|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
| Практическое задание № 1 | | |
| по дисциплине «Цифровые модели и оценивание параметров» | | |
| **Линейные обратные задачи** | | |
|  | | |
|  |  |  |
| Группа ПМ-13 | Исакин даниил |
|  |  |
|  |  |
|  |  |
| Преподаватель | вагин денис владимирович |
|  |  |
|  |  |
| Новосибирск, 2024 | | |

# Задание

Положение приёмников: M1(200,0,0), N1(300,0,0); M2(500,0,0), N2(600,0,0); M3(1000,0,0), N3(1100,0,0)

Положение источников: A1(0,–500,0), B1(100,–500,0); A2(0,0,0), B2(100,0,0); A3(0,500,0), B3(100,500,0)

Однородное полупространство. Приёмники 1–3. Источники 1–3.

Определить значения сил токов I1, I2 , I3 в источниках.

# Математическая модель

Потенциал электрического поля V , создаваемый электрической линиями A1B1, A2B2 иA3B3 с постоянным током, расположенными на поверхности земли, в однородном полупространстве складывается из потенциалов, создаваемых их электродами: . Для электрода, по которому ток втекает в среду, . Получаем . Следовательно разность потенциалов на линиях MjNj будет равна 

Так как значения в приемниках в данной задаче могут отличаться на несколько порядков, введем весовые коэффициенты , где  - практическое значение в приемнике. Также, так как искомых параметров несколько, применим регуляризацию Тихонова. В итоге получим следующий минимизируемый функционал:



Решим эту задачу методом Гаусса–Ньютона. Дифференцируя по Ii, получаем



В качестве некоторых известных  будем брать начальное приближение. Тогда получится следующая СЛАУ:



Где:

, 

# Тестирование

Истинное значение токов: 1; 2; 3

Начальное приближение: 0.7; 1.7; 2.7

Значение α: 1e-10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Итерация |  |  |  | Ф(**I**) |
| 1 | 0.7 | 1.7 | 2.7 | 0.067500 |
| 2 | 1.00000001 | 2.000000 | 2.99999999 | 5.92561e-20 |

Истинное значение токов: 1; 2; 3

Начальное приближение: 2.5; 3.5; 4.5

Значение α: 1e-10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Итерация |  |  |  | Ф(**I**) |
| 1 | 2.5 | 3.5 | 4.5 | 1.664558866 |
| 2 | 1.04999983 | 2.0000000 | 2.95000018 | 1.378563e-18 |

Вывод: 1) Задача является некорректно поставленной, ввиду того, что матрица является вырожденной. Ввиду этого необходимо применять регуляризацию Тихонова

2) Тесты показали, что результаты довольно сильно зависят от начального приближения и параметра регуляризации

**Код программы**

import numpy as np

from numpy.linalg import norm as Enorm # Норма евклида

from scipy.linalg import solve

# Положения источников

A1 = np.array((0, -500, 0))

B1 = np.array((100, -500, 0))

A2 = np.array((0, 0, 0))

B2 = np.array((100, 0, 0))

A3 = np.array((0, 500, 0))

B3 = np.array((100, 500, 0))

# Положения приемников

M1 = np.array((200, 0, 0))

N1 = np.array((300, 0, 0))

M2 = np.array((500, 0, 0))

N2 = np.array((600, 0, 0))

M3 = np.array((1000, 0, 0))

N3 = np.array((1100, 0, 0))

A = [A1, A2, A3]

B = [B1, B2, B3]

M = [M1, M2, M3]

N = [N1, N2, N3]

I = np.array((1,2, 3)) # Истенное значение силы тока каждого источника

delta\_I = np.zeros((3)) # Дельта смещения для поиска силы тока

In = np.array((2.5, 3.5, 4.4)) # Начальное приближение для силы тока

I\_approx = In.copy() # Примерное значение для регуляризации

sigma = 0.1 # Проводимость среды

alpha = 1e-9 # Параметр регуляризации

# Потенциал на измерителе с учетом того, что источников тока 3 штуки

# A,B - массив координат источника

# M, N - точка измерителя

# I - массив токов источников

# sigma - коэффициент проводимости

def V\_AB\_MN(A, B, M, N, I, sigma):

res = 0.0

for i in range(0, 3):

const\_val = I[i]/(2\*np.pi \* sigma)

r\_BM = Enorm(B[i]-M)

r\_AM = Enorm(A[i]-M)

r\_BN = Enorm(B[i]-N)

r\_AN = Enorm(A[i]-N)

val2 = 1.0/r\_BM - 1.0/r\_AM

val3 = 1.0/r\_BN - 1.0/r\_AN

res = res + const\_val\*(val2 - val3)

return res # Значение напряжения на линии

# Потенциал на измерителе с учетом того, что источников тока 3 штуки

# A,B - массив координат источника

# M, N - точка измерителя

# I - массив токов источников начального приближения

# sigma - коэффициент проводимости

def dV\_AB\_MN\_dI1(A, B, M, N, I, sigma):

res = 0.0

const\_val = 1.0/(2\*np.pi \* sigma)

r\_BM = Enorm(B[0]-M)

r\_AM = Enorm(A[0]-M)

r\_BN = Enorm(B[0]-N)

r\_AN = Enorm(A[0]-N)

val2 = 1.0/r\_BM - 1.0/r\_AM

val3 = 1.0/r\_BN - 1.0/r\_AN

res = res + const\_val\*(val2 - val3)

return res # Значение напряжения на линии

def dV\_AB\_MN\_dI2(A, B, M, N, I, sigma):

res = 0.0

const\_val = 1.0/(2\*np.pi \* sigma)

r\_BM = Enorm(B[1]-M)

r\_AM = Enorm(A[1]-M)

r\_BN = Enorm(B[1]-N)

r\_AN = Enorm(A[1]-N)

val2 = 1.0/r\_BM - 1.0/r\_AM

val3 = 1.0/r\_BN - 1.0/r\_AN

res = res + const\_val\*(val2 - val3)

return res # Значение напряжения на линии

def dV\_AB\_MN\_dI3(A, B, M, N, I, sigma):

res = 0.0

const\_val = 1.0/(2\*np.pi \* sigma)

r\_BM = Enorm(B[2]-M)

r\_AM = Enorm(A[2]-M)

r\_BN = Enorm(B[2]-N)

r\_AN = Enorm(A[2]-N)

val2 = 1.0/r\_BM - 1.0/r\_AM

val3 = 1.0/r\_BN - 1.0/r\_AN

res = res + const\_val\*(val2 - val3)

return res # Значение напряжения на линии

# Функционал невязки МНК

def F(I\_, w, V):

val = 0.0

for i in range(0, 3):

val = val + (w[i]\*(V\_AB\_MN(A, B, M[i], N[i], I\_, sigma) - V[i]))\*\*2

return val

# Например для 1 - ого источника имеем

V = [V\_AB\_MN(A, B, Mi, Ni, I, sigma) for Mi, Ni in zip(M, N)]

w = [1/Vi for Vi in V] # Весовые коэффициенты

dV = [dV\_AB\_MN\_dI1, dV\_AB\_MN\_dI2, dV\_AB\_MN\_dI3]

Matr = np.zeros((3,3))

b = np.zeros((3))

for iteration in range(0, 100):

descripency = F(In, w, V)

print("Iter = ", iteration, " Невязка = " , descripency)

if descripency <= 1e-14 or iteration == 100-2:

#print("Iter = ", iteration, " Невязка = " , descripency)

break

# Генерация матрицы

for i in range(0, 3):

for j in range(0, 3):

sum = 0.0

for k in range(0, 3):

sum += w[k]\*\*2\* dV[i](A, B,M[k] ,N[k], In, sigma)\*dV[j](A, B,M[k] ,N[k], In, sigma)

Matr[i][j]= sum

Matr[i][i] = Matr[i][i] + alpha # Регуляризация системы уравнений

for i in range(0, 3):

val = 0.0

for j in range(0, 3):

val = val + (w[j]\*\*2)\*dV[i](A, B, M[j], N[j],In, sigma)\*(V\_AB\_MN(A, B, M[j], N[j], In, sigma) - V[j])

b[i] = -val

# Регуляризация

b[i] = b[i] - alpha\*(In[i] - I\_approx[i])

#print(b)

delta\_I = solve(Matr, b.reshape((3,1)))

delta\_I = delta\_I.reshape(3)

In = In + delta\_I

print("alpha = ", alpha)

print("Истенный ток: ", I)

print("Расчитанный: ", In)