

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ»

Кафедра систем сбора и обработки данных

**ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине: Современные проблемы прикладной математики и  
научноёмкого программирования

на тему: Решение одномерных обратных задач

Вариант №3

Факультет: ФПМИ

Группа: ПММ-21

Выполнил: Черненко Д.А., Бариев Р.Р.

Проверил: к.т.н Денис Владимирович Вагин

Дата выполнения: 16.12.22

Отметка о защите:

Новосибирск 2022

**Цель работы:** Изучить принципы решения обратных задач. В ходе работы необходимо найти величину источника (линейная обратная задача), ток задавать  $I = 1..10$  А. Также необходимо найти значение коэффициента удельной проводимости  $\sigma_i$ , её задавать в диапазоне  $0.001..10$  См/м.

При выполнении задания необходимо исследовать влияние зашумления (в диапазоне  $1..10\%$ ) на решение обратной задачи. Расстояние между соседними приёмниками задавать не менее 10 м.

### **Ход работы:**

#### **Начальные данные. Общие моменты.**

Однородное полупространство, источник — ВЭЛ. Неизвестна проводимость полупространства. Дано три приёмника. Исследовать влияние зашумления в одном, двух и трёх приёмниках на решение задачи.

Для начала определимся с расположением источника и трёх приёмников. Координаты источника: А (0, 0, 0), В (100, 0, 0); первого приёмника: М<sub>1</sub> (1000, 0, 0) N<sub>1</sub> (1200, 0, 0); второго приёмника: М<sub>2</sub> (2000, 0, 0) N<sub>2</sub> (2800, 0, 0); третьего приёмника: : М<sub>3</sub> (3500, 0, 0) N<sub>3</sub> (4000, 0, 0).

Далее, задавшись значением тока  $I = 1$ А, истинным значением коэффициента удельной проводимости  $\sigma_{\text{ист}} = 0.1$  См / м, начальным значением коэффициента проводимости  $\sigma_{\text{нач}} = 0.001..1$  См / м.

Теперь решаем обратную задачу для постепенного уменьшения функционала

$$\Phi(\sigma) = \sum_i^n (w_i(V_i(\sigma) - \bar{V}))^2 \rightarrow_{\sigma} \min$$

И, находим расчётное значение  $\sigma_{\text{нач}}$ .

Для этого нашей бригадой был написан код на языке программирования Python, полный текст которого можно увидеть в Приложении 1.

### Основная часть.

Если ни один приёмник не будет зашумлён, то при  $\sigma_{\text{нач}}(\sigma false) = 0.01$  и  $\sigma_{\text{истинная}}(\sigma true) = 0.1$  мы получим следующую картину:

```
Enter sigma true(0.1)
0.1
Enter sigma false(0.01)
0.01
Enter the current
1
Number of true voltage 0 : 5.626689907289235e-05
Number of true voltage 1 : 2.083063861130348e-05
Number of true voltage 2 : 3.1721250239172616e-06

F(sigma) is:
54.523545706371195
10.919354823708105
1.713815000530695
0.15519877056195358
0.0037930671426916
4.18010460157721e-06
5.797021066080654e-12
1.1201454654910256e-23
0.0
0.0
Value of new sigma: 0.09999999999999999
```

Рисунок 1 — значение коэффициента проводимости при уменьшении функционала

Как мы можем увидеть, функционал  $\Phi(\sigma)$  уменьшается в соответствии с теорией, а также вместе с его уменьшением мы получаем приближение расчётного коэффициента проводимости к истинному значению, а значит алгоритм работает верно.

### Один приёмник зашумлён:

В данной части работы мы продемонстрируем, как работает написанный нами функционал, а также предоставим данные по зашумлению напряжений на приёмниках и то, как это всё влияет на расчётное значение коэффициента проводимости.

Для начала возьмём  $\sigma_{\text{нач}}(\text{sigma false}) = 0.001$ , а также зашумим первый приёмник (+10%) с координатами  $M_1 (1000, 0, 0)$   $N_1 (1200, 0, 0)$ , остальные при этом трогать не будем.

```
Enter sigma false(0.001)
0.001
Enter the current
1
Number of true voltage 0 : 5.626689907289235e-05
Number of true voltage 1 : 2.083063861130348e-05
Number of true voltage 2 : 3.1721250239172616e-06

Value of true voltage with +10% noise 0 : 6.189358898018159e-05
1e-10
F(sigma) is:
6849.994344738469
1677.2534509224756
402.147344260187
92.39704864655428
19.446441162326
3.402292142214042
0.39908230517272025
0.023208006292364948
0.0059237595336498906
0.005847955097594575
0.005847953216374275
0.00584795321637427
Value of new sigma: 0.0971590909090909
```

Рисунок 2 — значение коэффициента проводимости и  $\Phi(\sigma)$  при зашумлении первого приёмника

Истинная сигма в данном случае была равна всё также 0.1. Как мы увидим в дальнейшем, при определённых условиях функционал способен уменьшаться до 0, но в данном случае этого не произошло, а следовательно, вычисление  $\Phi(\sigma)$  мы остановили из-за того, что

$$\Phi(\sigma_{\text{текущее}}) - \Phi(\sigma_{\text{предыдущее}}) < 10^{-10}$$

Значение расчётной сигмы почти вплотную подошло к истинному, но из-за помех в первом приёмнике, расчёт производится с погрешностями.

Абсолютно точно также будут рассчитаны  $\Phi(\sigma)$  и  $\sigma_{\text{нач}}$  если зашумлению подвергнутся отдельно второй и третий приёмники (см. рисунки 3 и 4).

```

Enter sigma false(0.001)
0.001
Enter the current
1
Number of true voltage 0 : 5.626689907289235e-05
Number of true voltage 1 : 2.083063861130348e-05
Number of true voltage 2 : 3.1721250239172616e-06

Value of true voltage with +10% noise 1 : 2.291370247243383e-05
1e-10
F(sigma) is:
6849.994344738469
1677.253450922476
402.1473442601873
92.39704864655431
19.446441162326003
3.402292142214045
0.3990823051727208
0.023208006292364945
0.005923759533649901
0.005847955097594593
0.005847953216374288
0.005847953216374283
Value of new sigma: 0.0971590909090909

```

Рисунок 3 — значение коэффициента проводимости и  $\Phi(\sigma)$  при зашумлении второго приёмника

```

Enter sigma false(0.001)
0.001
Enter the current
1
Number of true voltage 0 : 5.626689907289235e-05
Number of true voltage 1 : 2.083063861130348e-05
Number of true voltage 2 : 3.1721250239172616e-06

Value of true voltage with +10% noise 2 : 3.489337526308988e-06
1e-10
F(sigma) is:
6849.994344738467
1677.2534509224756
402.14734426018725
92.39704864655428
19.446441162326
3.402292142214042
0.39908230517272036
0.023208006292364938
0.005923759533649878
0.005847955097594575
0.005847953216374277
0.005847953216374281
Value of new sigma: 0.0971590909090909

```

Рисунок 4 — значение коэффициента проводимости и  $\Phi(\sigma)$  при зашумлении третьего приёмника

### Два приёмника зашумлено:

В данном случае зашумим первый и второй приёмник на -5% ( $\sigma_{\text{false}} = 0.05$ ), второй и третий на -5%, а также первый и третий на такое же значение. Узнаем, что реакция на зашумление будет одинаковая и расчётная сигма выйдет идентичная для всех случаев (см. рисунки 5, 6, 7).

```
Enter sigma true(0.1)
0.1
Enter sigma false(0.05)
0.05
Enter the current
1
Number of true voltage 0 : 5.626689907289235e-05
Number of true voltage 1 : 2.083063861130348e-05
Number of true voltage 2 : 3.1721250239172616e-06

Value of true voltage with -5% noise 0 : 5.345355411924773e-05
Value of true voltage with -5% noise 1 : 1.9789106680738306e-05
1e-10
F(sigma) is:
0.4016741452192599
0.01953899727511189
0.0018021268522409235
0.0017226549493021637
0.0017226528854435766
0.0017226528854435777
Value of new sigma: 0.10356824264049953
```

Рисунок 5 — значение коэффициента проводимости и  $\Phi(\sigma)$  при зашумлении первого и второго приёмников

```

Enter sigma true(0.1)
0.1
Enter sigma false(0.05)
0.05
Enter the current
1
Number of true voltage 0 : 5.626689907289235e-05
Number of true voltage 1 : 2.083063861130348e-05
Number of true voltage 2 : 3.1721250239172616e-06

Value of true voltage with -5% noise 1 : 1.9789106680738306e-05
Value of true voltage with -5% noise 2 : 3.0135187727213984e-06
1e-10
F(sigma) is:
0.40167414521926015
0.019538997275111894
0.0018021268522409274
0.0017226549493021613
0.0017226528854435864
0.001722652885443585
Value of new sigma: 0.10356824264049955

```

Рисунок 6 — значение коэффициента проводимости и  $\Phi(\sigma)$  при зашумлении второго и третьего приёмников

```

Enter sigma true(0.1)
0.1
Enter sigma false(0.05)
0.05
Enter the current
1
Number of true voltage 0 : 5.626689907289235e-05
Number of true voltage 1 : 2.083063861130348e-05
Number of true voltage 2 : 3.1721250239172616e-06

Value of true voltage with -5% noise 0 : 5.345355411924773e-05
Value of true voltage with -5% noise 2 : 3.0135187727213984e-06
1e-10
F(sigma) is:
0.40167414521926015
0.019538997275111908
0.0018021268522409287
0.0017226549493021737
0.001722652885443592
0.0017226528854435853
Value of new sigma: 0.10356824264049955

```

Рисунок 7 — значение коэффициента проводимости и  $\Phi(\sigma)$  при зашумлении второго и третьего приёмников

### Три приёмника зашумлено:

Для начала возьмём  $\sigma_{\text{нач}}(\sigma false) = 0.01$  и добавим зашумление (+1%, +8%, +10%), посмотрим, как данный факт повлияет на значение расчётной  $\sigma_{\text{нач}}$ .

```
Number of true voltage 0 : 5.626689907289235e-05
Number of true voltage 1 : 2.083063861130348e-05
Number of true voltage 2 : 3.1721250239172616e-06

Value of true voltage with +1% noise 0 : 5.682956806362127e-05
Value of true voltage with +1% noise 1 : 2.1038944997416515e-05
Value of true voltage with +1% noise 2 : 3.2038462741564344e-06

F(sigma) is:
53.268116955754095
10.641741405915042
1.6612894108719072
0.14858057122809598
0.0035238014875844333
3.6250672742262483e-06
4.361173712629723e-12
6.3406485867912924e-24
2.5934189102206695e-32
2.5934189102206695e-32
Value of new sigma: 0.099009900990099
```

Рисунок 8 — расчётное значение проводимости при добавлении шума (1%)



```

Number of true voltage 0 : 5.626689907289235e-05
Number of true voltage 1 : 2.083063861130348e-05
Number of true voltage 2 : 3.1721250239172616e-06

Value of true voltage with +8% noise 0 : 6.0768250998723736e-05
Value of true voltage with +8% noise 1 : 2.2497089700207756e-05
Value of true voltage with +8% noise 2 : 3.4258950258306426e-06

F(sigma) is:
45.48737730117235
8.931001511667148
1.342157365069699
0.10987374386992094
0.0021046439305828085
1.331683820287378e-06
5.895550450402896e-13
1.1581834252218179e-25
0.0
0.0
Value of new sigma: 0.09259259259259259

```

Рисунок 9 — расчётное значение проводимости при добавлении шума (8%)

```

Number of true voltage 0 : 5.626689907289235e-05
Number of true voltage 1 : 2.083063861130348e-05
Number of true voltage 2 : 3.1721250239172616e-06

Value of true voltage with +10% noise 0 : 6.189358898018159e-05
Value of true voltage with +10% noise 1 : 2.291370247243383e-05
Value of true voltage with +10% noise 2 : 3.489337526308988e-06

F(sigma) is:
43.54840995677934
8.50761541633112
1.2645247051736082
0.10089147048932559
0.00181633217799406
9.989486794391988e-07
3.3186636989665863e-13
3.662703207476103e-26
0.0
0.0
Value of new sigma: 0.0909090909090909

```

Рисунок 10 — расчётное значение проводимости при добавлении шума (10%)

Как можно заметить, при возникновении шума и его воздействия на напряжения в трёх приёмниках, расчётное значение коэффициента проводимости начинает рассчитываться не так точно, как это было бы без зашумления любого вида (1%, 8%, 10%). Кроме того, фактор зашумления также влияет и на расчётное значение уменьшаемого функционала  $\Phi(\sigma)$ .

**Вывод:**

Выяснилось, что для ВЭЛ не важно, какой из приёмников оказывается зашумлённым, значения коэффициента проводимости одинаково неточно будут рассчитаны.

В задачах георазведки крайне важно учитывать фактор зашумления, так как данное обстоятельство может помешать правильному расчёту коэффициента проводимости, а следовательно, приведёт к неверной оценке местности.

## Приложение 1

(код на python)

```
import math

from numpy import double

#MAIN FUNCTION FOR CALCULATION SIGMA TRUE
def calculation_main():
    data_x = []
    sigm_01 = sigm_sin("true(0.1)")
    sigm_001 = sigm_sin("false(0.01)")
    Istart = current()
    F = 1
    F_old = 1

    with open('Points.txt', 'r') as f:
        for line in f:
            data_x.append([double(x) for x in line.split()])

    # CALCULATE Uabmn of sigm_0.1
    uTrue = []
    for i in range(3):
        uTrue.append(U_CALCULATOR(Istart, sigm_01, data_x, i))
        print("Number of true voltage", i, ":", uTrue[i])
    print("\n")

    # ADD SOME NOISE (+ 10%) FOR uTrue
    #for i in range(3):
    uTrue[0] = uTrue[0] + 0.1 * uTrue[0]
    print("Value of true voltage with +10% noise", 0, ":", uTrue[0])
    #print("\n")

    eps = 1
    x = 10
    x = math.pow(x, -10)
    print(x)

    print("F(sigma) is:")
    while eps > x:
        # CALCULATE Uabmn of sigm_0.01
        uFalse = []
        for i in range(3):
            uFalse.append(U_CALCULATOR(Istart, sigm_001, data_x, i))
            #print(uFalse[i])

        #CALCULATE MATRIX A11
        A11 = 0
        w = []
        for i in range(3):
            w.append(1 / uTrue[i])
            A11 += math.pow(w[i] * dV_for_dsigm(Istart, sigm_001, data_x, i),
```

2)

```

#CALCULATE VECTOR b11
b1 = 0
for i in range(3):
    b1 += ((-1) * math.pow(w[i], 2) * dV_for_dsigm(Istart, sigm_001,
data_x, i) * uFalse_minus_uTrue(Istart, uTrue[i], sigm_001, data_x, i))

#CALCULATE DELTA_SIGMA AND NEW SIGMA
delta_sigm = b1 / A11
new_sigm = sigm_001 + delta_sigm

#CHECK FUNCTION F(NEW_SIGMA)
sigm_001 = new_sigm
F = 0
for i in range(3):
    F += math.pow(w[i] * uFalse_minus_uTrue(Istart, uTrue[i],
sigm_001, data_x, i), 2)
print(F)
eps = math.copysign(F - F_old, 1)
F_old = F
print("Value of new sigma:", sigm_001)
return 0

def sigm_sin(m):
    print("Enter sigma", m)
    sintetic = double(input())
    return sintetic

def current():
    print("Enter the current")
    I = double(input())
    return I

def rCalculation(Ax, Bx, Mx, Nx):
    R = (1/(Bx - Mx) - 1/(Ax - Mx) - 1/(Bx - Nx) + 1/(Ax - Nx))
    return math.copysign(R, 1)

def dV_for_dsigm(I, sigm, data_X, i):
    const = (-I / (2 * math.pi * math.pow(sigm, 2)))
    dV_dsigm = (const * rCalculation(data_X[0][0], data_X[1][0], data_X[2 * i + 2][0], data_X[2 * i + 3][0]))
    return dV_dsigm

def U_CALCULATOR(I, sigm, data_X, i):
    const = (I / (2 * math.pi * sigm))
    u_True = const * (rCalculation(data_X[0][0], data_X[1][0], data_X[2 * i + 2][0], data_X[2 * i + 3][0]))
    return u_True

def uFalse_minus_uTrue(I, uTrue, sigmFalse, data_X, i):
    uFalse = U_CALCULATOR(I, sigmFalse, data_X, i)
    uResult = uFalse - uTrue
    return uResult

#####POINT OF INPUT#####
if __name__ == '__main__':
    calculation_main()

```