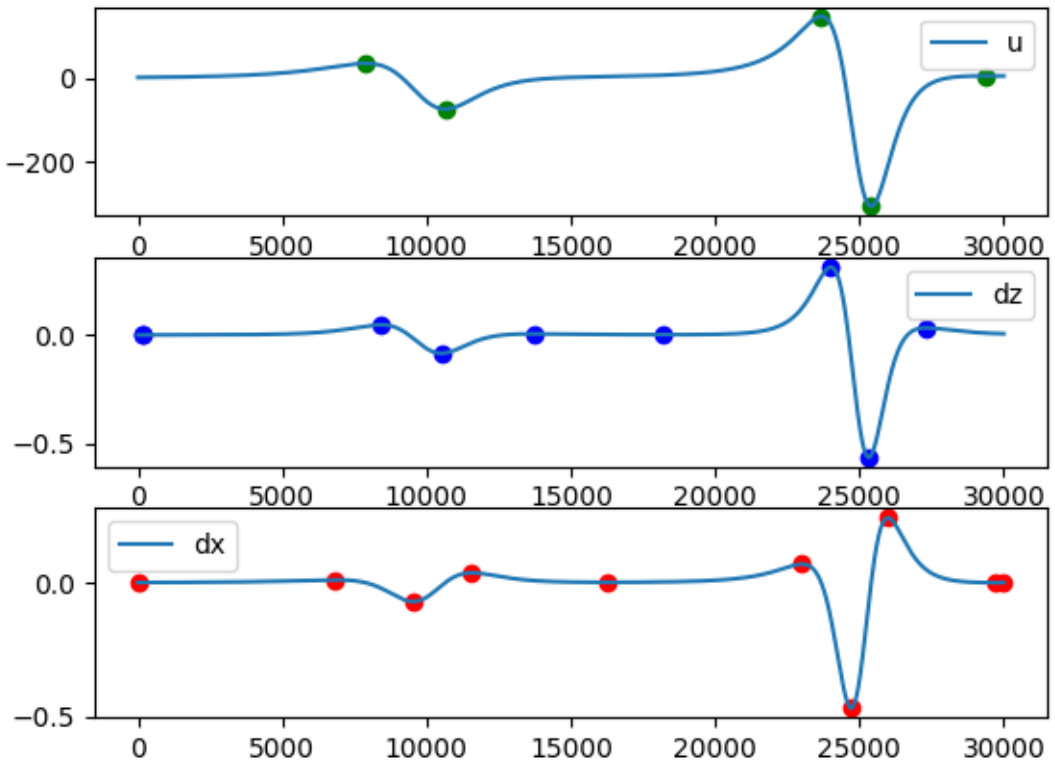
(continued from previous page)

```
                                        method_peak='find_peaks',
                                        fix_peak_nb=5,
                                        Xaxis=x_axis,
                                        showfig=True)
```



Ridges identification at all levels: plot extremas obtained via find_peaks function (numpy) for all 3 types of extremas familly RI, RII and RIII

```
dfI,dfII, dfIII, _ = dEXP.ridges_minmax(xp, yp, mesh, p1, p2,
                                        label=label_prop,
                                        method_peak='find_peaks',
                                        fix_peak_nb=5,
                                        Xaxis=x_axis,
                                        showfig=True)
```



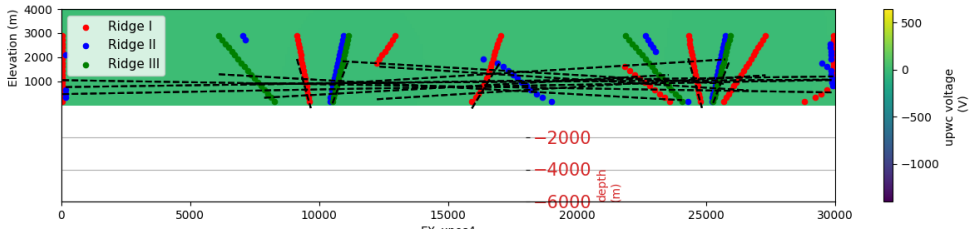
filter ridges using a minimum length criterium and and filter for a specific range of altitude

```
dfI_f,dfII_f, dfIII_f = dEXP.filter_ridges(dfI,dfII,dfIII,
                                          1,maxAlt_ridge,
                                          minlength=8,rmvNaN=True)
df_f = dfI_f, dfII_f, dfIII_f
```

```
NaN or Inf detected - trying to remove
```

plot filtered ridges fitted over continuated section

```
fig, ax = plt.subplots(figsize=(15,3))
pEXP.plot_xy(mesh, label=label_prop, ax=ax) #, ldg=)
pEXP.plot_ridges_harmonic(dfI_f,dfII_f,dfIII_f,ax=ax,label=False)
df_fit = dEXP.fit_ridges(df_f) # fit ridges on filtered data

pEXP.plot_ridges_sources(df_fit, ax=ax, z_max_source=-max_elevation*1.2,
                         ridge_type=[0,1,2],ridge_nb=None)
```




```
/home/docs/checkouts/readthedocs.org/user_builds/dexp-imaging/envs/latest/lib/python3.7/
↪site-packages/scipy/optimize/minpack.py:834: OptimizeWarning: Covariance of the␣
↪parameters could not be estimated
  category=OptimizeWarning)

<AxesSubplot:xlabel='y (m)', ylabel='depth\n(m)'>
```

Total running time of the script: (3 minutes 48.834 seconds)

Magnetic field data analysis using pyDEXP: a 2-sources case

This code shows a step-by-step processing of potential field imaging aiming at giving an estimate of magnetic sources positions and depth using the dEXP tranformation method. dEXP method implementation from Fedi et al. 2012. Calculations used *dEXP*, while plotting use the *plot_dEXP* module.

The model data was created using geometric objects from `fatiando.mesher`. The forward simulation of the data was done using `fatiando.gravmag` module.

Sources locations:

- S_{A} = [10e3,10e3,2e3] # xyz coordinates
- S_{B} = [25e3,10e3,1e3]

Sources properties:

- radius = 1.5e3
- inc = 50
- dec = -30

Note: This is part of a larger project aiming at inverting current sources density (see more at: https://icsd-dev.readthedocs.io/en/latest/)

References

Uieda, L., V. C. Oliveira Jr, and V. C. F. Barbosa (2013), Modeling the Earth with Fatiando a Terra, Proceedings of the 12th Python in Science Conference, pp. 91 - 98.

Uieda, L. (2018). Verde: Processing and gridding spatial data using Green's functions. Journal of Open Source Software, 3(29), 957. doi:10.21105/joss.00957

Fedi, M., and M. Pilkington (2012), Understanding imaging methods for potential field data, Geophysics, 77(1), G13, doi:10.1190/geo2011-0078.1

```
import matplotlib.pyplot as plt
import numpy as np
```

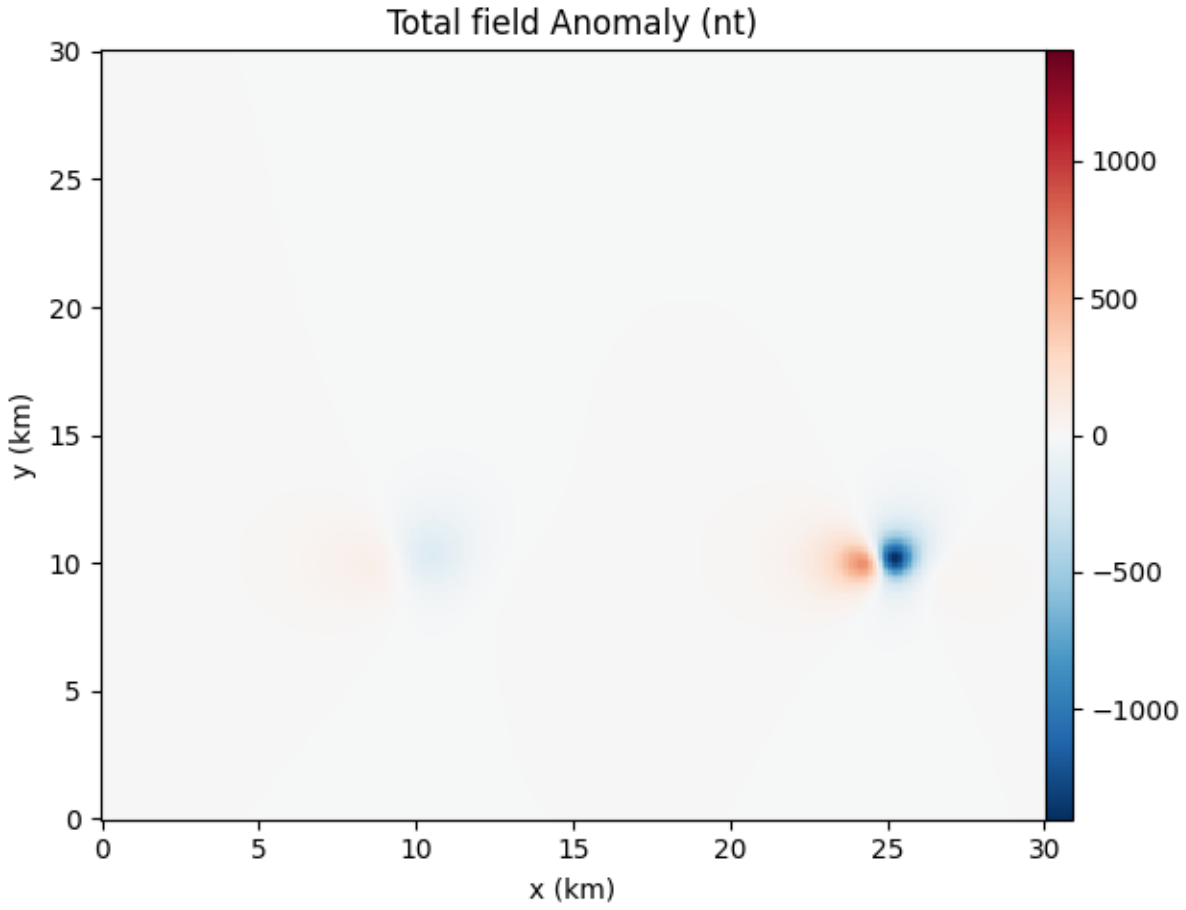
(continues on next page)

(continued from previous page)

```
import lib.dEXP as dEXP
from lib.dEXP import _fit
import lib.plot_dEXP as pEXP
import lib.set_parameters as para
import examples.magnetic.fwdmag.fwd_mag_sphere as magfwd
```

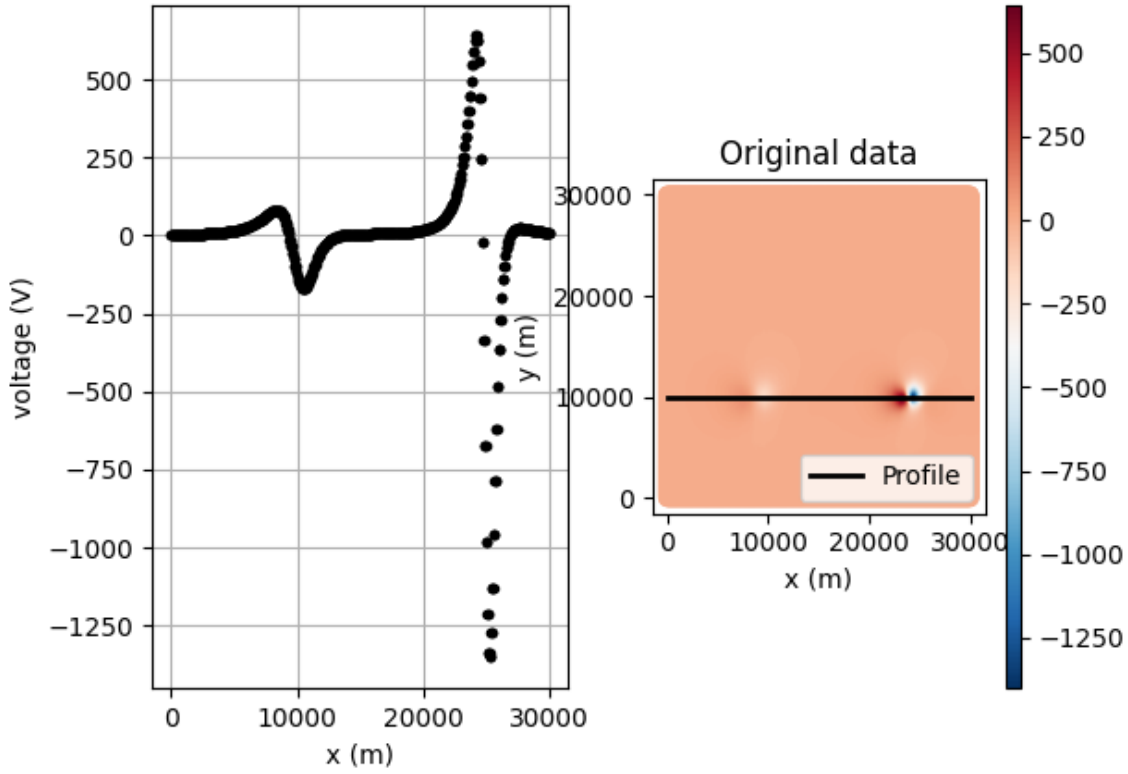
Create a model using geometric objects from fatiando.mesher

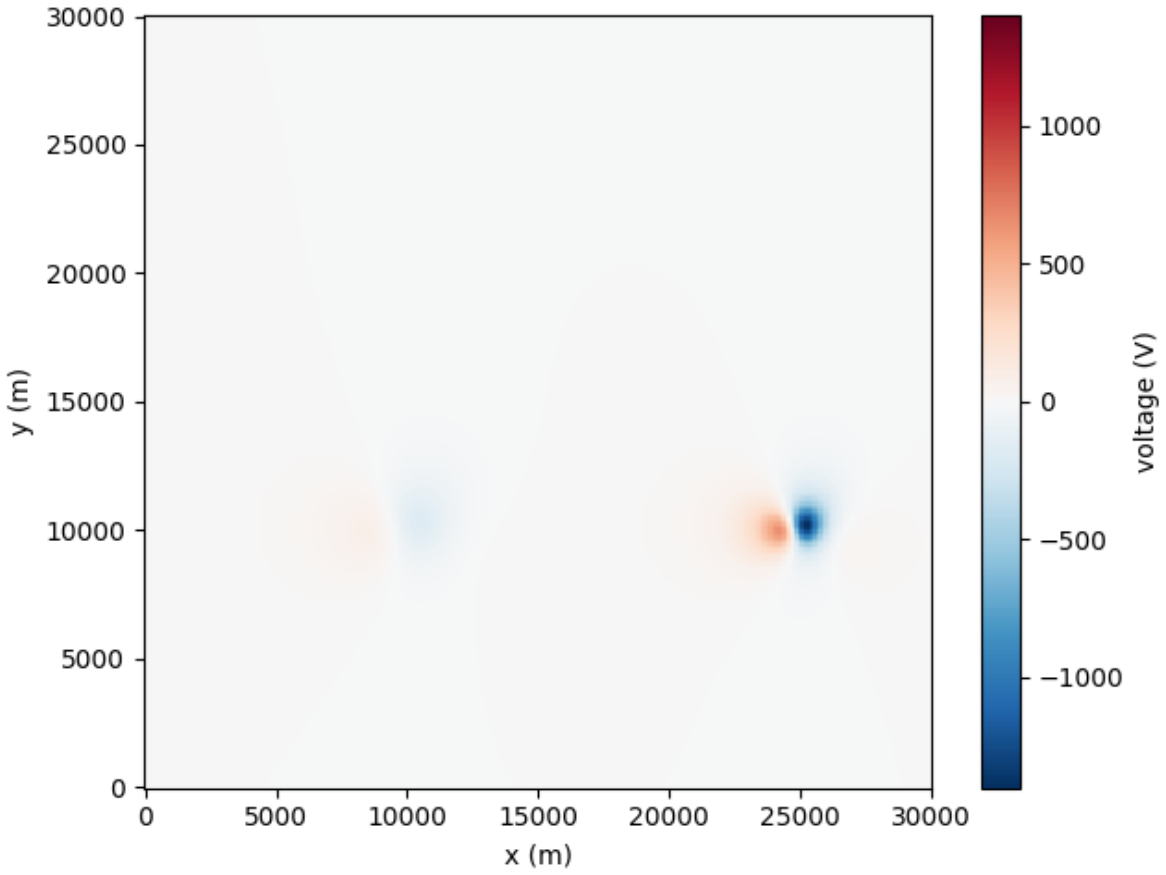
```
xp, yp, zp, U, shape, p1, p2, coord= magfwd.load_mag_synthetic()
max_elevation=2*max(coord[:,2])
scaled, SI, zp, qorder, nlay, minAlt_ridge, maxAlt_ridge = para.set_par(shape=shape,max_
↪elevation=max_elevation)
interp = True
x_axis= 'y'
qorder = 0
```



Plot field data over a 2d line crossing the anomalies

```
pEXP.plot_line(xp, yp, U,p1,p2, interp=True, Xaxis=x_axis)
pEXP.plot_field(xp, yp, U, shape)
```

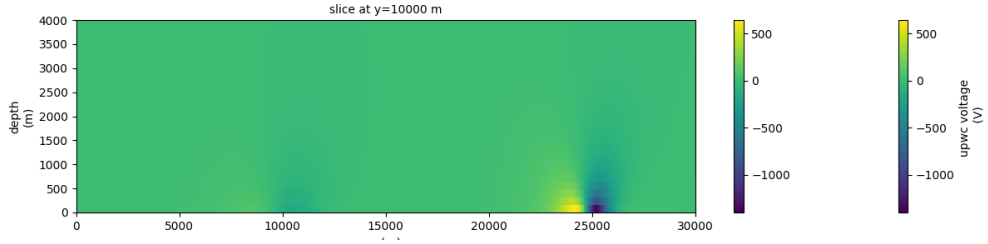

voltage (V)
500
250
0
−250
−500
−750
−1000
−1250
0
10000
20000
30000
x (m)
Original data
y (m)
30000
20000
10000
0
Profile
0
10000
20000
30000
x (m)
500
250
0
−250
−500
−750
−1000
−1250




```
(<AxesSubplot:xlabel='x (m)', ylabel='y (m)'>, <module 'matplotlib.pyplot' from '/home/
↪docs/checkouts/readthedocs.org/user_builds/dexp-imaging/envs/latest/lib/python3.7/site-
↪packages/matplotlib/pyplot.py'>)
```

Upward continuation of the field data with discretisation in altitude controlled by the number of layers (nlay) and the maximum elevation desired (max_elevation)

```
mesh, label_prop = dEXP.upwc(xp, yp, zp, U, shape,
                 zmin=0, zmax=max_elevation, nlayers=nlay,
                 qorder=qorder)

# plt, cmap = pEXP.plot_xy(mesh, label=label_prop)
# plt.colorbar(cmap)
plt, cmap = pEXP.plot_xy(mesh, label=label_prop, Xaxis=x_axis, p1p2=np.array([p1, p2]))
plt.colorbar(cmap)
```

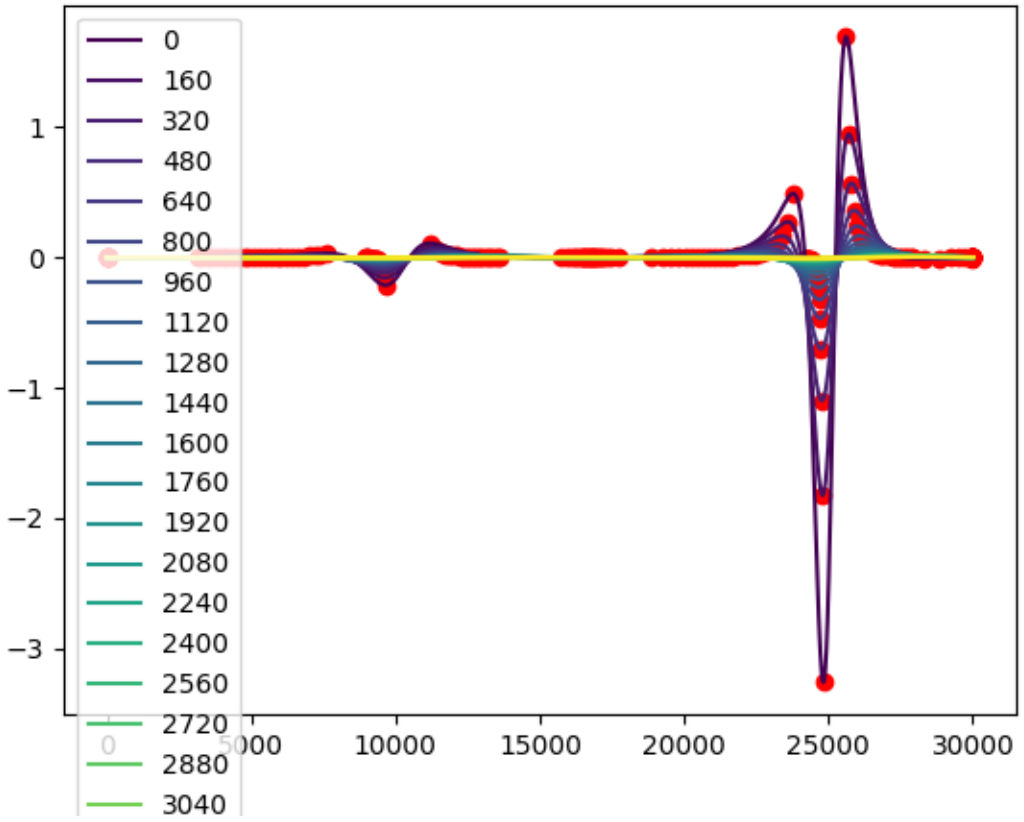


```
/home/docs/checkouts/readthedocs.org/user_builds/dexp-imaging/checkouts/latest/fatiando/
↪gravmag/transform.py:182: UserWarning: Using 'height' <= 0 means downward continuation,
↪ which is known to be unstable.
  warnings.warn("Using 'height' <= 0 means downward continuation, " +
need to rotate first?

<matplotlib.colorbar.Colorbar object at 0x7f7cef4d3b50>
```

Ridges identification: plot all extremas obtained via find_peaks function (numpy) for a given altitude dEXP.ridges_minmax_plot(xp, yp, mesh, p1, p2,

label=label_prop, method_peak='find_peaks')

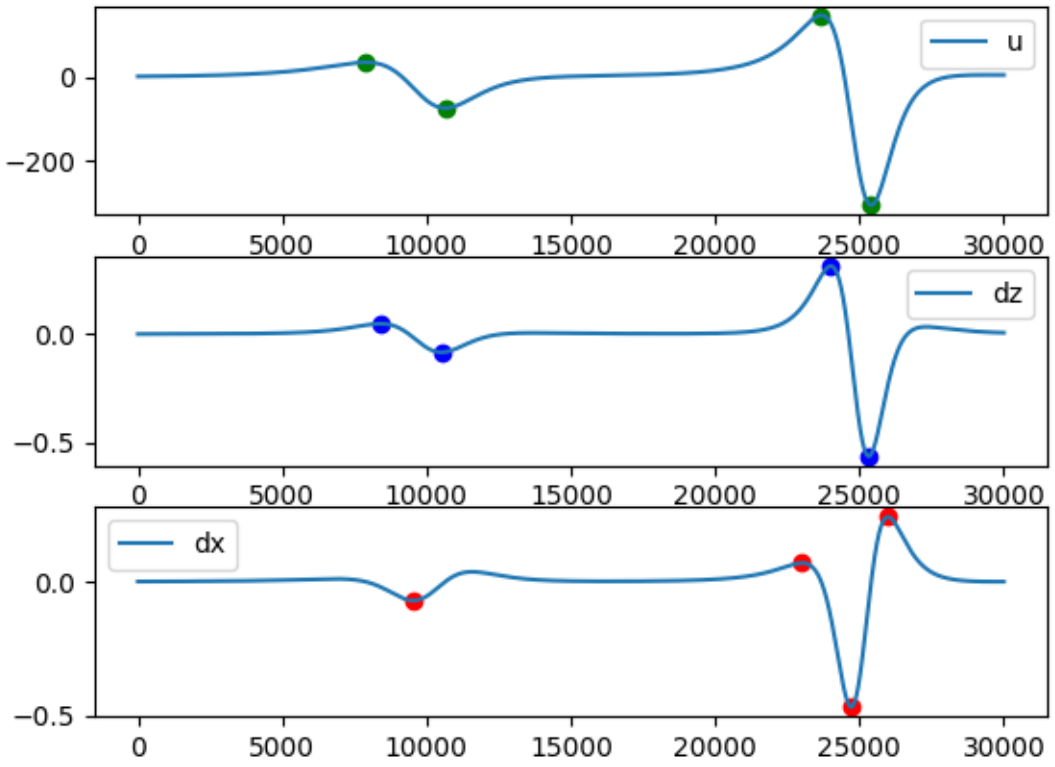
```
dEXP.ridges_minmax_plot(xp, yp, mesh, p1, p2,
                                      label=label_prop,
                                      method_peak='find_peaks',
                                      showfig=True,
                                      interp=True,smooth=True,
                                      Xaxis=x_axis)
```



Ridges identification at all levels: plot extremas obtained via find_peaks function (numpy) for all 3 types of extremas familly RI, RII and RIII dfI,dfII, dfIII = dEXP.ridges_minmax(xp, yp, mesh, p1, p2,

label=label_prop, method_peak='find_peaks')

```
D  = dEXP.ridges_minmax(xp, yp, mesh, p1, p2,
                                      label=label_prop,
                                      method_peak='find_peaks',
                                      qorder=qorder,
                                      interp=True,smooth=True,
                                      fix_peak_nb=2,
                                      returnAmp=True,
                                      showfig=True,
                                      Xaxis=x_axis)
dfI, dfII, dfIII =  D[0:3]
hI, hII, hIII  = D[3:6]
H  = D[3:6]
```



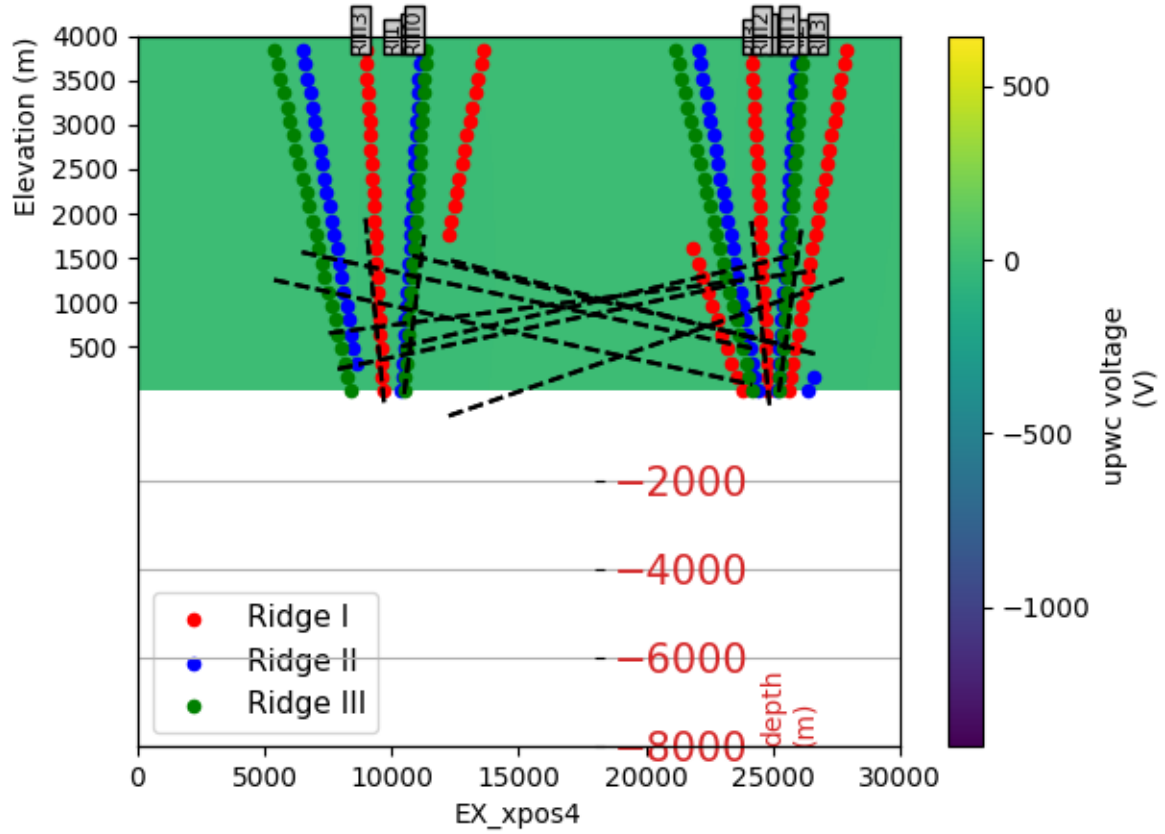
plot filtered ridges fitted over continuated section

```
fig = plt.figure()
ax = plt.gca()

pEXP.plot_xy(mesh, label=label_prop, ax=ax) #, ldg=)
pEXP.plot_ridges_harmonic(dfI,dfII,dfIII,ax=ax,label=True)

df_fit = dEXP.fit_ridges(D[0:3], rmvOutliers=True) # fit ridges on filtered data
```

(continues on next page)

(continued from previous page)

```
pEXP.plot_ridges_sources(df_fit, ax=ax, z_max_source=-max_elevation*2,
                         ridge_type=[0,1,2],ridge_nb=None)
```




```
/home/docs/checkouts/readthedocs.org/user_builds/dexp-imaging/envs/latest/lib/python3.7/
↪site-packages/scipy/optimize/minpack.py:834: OptimizeWarning: Covariance of the␣
↪parameters could not be estimated
  category=OptimizeWarning)

<AxesSubplot:xlabel='y (m)', ylabel='depth\n(m)'>
```

filter ridges using a minimum length criterium and and filter for a specific range of altitude dfI_f,dfII_f, dfIII_f = dEXP.filter_ridges(dfI,dfII,dfIII,

1,maxAlt_ridge, minlength=8,rmvNaN=True)

df_f = dfI_f, dfII_f, dfIII_f

```
D_f = dEXP.filter_ridges(dfI,dfII,dfIII,
                         minDepth=0,
                         maxDepth=10000,
                         minlength=8,
                         rmvNaN=True,
                         heights=[hI, hII, hIII])
```

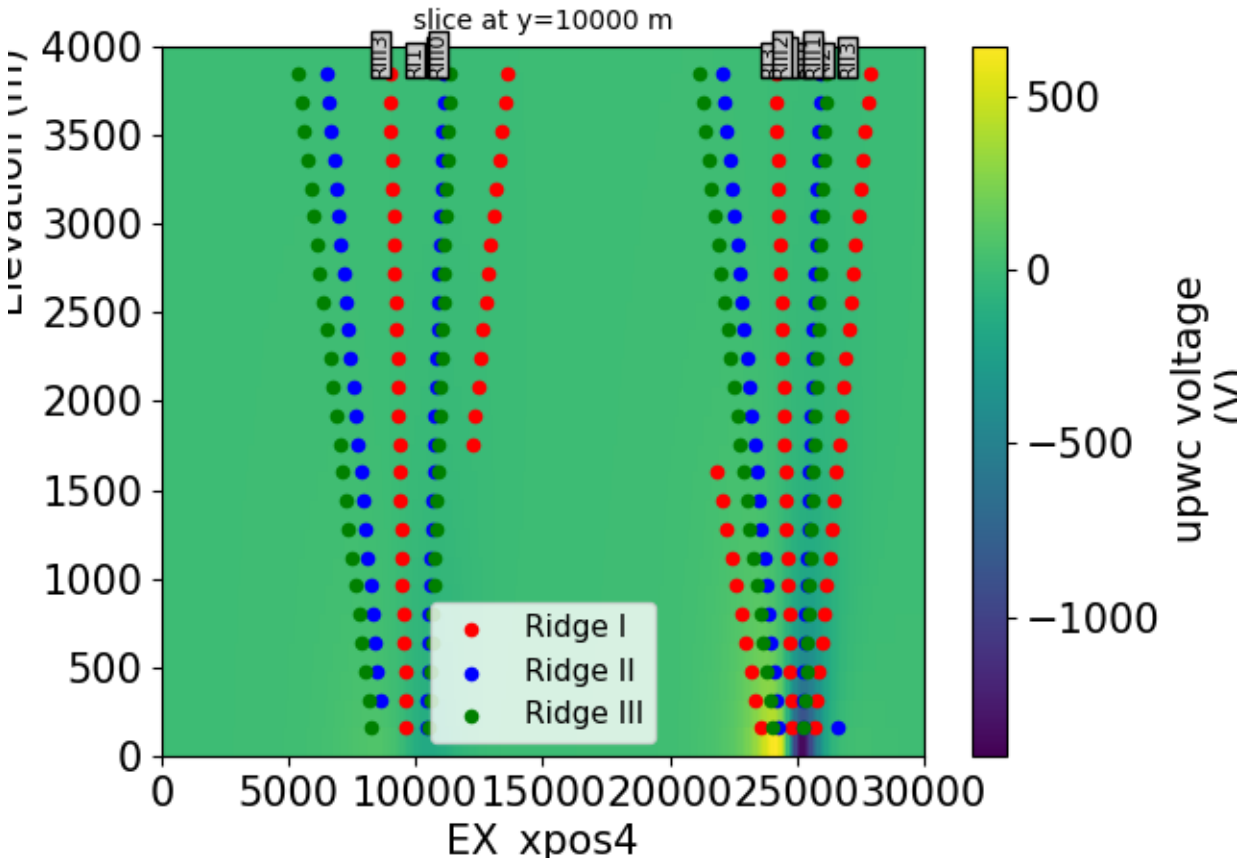
(continues on next page)

(continued from previous page)

```
dfI_f, dfII_f, dfIII_f =  D_f[0:3]
hI_f, hII_f, hIII_f = D_f[3:6]
df_f = D_f[0:3]
```

plot filtered ridges fitted over continuated section

```
fig = plt.figure()
ax = plt.gca()
# plt, cmap = pEXP.plot_xy(mesh, label=label_prop, Xaxis=x_axis, p1p2=np.array([p1, p2]))

pEXP.plot_xy(mesh, label=label_prop, ax=ax, Xaxis=x_axis, p1p2=np.array([p1, p2]))
pEXP.plot_ridges_harmonic(dfI_f,dfII_f,dfIII_f,ax=ax,label=True)

df_fit = dEXP.fit_ridges(df_f, rmvOutliers=True) # fit ridges on filtered data

# pEXP.plot_ridges_sources(df_fit, ax=ax, z_max_source=-max_elevation*2,
#                          ridge_type=[0,1,2],ridge_nb=None)
```



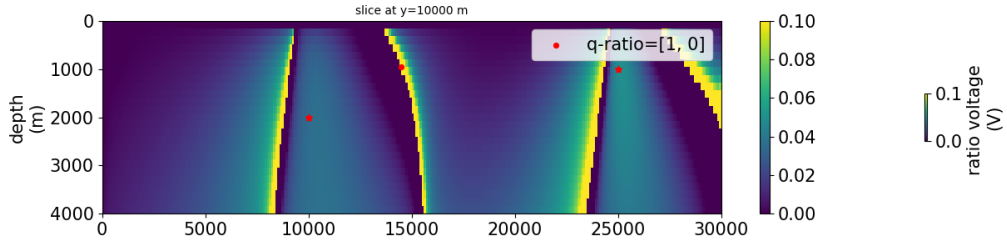

```
need to rotate first?
/home/docs/checkouts/readthedocs.org/user_builds/dexp-imaging/envs/latest/lib/python3.7/
↪site-packages/scipy/optimize/minpack.py:834: OptimizeWarning: Covariance of the␣
```

(continues on next page)

(continued from previous page)

```
↪parameters could not be estimated
  category=OptimizeWarning)
```

```
qratio = [1,0]
mesh_dexp, label_dexp = dEXP.dEXP_ratio(xp, yp, zp, U, shape,
                  zmin=0, zmax=max_elevation, nlayers=nlay,
                  qorders=qratio)
fig, ax = plt.subplots(figsize=(15,3))


plt, cmap = pEXP.plot_xy(mesh_dexp, label=label_dexp,
              markerMax=True,qratio=str(qratio), Vminmax = [0,1e-1],
              p1p2=np.array([p1,p2]), ax=ax, Xaxis=x_axis) #, ldg=)
plt.colorbar(cmap)

if x_axis=='y':
    plt.scatter(coord[0,0], coord[0,2],marker=(5, 1),c='red')
    plt.scatter(coord[1,0], coord[1,2],marker=(5, 1),c='red')
    # plt.annotate(str(masses),[easting, upward])
else:
    plt.scatter(coord[0,1], coord[0,2],marker=(5, 1),c='red')
    plt.scatter(coord[1,1], coord[1,2],marker=(5, 1),c='red')
    # plt.annotate(str(masses),[northing, upward])
```




```
/home/docs/checkouts/readthedocs.org/user_builds/dexp-imaging/checkouts/latest/fatiando/
↪gravmag/transform.py:182: UserWarning: Using 'height' <= 0 means downward continuation,
↪ which is known to be unstable.
  warnings.warn("Using 'height' <= 0 means downward continuation, " +
need to rotate first?
Markermax_z=960.0
Markermax_x=14500.0
```

Total running time of the script: (3 minutes 49.150 seconds)

4.4.3 Mise-à-la-masse analysis using the dEXP theory

The theory of the dEXP theory is applied here for the first time (to our knowledges) for active geoelectrical methods. The theory behind this choice and required adjustements are explained in the documentation section.

We show here:

- **preprocessing** of MALM for dEXP;
- a sensitivity analysis to **anomaly depth** (3d) of the dEXP theory apply to Mise-à-la-Masse data;
- effect of **noise level**;
- **in prep**: a case study where we used a Mise-à-la-Masse survey to identify a leakage in a landfill;
- **in prep**: effect of borehole data and downward continuation.

Preprocessing of MALM for dEXP

This code shows how to remove the influence of the return electrode B and correct as much as it is possible the field before dEXP analysis. Calculations used *utils_dEXP*, while plotting use the *plot_dEXP* module.

Application on a anomaly of electrical resistivity. The model data was created using geometric objects from `pygimli.meshtools`. The forward simulation of the data was done using `pygimli.ERTsimulate` module.

Note: This is part of a larger project aiming at inverting current sources density (see more at: https://icsd-dev.readthedocs.io/en/latest/)

References

Rucker, C., Gunther, T., Wagner, F.M., 2017. pyGIMLi: An open-source library for modelling and inversion in geophysics, Computers and Geosciences, 109, 106-123, doi: 10.1016/j.cageo.2017.07.011

```
import lib.utils_dEXP as uEXP
import lib.dEXP as dEXP
import lib.plot_dEXP as pEXP
import lib.set_parameters as para

from fatiando.vis.mpl import square
from fatiando import gridder
import matplotlib.pyplot as plt
from mpl_axes_aligner import align


import loadmalm.Load_sens_MALM as MALM
```

Import MALM data total field U_{T} = U{a} - U{b} The total voltage is the superposition of the injection +I on electrode A (masse) and injection -I on the return (and remote) electrode B filenames = ['MSoilR1000.0AnoR1Z-13.75W15H2.5L5S0Noise0', # Ratio = 1000

'MSoilR1AnoR1Z-13.75W15H2.5L5S0Noise0']

```
filenames = ['MSoilR1000.0AnoR1Z-13.75W15H2.5L5S0Noise0']
#filenames = ['MSoilR1AnoR1Z-13.75W15H2.5L5S0Noise0']
```

(continues on next page)

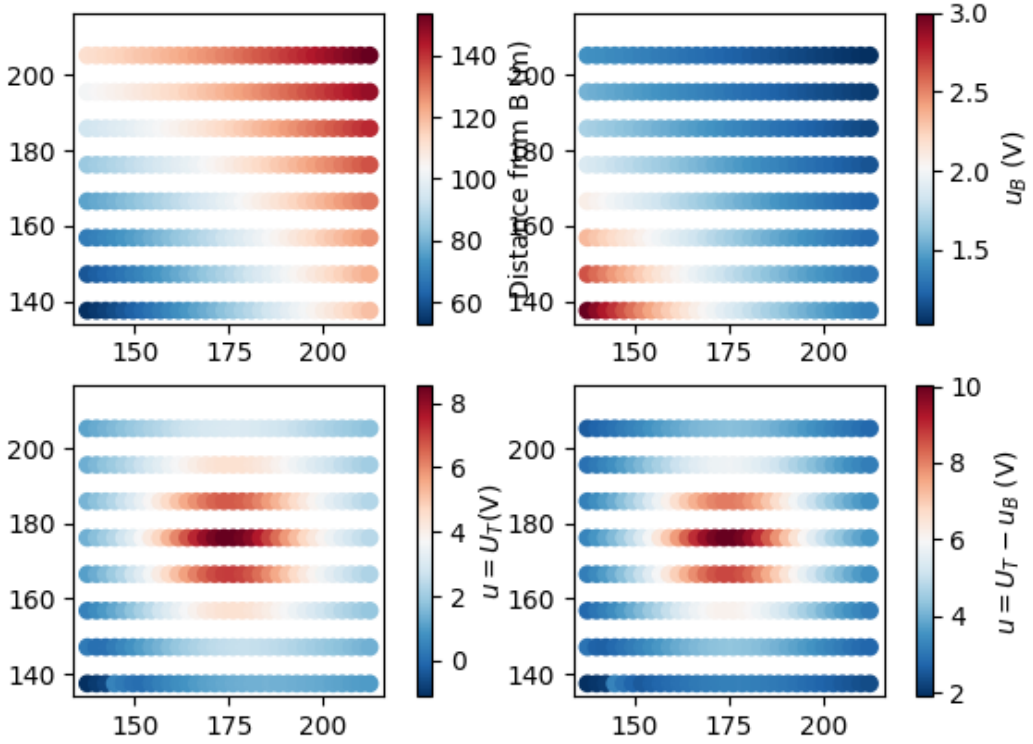
(continued from previous page)

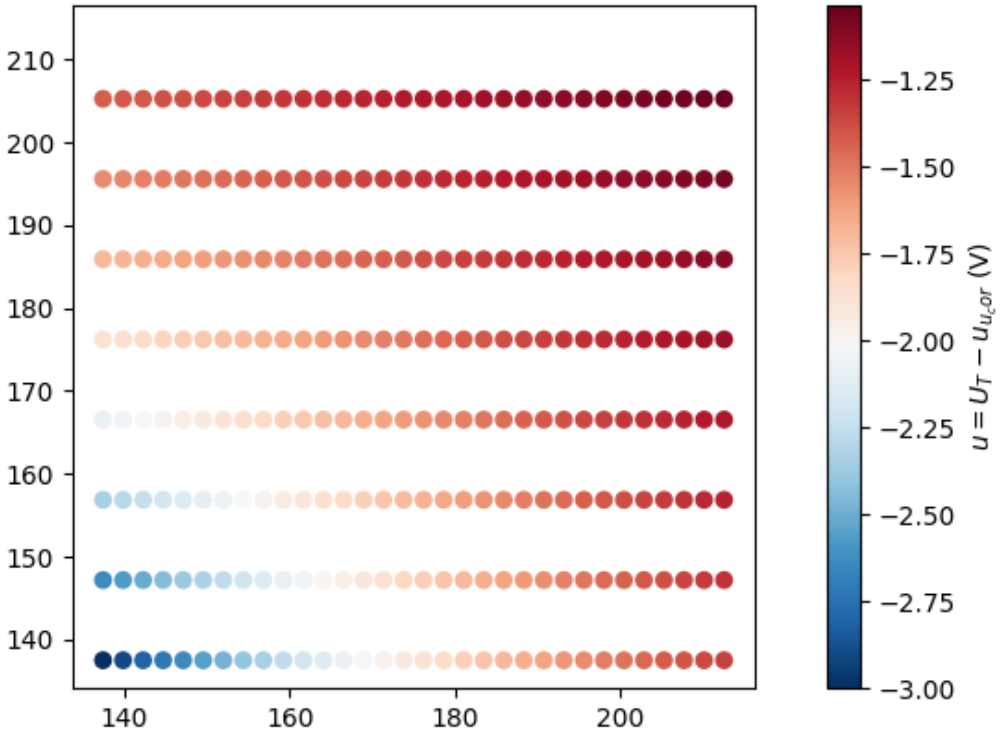
```
for fi in filenames:
    # Load data
    xp, yp, z, U, maxdepth, shape, p1, p2, SimName, model_prop= MALM.load_MALM_
↪sens3d(filename='./loadmalm/' +
                                                                fi + '.pkl')



    Bpos= model_prop['EA_EB_EN'][1]
    uT_elecs= model_prop['u_elecs']
    xyzs = model_prop['elecs']

    uA = model_prop['uAz0_grid'].T[3]
    uT = model_prop['uTz0_grid'].T[3]
    xp, yp, zp = model_prop['uTz0_grid'].T[0:3]

    parameters = para.set_par(shape=shape,max_elevation=maxdepth)

    zp, qorder, nlay = parameters[2:5]
    # minAlt_ridge, maxAlt_ridge = parameters[5:7]
    max_elevation = 50
    minAlt_ridge = max_elevation*0.05
    maxAlt_ridge = max_elevation*0.65

    x1, x2, z1, z2 = [max(xp)/2-model_prop['HWD'][1]/2,max(xp)/2 + model_prop['HWD'][1]/
↪2,
                      model_prop['HWD'][2]+ model_prop['HWD'][0]/2,
                      model_prop['HWD'][2]- model_prop['HWD'][0]/2]
    xxzz = [x1, x2, z1, z2]
    CT = model_prop['SoilR']/model_prop['AnoR']


    fix_peak_nb = 4
    smooth = False
    interp = True

    #%%
    shape = (200,200)
    _,_,uA = gridder.interp(xp,yp,uA,shape)
    xp,yp,uT = gridder.interp(xp,yp,uT,shape)

    #%%
    # remove B return electrode effects using :mod:`uEXP.cor_field_B` using a resistivity␣
↪of 1 Ohm.m

    U_cor = uEXP.cor_field_B(xyzs[:-3,0],xyzs[:-3,1],xyzs[:-3,2],uT_elecs,Bpos,
                             rho=model_prop['SoilR'])

    #%%
    # Compare the voltage seen by the surface electrode using 2d plot over a line
    p1_elecs= [min(xyzs[:-3,0]),(max(xyzs[:-3,0])+min(xyzs[:-3,0]))/2]
    p2_elecs= [max(xyzs[:-3,0]),(max(xyzs[:-3,0])+min(xyzs[:-3,0]))/2]
```

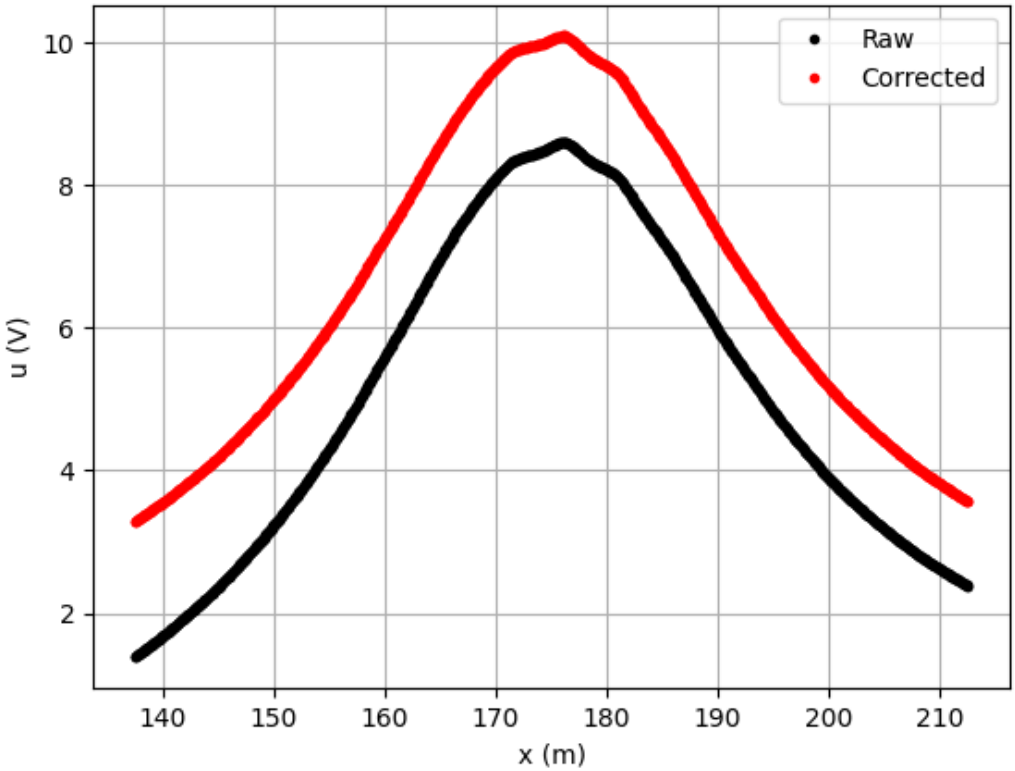
(continues on next page)

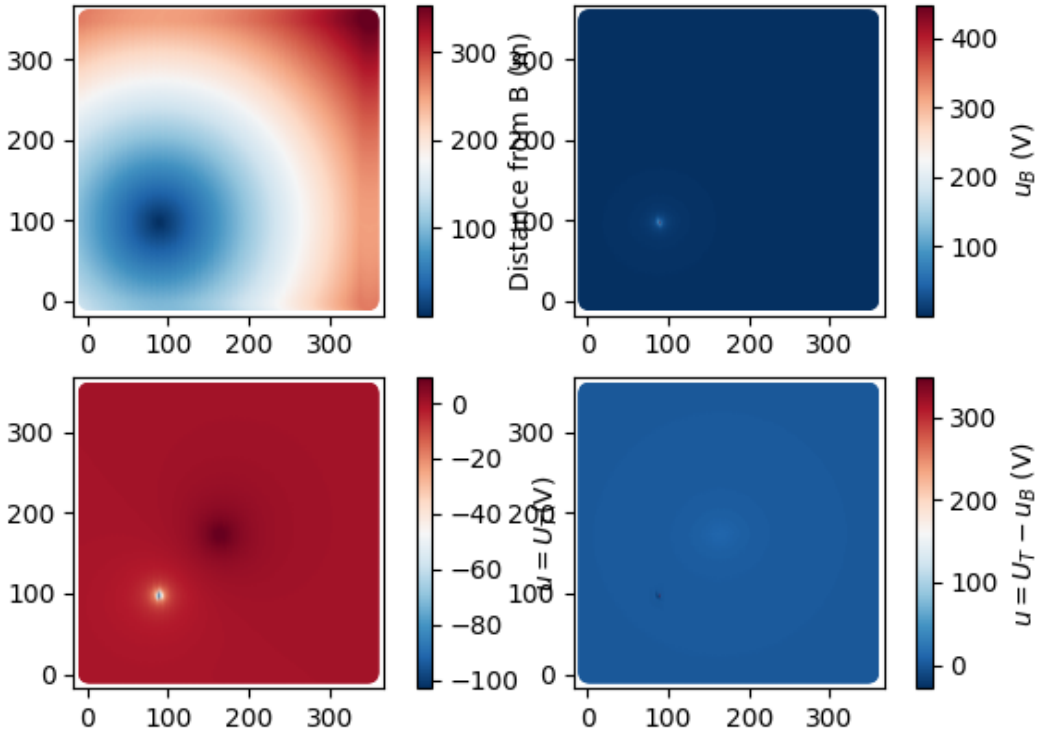
(continued from previous page)

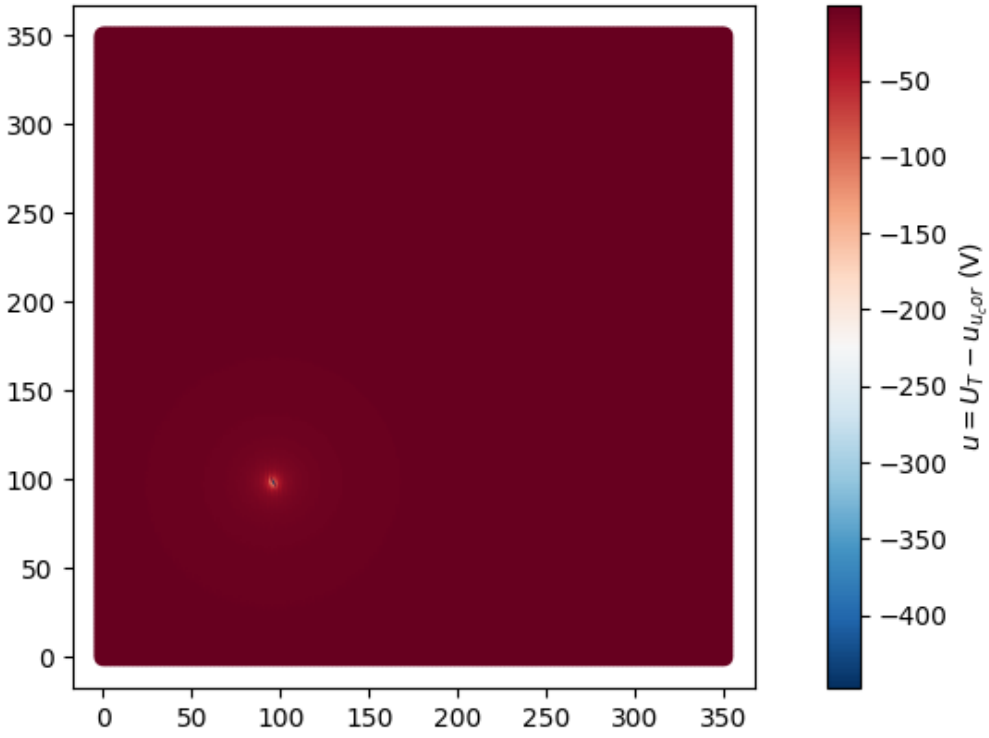
```
    xx, yy, distance, profile = gridder.profile(xyzs[:-3,0],xyzs[:-3,1], uT_elecs, p1_
↪elecs, p2_elecs, 1000)
    plt.figure()
    # plt.title(strname + '_data' + str(ZZ), fontsize=15)
    plt.plot(xx, profile, '.k',label='Raw')

    xx, yy, distance, profile = gridder.profile(xyzs[:-3,0],xyzs[:-3,1], U_cor, p1_elecs,
↪ p2_elecs, 1000)
    plt.plot(xx, profile, '.r',label='Corrected')
    plt.grid()
    plt.xlabel('x (m)')
    plt.ylabel('u (V)')

    plt.legend()

    #%%
    # Compare with numerical solution
    uT_field_cor = uEXP.cor_field_B(xp,yp,zp,uT,Bpos,rho=model_prop['SoilR'])

    #%%
    # Compare plot over a line
    xx, yy, distance, puT = gridder.profile(xp,yp, uT, p1, p2, 1000)

    plt.figure()
    plt.subplot(2,1,1)
    plt.scatter(xx, puT, c='b', marker='*', s=1, label='U_{T}')

    xx, yy, distance, puA = gridder.profile(xp,yp, uA, p1, p2, 1000)
    plt.scatter(xx, puA, c='k', marker='.', s=1, label='U_{A}')

    xx, yy, distance, puT_field_cor = gridder.profile(xp,yp, uT_field_cor, p1, p2, 1000)
    plt.scatter(xx, puT_field_cor,  c='r', marker='^', s=5, label='U_{cor} = U_{T} - U_
↪{B}')
    plt.grid()
    plt.legend()
    plt.xlabel('x (m)')
    plt.ylabel('u (V)')

    plt.subplot(2,1,2)
    diff1 = puT - puA
    diff2 = puA - puT_field_cor
    # plt.plot(xx, diff1, '.b', label='U_{T} - U_{A}')
    plt.plot(xx, diff2, '.g', label='U_{A} - U_{cor}')
    plt.xlabel('x (m)')
    plt.ylabel('u (V)')
    plt.legend()
```

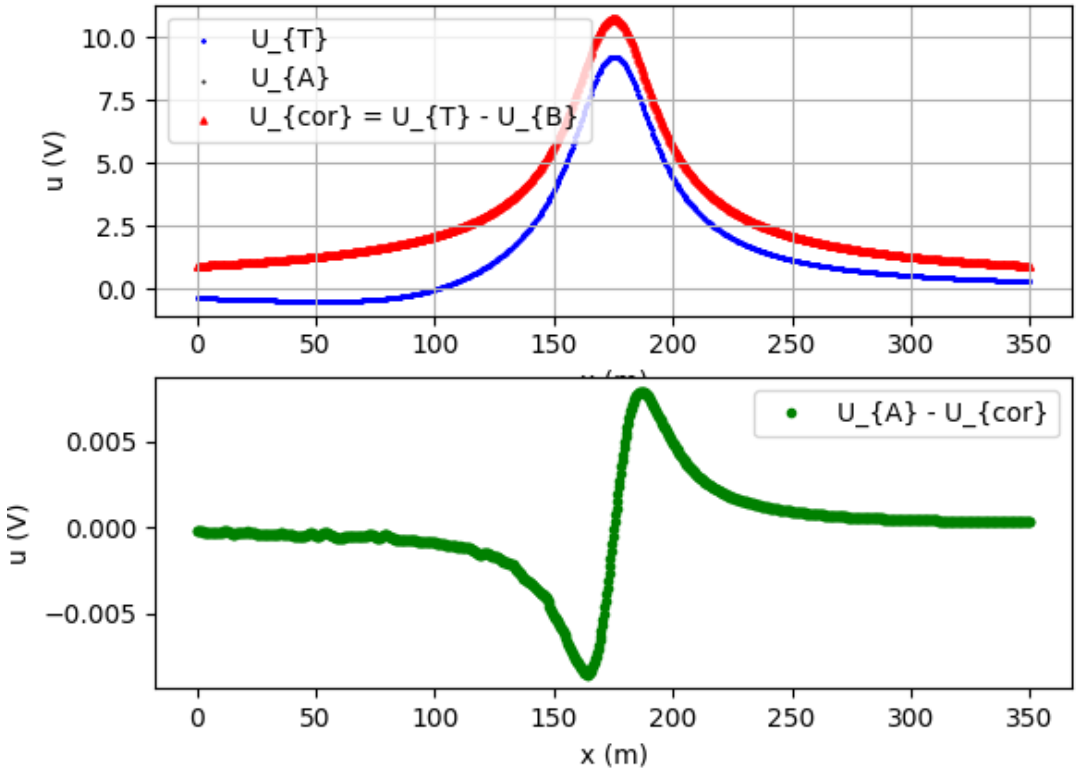

200
180
160
140
150
175
200
140
120
100
80
60
Distance from B (m)
3.0
2.5
2.0
1.5
u_B (V)
8
6
4
2
0
$u = U_T$(V)
10
8
6
4
2
$u = U_T - u_B$ (V)


210
200
190
180
170
160
150
140
140
160
180
200
−1.25
−1.50
−1.75
−2.00
−2.25
−2.50
−2.75
−3.00
$u = U_T - u_{u_c or}$ (V)


Raw
Corrected
u (V)
x (m)
10
8
6
4
2
140
150
160
170
180
190
200
210


Distance from B (m)
u_B (V)
$u = U_T$ (V)
$u = U_T - u_B$ (V)


350
300
250
200
150
100
50
0
0
50
100
150
200
250
300
350
−50
−100
−150
−200
−250
−300
−350
−400
$u = U_T - u_{u,or}$ (V)



Run dEXP over clear and raw field data

```
U = [uT,uA]

# MALM DATA synthetic anomaly: analysis of sensitivity

MESH = []
LABEL = []
DF_F = []
DF_FIT = []
XXZZ = []
CTm = []

import numpy
for i, ui in enumerate(U):

    #%%
    # Plot the data
    # pEXP.plot_line(xp, yp, ui,p1,p2, interp=interp, Xaxis='y')

    #%%
    # Pad the edges of grids (if necessary)
    # xp,yp,U, shape = dEXP.pad_edges(xp,yp,U,shape,pad_type=0) # reflexion=5
    # pEXP.plot_line(xp, yp,ui,p1,p2, interp=interp)
```

(continues on next page)

(continued from previous page)

```
#%%
# Upward continuation of the field data

mesh, label_prop = dEXP.upwc(xp, yp, zp, ui, shape,
                 zmin=0, zmax=max_elevation, nlayers=nlay,
                 qorder=qorder)

# plt, cmap = pEXP.plot_xy(mesh, label=label_prop, Xaxis='y')
# plt.colorbar(cmap)


#%%
# Ridges identification
# dEXP.ridges_minmax_plot(xp, yp, mesh, p1, p2,
#                                       label=label_prop,
#                                       fix_peak_nb=2,
#                                       Xaxis='y',
#                                       method_peak='find_peaks',
#                                       showfig=False)

# or  find_peaks or peakdet or spline_roots
D  = dEXP.ridges_minmax(xp, yp, mesh, p1, p2,
                                      label=label_prop,
                                      fix_peak_nb=fix_peak_nb,
                                      method_peak='find_peaks',
                                      Xaxis='y',
                                      smooth=smooth,
                                      returnAmp=True,
                                      showfig=True
                                      )
dfI, dfII, dfIII =  D[0:3]

df = dfI, dfII, dfIII
hI, hII, hIII  = D[3:6]
heights  = D[3:6]

#%%
fig = D[-1]
ax_list = fig.axes
ax_list[0].set_xlabel([])
ax_list[1].set_xlabel([])
ax_list[2].set_xlabel('x (m)')
ax_list[0].set_visible(True)

ax_list[0].set_ylabel('voltage u  \n (V)')
ax_list[1].set_ylabel('voltage\n ($V.m^{-2}$)')
ax_list[2].set_ylabel('voltage  \n ($V.m^{-2}$)')

#ax = plt.gca()
ax_list[0].xaxis.set_ticklabels([])
ax_list[1].xaxis.set_ticklabels([])
```

(continues on next page)

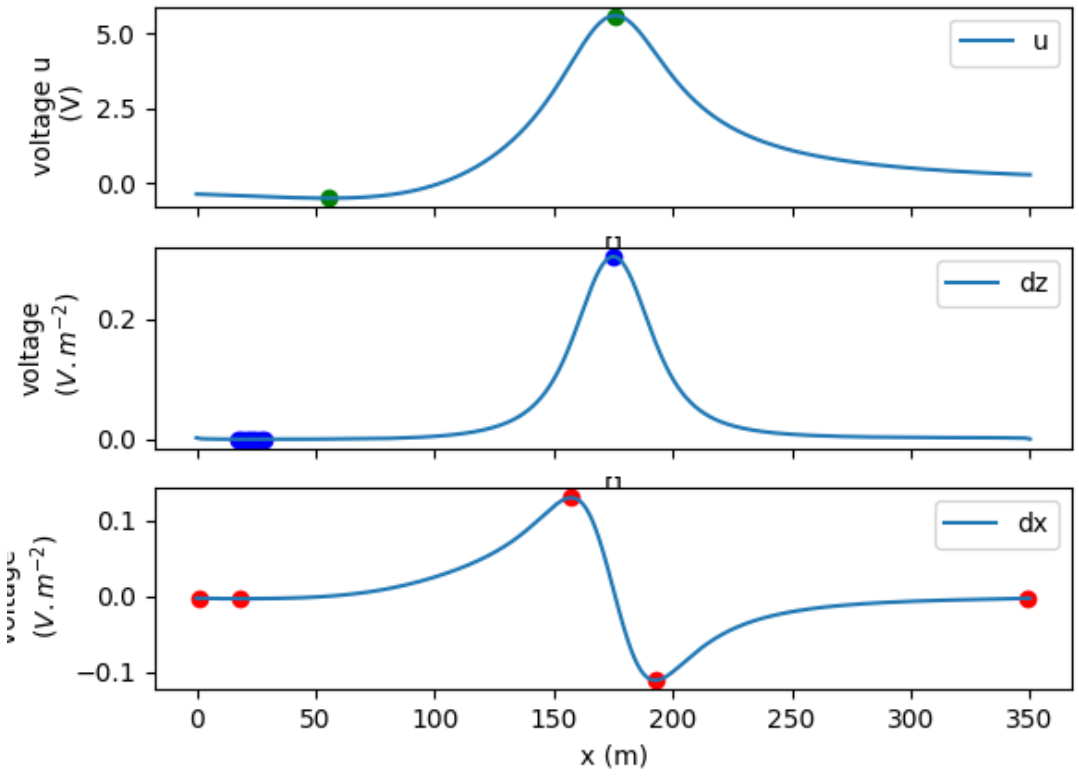
(continued from previous page)

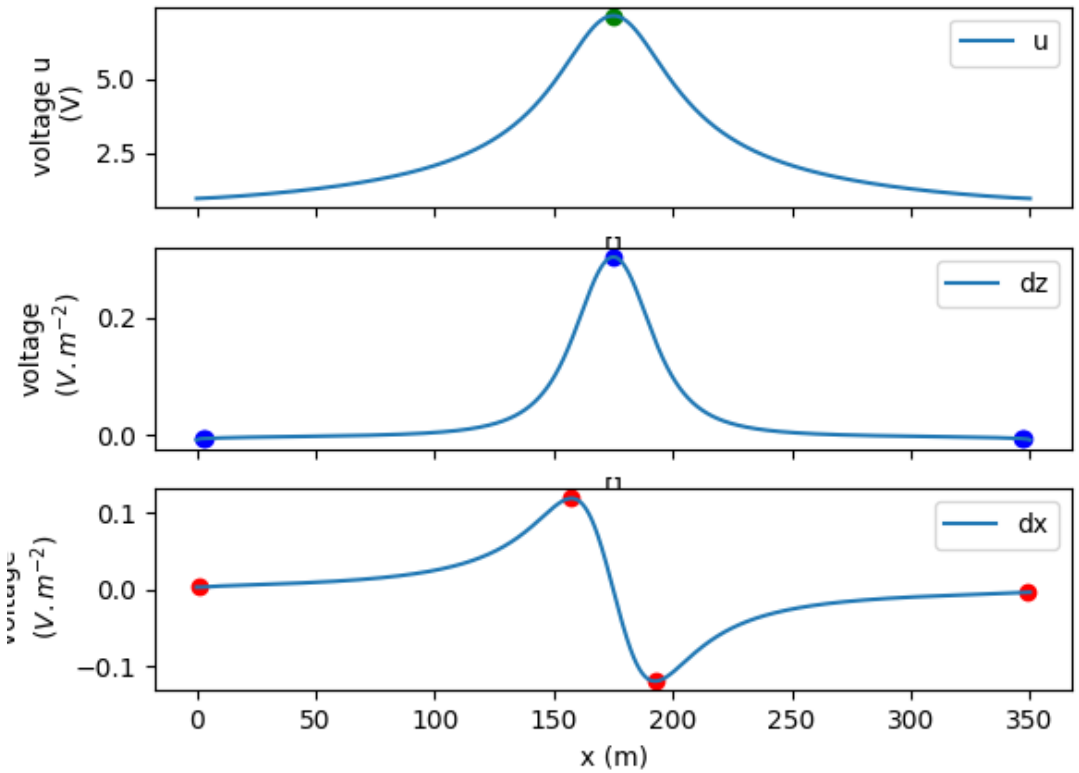
```
    fig

    if i==0:
       savename= 'fig2a_SI'
    else:
       savename= 'fig2b_SI'

    fig.savefig(savename+'.png', dpi=400, bbox_inches = "tight")
    fig.savefig(savename+'.svg', dpi=400, bbox_inches = "tight")
    fig.savefig(savename+'.pdf', bbox_inches = "tight")

    #%%
    # Plot ridges over continuated section

    # fig = plt.figure()
    # ax = plt.gca()
    # pEXP.plot_xy(mesh, label=label_prop, ax=ax) #, ldg=)
    # pEXP.plot_ridges_harmonic(dfI,dfII,dfIII,ax=ax)

    #%%
    # Filter ridges (regionally constrainsted)

    D_f = dEXP.filter_ridges(dfI,dfII,dfIII,
                                        minDepth=minAlt_ridge,
                                        maxDepth=maxAlt_ridge,
                                        minlength=5,rmvNaN=True,
                                        xmin=100, xmax=300,
                                        Xaxis='y',
                                        heights=heights)

    # df_f = dfI_f, dfII_f, dfIII_f

    dfI_f, dfII_f, dfIII_f =  D_f[0:3]
    hI_f, hII_f, hIII_f  = D_f[3:6]
    heights  = D_f[3:6]



    # dfI_f,dfII_f, dfIII_f = dEXP.filter_ridges(dfI,dfII,dfIII,
    #                                            Xaxis='y',
    #                                            minDepth=minAlt_ridge,
    #                                            maxDepth=maxAlt_ridge,
    #                                            minlength=7,rmvNaN=True)
    #                                             #,
    #                                             #xmin=100, xmax=300)
    df_f = dfI_f, dfII_f, dfIII_f


    #%%
    # plot ridges fitted over continuated section

    # fig = plt.figure()
```

(continues on next page)

(continued from previous page)

```
    # ax = plt.gca()

    # pEXP.plot_xy(mesh, label=label_prop, ax=ax) #, ldg=)
    # pEXP.plot_ridges_harmonic(dfI_f,dfII_f,dfIII_f,ax=ax,label=True)

    df_fit = dEXP.fit_ridges(df_f, rmvOutliers=True) # fit ridges on filtered data

    # pEXP.plot_ridges_sources(df_fit, ax=ax, z_max_source=-max_elevation*1.4,
    #                          ridge_type=[0,1,2],ridge_nb=None)
    # square([x1, x2, z1, z2])
    # plt.annotate(CT,[(x1 + x2)/2, -(z1+z2)/2])

    #%%
    # save data loop

    MESH.append(mesh)
    LABEL.append(label_prop)
    DF_F.append(df_f)
    DF_FIT.append(df_fit)
    XXZZ.append(xxzz)
    CTm.append(CT)
```





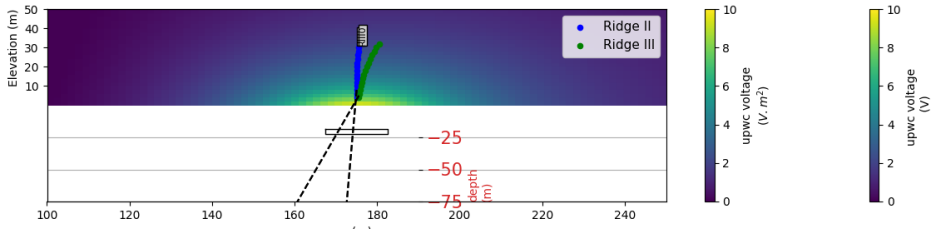

```
/home/docs/checkouts/readthedocs.org/user_builds/dexp-imaging/checkouts/latest/fatiando/
↪gravmag/transform.py:182: UserWarning: Using 'height' <= 0 means downward continuation,
↪ which is known to be unstable.
  warnings.warn("Using 'height' <= 0 means downward continuation, " +
NaN or Inf detected - trying to remove
/home/docs/checkouts/readthedocs.org/user_builds/dexp-imaging/envs/latest/lib/python3.7/
↪site-packages/scipy/optimize/minpack.py:834: OptimizeWarning: Covariance of the␣
↪parameters could not be estimated
  category=OptimizeWarning)
/home/docs/checkouts/readthedocs.org/user_builds/dexp-imaging/checkouts/latest/fatiando/
↪gravmag/transform.py:182: UserWarning: Using 'height' <= 0 means downward continuation,
↪ which is known to be unstable.
  warnings.warn("Using 'height' <= 0 means downward continuation, " +
NaN or Inf detected - trying to remove
/home/docs/checkouts/readthedocs.org/user_builds/dexp-imaging/envs/latest/lib/python3.7/
↪site-packages/scipy/optimize/minpack.py:834: OptimizeWarning: Covariance of the␣
↪parameters could not be estimated
  category=OptimizeWarning)
```

```
i = 0

dfI_f,dfII_f,dfIII_f = DF_F[i]
fig, ax1 = plt.subplots(figsize=(15,3))
x1, x2, z1, z2 = XXZZ[i]
```

(continues on next page)

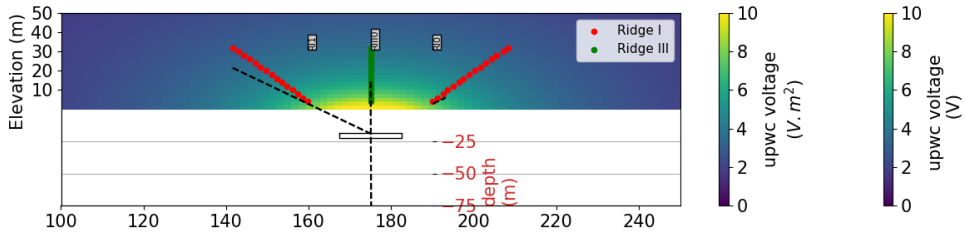
(continued from previous page)

```
square([x1, x2, z1, z2])
plt, cmap = pEXP.plot_xy(MESH[i], label=LABEL[i], ax=ax1, Xaxis='y',
          Vminmax=[0,10])
plt.colorbar(cmap, label='upwc voltage\n($V.m^2$)' )
pEXP.plot_ridges_harmonic(dfI_f,dfII_f,dfIII_f,ax=ax1,label=True)

df_fit = dEXP.fit_ridges(df_f, rmvOutliers=False) # fit ridges on filtered data

ax2 = pEXP.plot_ridges_sources(DF_FIT[i], ax=ax1, z_max_source=-max_elevation*1.2,
                       ridge_type=[0,1,2],ridge_nb=None,
                       xmin=100, xmax=250)
ax1.set_xlabel('x (m)')
fig.savefig('fig2c_SI.png', dpi=400, bbox_inches = "tight")
fig.savefig('fig2c_SI.svg', dpi=400, bbox_inches = "tight")
fig.savefig('fig2c_SI.pdf', dpi=400, bbox_inches = "tight")
```




```
/home/docs/checkouts/readthedocs.org/user_builds/dexp-imaging/envs/latest/lib/python3.7/
↪site-packages/scipy/optimize/minpack.py:834: OptimizeWarning: Covariance of the␣
↪parameters could not be estimated
  category=OptimizeWarning)
```

```
i = 1
dfI_f,dfII_f,dfIII_f = DF_F[i]
fig, ax1 = plt.subplots(figsize=(15,3))
x1, x2, z1, z2 = XXZZ[i]
square([x1, x2, z1, z2])
plt, cmap = pEXP.plot_xy(MESH[i], label=LABEL[i], ax=ax1, Xaxis='y',
          Vminmax=[0,10])
plt.colorbar(cmap, label='upwc voltage\n($V.m^2$)' )
pEXP.plot_ridges_harmonic(dfI_f,dfII_f,dfIII_f,ax=ax1,label=True)

df_fit = dEXP.fit_ridges(df_f, rmvOutliers=True) # fit ridges on filtered data

ax2 = pEXP.plot_ridges_sources(DF_FIT[i], ax=ax1, z_max_source=-max_elevation*1.2,
                       ridge_type=[0,1,2],ridge_nb=None,
                                          xmin=100, xmax=250)
ax1.set_xlabel('x (m)')
fig.savefig('fig2d_SI.png', dpi=400, bbox_inches = "tight")
fig.savefig('fig2d_SI.svg', dpi=400, bbox_inches = "tight")
fig.savefig('fig2d_SI.pdf', dpi=400, bbox_inches = "tight")
```




```
/home/docs/checkouts/readthedocs.org/user_builds/dexp-imaging/envs/latest/lib/python3.7/
↪site-packages/scipy/optimize/minpack.py:834: OptimizeWarning: Covariance of the␣
↪parameters could not be estimated
  category=OptimizeWarning)
```

Total running time of the script: (2 minutes 37.003 seconds)

CHAPTER

FIVE

API DOCUMENTATION

5.1 API reference: The pydEXP package

Imaging methods for potential fields.

Implements the DEXP method described in Fedi and Pilkington (2012).

Note: This is part of a larger project aiming at inverting current sources density (see more at: https://icsd-dev.readthedocs.io/en/latest/)

References

Fedi, M., and M. Pilkington (2012), Understanding imaging methods for potential field data, Geophysics, 77(1), G13, doi:10.1190/geo2011-0078.1

---

dEXP.**dEXP**(*x*, *y*, *z*, *data*, *shape*, *zmin*, *zmax*, *nlayers*, *qorder=0*, *SI=1*)

DEXP model (Fedi, 2012). (NOT YET TESTED)

Calculates a physical property distribution given potential field data on a **regular grid**. Uses depth weights. Parameters:

- **x, y**
  [1D-arrays] The x and y coordinates of the grid points
- **z**
  [float or 1D-array] The z coordinate of the grid points
- **data**
  [1D-array] The potential field at the grid points
- **shape**
  [tuple = (ny, nx)] The shape of the grid
- **zmin, zmax**
  [float] The top and bottom, respectively, of the region where the physical property distribution is calculated
- **nlayers**
  [int] The number of layers used to divide the region where the physical property distribution is calculated
- **qorder**
  [float] The order of the vertical derivative

- **SI**
  [float] The structural index

Returns:

- **mesh**
  [fatiando.mesher.PrismMesh] The estimated physical property distribution set in a prism mesh (for easy 3D plotting)

dEXP.**dEXP_ratio**(*x*, *y*, *z*, *data*, *shape*, *zmin*, *zmax*, *nlayers*, *qorders=[1, 0]*)

DEXP ratio model (NOT YET validated) Abbas, M. A., and Fedi, M. (2014). Automatic DEXP imaging of potential fields independent of the structural index. Geophysical Journal International, 199 (3), 1625-1632.

Calculates a physical property distribution given potential field data on a **regular grid**. Uses depth weights. Parameters:

- **x, y**
  [1D-arrays] The x and y coordinates of the grid points
- **z**
  [float or 1D-array] The z coordinate of the grid points
- **data**
  [1D-array] The potential field at the grid points
- **shape**
  [tuple = (ny, nx)] The shape of the grid
- **zmin, zmax**
  [float] The top and bottom, respectively, of the region where the physical property distribution is calculated
- **nlayers**
  [int] The number of layers used to divide the region where the physical property distribution is calculated
- **qorders**
  [1D-array] The order of the derivatives for the ratio calculation

Returns:

- **mesh**
  [fatiando.mesher.PrismMesh] The estimated physical property distribution set in a prism mesh (for easy 3D plotting)

dEXP.**filter_ridges**(*dfIf*, *dfIIf*, *dfIIIf*, *minDepth*, *maxDepth*, *minlength=3*, *rmvNaN=False*, ***kwargs*)

Filter non-rectiligne ridges (denoising)

Parameters: dfI: dataframe

contains ridges of type I

dfII: dataframe
contains ridges of type II

dfIII: dataframe
contains ridges of type III

- **minDepth**
  Text here
- maxDepth

- kwargs

heights: dataframe
contains heights of ridges of type I

Returns:

- **BB :**
  Text here

dEXP.**fit_ridges**(*df*, *rmvOutliers=False*)

Fit ridges and return points and fit equations to plot

Parameters:

- **df**
  dataframe including all tree types of ridges

Returns:

- **BB :**
  points and fit equations to plot

dEXP.**peakdet**(*v*, *delta*, *x=None*)

Converted from MATLAB script at http://billauer.co.il/peakdet.html

Returns two arrays

function [maxtab, mintab]=peakdet(v, delta, x) %PEAKDET Detect peaks in a vector % [MAXTAB, MINTAB] = PEAKDET(V, DELTA) finds the local % maxima and minima ("peaks") in the vector V. % MAXTAB and MINTAB consists of two columns. Column 1 % contains indices in V, and column 2 the found values. % % With [MAXTAB, MINTAB] = PEAKDET(V, DELTA, X) the indices % in MAXTAB and MINTAB are replaced with the corresponding % X-values. % % A point is considered a maximum peak if it has the maximal % value, and was preceded (to the left) by a value lower by % DELTA.

% Eli Billauer, 3.4.05 (Explicitly not copyrighted). % This function is released to the public domain; Any use is allowed.

dEXP.**profile_extra**(*x*, *y*, *v*, *point1*, *point2*, *size*, *algorithm='cubic'*)

Extract a profile between 2 points from spacial data.

Uses interpolation to calculate the data values at the profile points.

Parameters:

- **x, y**
  [1D arrays] Arrays with the x and y coordinates of the data points.
- **v**
  [1D array] Array with the scalar value assigned to the data points.
- **point1, point2**
  [lists = [x, y]] Lists the x, y coordinates of the 2 points between which the profile will be extracted.
- **size**
  [int] Number of points along the profile.
- **algorithm**
  [string] Interpolation algorithm. Either `'cubic'`, `'nearest'`, `'linear'` (see scipy.interpolate.griddata).

Returns:

- **[xp, yp, distances, vp]**
  [1d arrays] `xp` and `yp` are the x, y coordinates of the points along the profile. `distances` are the distances of the profile points from `point1`. `vp` are the data points along the profile.

dEXP.**profile_noInter**(*x*, *y*, *v*, *point1*, *point2*, *size=None*, ***kwargs*)

Extract a profile between 2 points from spacial data.

NO interpolation to calculate the data values at the profile points (find nearest point).

Parameters:

- **x, y**
  [1D arrays] Arrays with the x and y coordinates of the data points.
- **v**
  [1D array] Array with the scalar value assigned to the data points.
- **point1, point2**
  [lists = [x, y]] Lists the x, y coordinates of the 2 points between which the profile will be extracted.
- **size**
  [int] Number of points along the profile.

Returns:

- **[xp, yp, distances, vp]**
  [1d arrays] `xp` and `yp` are the x, y coordinates of the points along the profile. `distances` are the distances of the profile points from `point1`. `vp` are the data points along the profile.

dEXP.**ridges_intersection_Z0**(*fit*, *ax=None*, *ridge_nb=None*)

Find intersection of ridges (NOT YET IMPLEMENTED)

Parameters:

- **a**
  Text here

Returns:

- **BB :**
  return intersection by pairs

dEXP.**ridges_minmax**(*x*, *y*, *mesh*, *p1*, *p2*, *qorder=0*, *z=0*, *label='upwc'*, *fix_peak_nb=None*, *interp=True*, *smooth=False*, *showfig=False*, ***kwargs*)

Form a multiridge set RI and RII : zeros of the first horizontal and first vertical derivatives of the potential field RIII :zeros of the potential field itself

---

Note: ridges generated by isolated simple sources point, line, sheet, and contact are straight lines defined by the zeros of a potential

---

field and its horizontal and vertical derivatives at all measured or computed levels

Parameters:

- **x, y**
  [1D-arrays] The x and y coordinates of the grid points
- **mesh**
  [fatiando.mesher.PrismMesh] The estimated physical property distribution set in a prism mesh (for easy 3D plotting). The upward continuated field mesh (of order-q derivative). Read the property label 'upwc' of the mesh by default

- **p1, p2**
  [1D-arrays] The p1 and p2 coordinates of the extracted profile end points
- **qorder**
  [int] The derivative order
- **z**
  [float or 1D-array] The z coordinate of the grid points
- **label**
  [string] Label of the estimated physical property distribution
- **fix_peak_nb**
  [int] The maximum number of peak to identify
- **interp**
  [True or False] If True, will interpolate values between the data points.
- **smooth**
  [True or False] If True, will apply a low-pass filter values.
- **showfig**
  [True or False] If True, will display the figure.

****kwargs**
prominence for peak detection reverse: start peak analysis from bottom to top

Returns:

- **MinMax_peaks :**
  Panda dataframe containing ridges RI, RII and RII

dEXP.**ridges_minmax_plot**(*x*, *y*, *mesh*, *p1*, *p2*, *qorder=0*, *z=0*, *fix_peak_nb=None*, *label='upwc'*, *interp=True*, *smooth=False*, *showfig=False*, ***kwargs*)

- **x, y**
  [1D-arrays] The x and y coordinates of the grid points
- **mesh**
  [fatiando.mesher.PrismMesh] The estimated physical property distribution set in a prism mesh (for easy 3D plotting)
- **p1, p2**
  [1D-arrays] The p1 and p2 coordinates of the extracted profile end points
- **qorder**
  [int] The derivative order
- **z**
  [float or 1D-array] The z coordinate of the grid points
- **fix_peak_nb**
  [int] The maximum number of peak to identify
- **label**
  [string] Label of the estimated physical property distribution
- **interp**
  [True or False] If True, will interpolate values between the data points.
- **smooth**
  [True or False] If True, will apply a low-pass filter values.

- **showfig**
  [True or False] If True, will display the figure.

Returns:

dEXP.**scalEULER**(*df*, *EXTnb=[1]*, *z0=0*)

Analysis (Euler deconvolution of ridges, Fedi 2019) (NOT YET IMPLEMENTED)

Parameters:

- **a**
  Text here

Returns:

- **BB :**
  Text here

dEXP.**scalFUN**(*df*, *EXTnb=[1]*, *z0=0*, ***kwargs*)

Analysis of ridges (NOT YET IMPLEMENTED)

Parameters:

- **a**
  Text here

Returns:

- **BB :**
  Text here

dEXP.**upwc**(*x*, *y*, *z*, *data*, *shape*, *zmin*, *zmax*, *nlayers*, *qorder=0*)

Upward continuation model (Fedi, 2012).

Calculates the upward continuation for given potential field data on a **regular grid**. Parameters:

- **x, y**
  [1D-arrays] The x and y coordinates of the grid points
- **z**
  [float or 1D-array] The z coordinate of the grid points
- **data**
  [1D-array] The potential field at the grid points
- **shape**
  [tuple = (ny, nx)] The shape of the grid
- **zmin, zmax**
  [float] The top and bottom, respectively, of the region where the physical property distribution is calculated
- **nlayers**
  [int] The number of layers used to divide the region where the physical property distribution is calculated
- **qorder**
  [float] The order of the vertical derivative

Returns:

- **mesh**
  [`fatiando.mesher.PrismMesh`] The estimated physical property distribution set in a prism mesh (for easy 3D plotting)

5.1.1 plot_dEXP: plotter

Some functions to plot results from DEXP transformation

plot_dEXP.**label2unit**(*label*)

Convert label name to unit

Parameters

label : str

Returns

unit : str

plot_dEXP.**plot_line**(*x*, *y*, *data*, *p1*, *p2*, *ax=None*, *interp=True*, ***kwargs*)

Parameters

x
[TYPE] x coords of the 2d map.

y
[TYPE] y coords of the 2d map..

data
[TYPE] 2d map dataset.

p1
[list] initial point coordinate (x,y) of the extracted profile.

p2
[list] final point coordinate (x,y) of the extracted profile.

**kwargs

Returns

xx
[np.array] interpolated x.

yy
[np.array] interpolated y.

distance
[np.array] profile distance.

profile
[TYPE] data values along the profile.

plot_dEXP.**plot_ridges_harmonic**(*RI=None*, *RII=None*, *RIII=None*, *ax=None*, *label=False*, *legend=True*, ***kwargs*)

Plot ridges in the harmonic domain

Parameters:

- **a**
  Text here

Returns:

- **BB**
  [] Text here

plot_dEXP.**plot_ridges_sources**(*df_fit*, *ax=None*, *ridge_type=None*, *ridge_nb=None*, *z_max_source=None*, ***kwargs*)

Plot ridges in the source domain and observe intersection point

Parameters: * df_fit

dataframe fit

Returns:

- **BB**
  [] Text here

plot_dEXP.**plot_scalFUN**(*points*, *fit*, *ax=None*, *z0=None*, *label=None*)

Plot scalfun function analysis

Parameters:

- **a**
  Text here

Returns:

- **BB**
  [] Text here

plot_dEXP.**plot_xy**(*mesh*, *scaled=0*, *label=None*, *ax=None*, *markerMax=False*, ***kwargs*)

Get vertical xy section of the mesh at max/2 and plot it

Parameters:

- **mesh**
  Fatiando mesh type
- **scaled**
  Txt here
- **label**
  Txt here
- **ax**
  Txt here
- **markerMax**
  Txt here

plot_dEXP.**plot_z**(*mesh*)

Get horizontal sections at z levels and plot it

Parameters:

- **mesh**
  Fatiando mesh type

plot_dEXP.**slice_mesh**(*x*, *y*, *mesh*, *label*, *p1*, *p2*, *ax=None*, *interp=True*, ***kwargs*)

Get vertical xy section of the mesh at max/2 and plot it

Parameters:

- **mesh**
  Fatiando mesh type
- **scaled**
  Txt here
- **label**
  Txt here Txt here
- **markerMax**
  Txt here

5.1.2 utils_dexp: utils

Miscellaneous utility functions.

@author: Benjamin

utils_dEXP.**cor_field_B**(*x*, *y*, *z*, *u*, *B*, *rho=100*, ***kwargs*)

Calculates the potential field (electric) produced by a current injection in B (return electrode) for a given homogeneous electrical resistivity rho.

Parameters

- **x, y**
  [1D-arrays] The x and y coordinates of the grid points
- **z**
  [float or 1D-array] The z coordinate of the grid points
- **u**
  [1D-array] The potential field at the grid points
- **Bu**
  [1D-array] The position of the return electrode B
- **rho**
  [int] The estimated electrical resistivity of the medium

Returns

u_cor
[1D-arrays] The corrected from the B return electrode influence potential field at the grid points

utils_dEXP.**load_surfer**(*fname*, *dtype='float64'*)

Read data from a Surfer ASCII grid file.

Surfer is a contouring, griding and surface mapping software from GoldenSoftware. The names and logos for Surfer and Golden Software are registered trademarks of Golden Software, Inc.

http://www.goldensoftware.com/products/surfer

Parameters:

- **fname**
  [str] Name of the Surfer grid file
- **dtype**
  [numpy dtype object or string] The type of variable used for the data. Default is numpy.float64. Use numpy.float32 if the data are large and precision is not an issue.

Returns:

- **data**
  [dict] The data in a dictionary with some metadata:
  - `'file'`
    [string] The name of the original data file
  - `'shape'`
    [tuple] (nx, ny), the number of grid points in x (North) and y (East)
  - `'area'`
    [tuple] (x1, x2, y1, y2), the spacial limits of the grid.
  - `'x'`
    [1d-array] Value of the North-South coordinate of each grid point.
  - `'y'`
    [1d-array] Value of the East-West coordinate of each grid point.
  - `'data'`
    [1d-array] Values of the field in each grid point. Field can be for example topography, gravity anomaly, etc. If any field values are >= 1.70141e+38 (Surfers way of marking NaN values), the array will be masked at those values (i.e., `'data'` will be a numpy masked array).

CHAPTER

SIX

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX