МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

Кафедра информатики, математического и компьютерного моделирования

ОТЧЕТ

к работе №2 по дисциплине
«Методы оптимизации»

Направление подготовки
01.03.02 «Прикладная математика и информатика»

Выполнил студент
гр. Б9118-01.03.02миопд
Охроменко Д.А. ________
(ФИО) *(подпись)*

« 23 » ноября 2021 г.

г. Владивосток
2021

Постановка задачи

Функция оптимизации $f(x) = \dfrac{1}{2}(x^T A x) + b \cdot x$, A - невырожденная симметричная неопределенная матрица размерности 4×4, векторы b, x соответствующей размерности. Ограничение вида $f(x) \leq 0$ при $f(x) = ||x - x_0|| - r$. Необходимо найти минимум функции.

$$A = \begin{pmatrix} 7 & 4 & -9 & -2 \\ 4 & 11 & -7 & -8 \\ -9 & -7 & 6 & 10 \\ -2 & -8 & 10 & 3 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ -7 \\ 5 \\ 2 \end{pmatrix}, \quad x_0 = \begin{pmatrix} -4 \\ 2 \\ 2 \\ 1 \end{pmatrix}, \quad r = 10$$

Решение

Составим функцию Лагранжа для этой задачи.

$$L(x, y) = \frac{1}{2}(x^T A x) + b \cdot x + y(||x - x_0||^2 - r^2)$$

Вычислим производную функции $L(x, y)$ по x и приравняем ее к нулю.

$$\frac{\partial L}{\partial x} = Ax + b + 2y \cdot (x - x_0) = 0$$

Различим два случая:

1. $y = 0 \;\Rightarrow\; Ax + b = 0 \;\Rightarrow\; x_{1*} = -A^{-1} b$

   $x_1 = [-3.80128205,\ -5.91666667,\ -3.99358974,\ -5.66666667]$

   Получили подозрительную на оптимум точку, проверим ее на соответствие условию $||x_{1*} - x_0|| \leq r$

   $||x_{1*} - x_0|| = 14.279396878498593 \not\leq 10$

   Точка x_0 не удовлетворяет условию, отбрасываем ее.

2.

$$y > 0 \Rightarrow \begin{cases} (A + 2Iy)x + (b - 2yx_0) = 0 \\ ||x - x_0||^2 = r^2 \end{cases}$$

В результате преобразований мы получили систему нелинейных уравнений с 5 неизвестными. Решим ее с помощью метода Ньютона, решение находится по следующей формуле:

$$x^{k+1} = x^k - W(x^k)^{-1} * F(x^k),$$

где W - матрица Якоби, имеющая вид:

$$\begin{pmatrix} (A + 2Iy) & 2(x - x_0) \\ 2(x - x_0)^T & 0 \end{pmatrix}$$

Мы не можем использовать в качестве начального приближения x_0, т.к. в этом случае получим вырожденную матрицу Якоби. Т.к. функция может иметь несколько оптимальных точек - будем запускать метод Ньютона на различных начальных приближениях:

$$\begin{gathered} [x_{01} + r,\ x_{02},\ x_{03},\ x_{04}] \\ [x_{01},\ x_{02} + r,\ x_{03},\ x_{04}] \\ [x_{01},\ x_{02},\ x_{03} + r,\ x_{04}] \\ [x_{01},\ x_{02},\ x_{03},\ x_{04} + r] \end{gathered}$$

$$\begin{gathered} [x_{01} - r,\ x_{02},\ x_{03},\ x_{04}] \\ [x_{01},\ x_{02} - r,\ x_{03},\ x_{04}] \\ [x_{01},\ x_{02},\ x_{03} - r,\ x_{04}] \\ [x_{01},\ x_{02},\ x_{03},\ x_{04} - r] \end{gathered}$$

Построим таблицу с результатами:

i	нач. прибл.	результат x_i	y_i	$f(x_i)$
2	(6, 2, 2, 1)	5.3487, 2.7084, 2.4194, 4.8891	-1.0570	106.4352
3	(-4, 12, 2, 1)	5.4180, 1.6051, 1.6928, 4.4163	-1.2363	102.9669
4	(-4, 2, 12, 1)	5.5206, -0.0302, 0.6159, 3.7156	-1.5020	107.0025
5	(-4, 2, 2, 11)	5.3120, 3.2932, 2.8045, 5.1396	-0.9619	110.2958
6	(-14, 2, 2, 1)	-2.9874, -4.6060, -3.0138, -4.4951	0.1091	-39.9673
7	(-4, -8, 2, 1)	-2.9874, -4.6060, -3.0138, -4.4951	0.1091	-39.9673
8	(-4, 2, -8, 1)	-2.9874, -4.6060, -3.0138, -4.4951	0.1091	-39.9673
9	(-4, 2, 2, -9)	-2.9874, -4.6060, -3.0138, -4.4951	0.1091	-39.9673

Определим в какой из точек достигается оптимум. Для этого проверим на соответсвие условию $y < 0$ и выберем минимальное значение функции.

Минимальное значение функции f при заданных ограничениях достигается в точке $x_6 = (-2.9874, -4.6060, -3.0138, -4.4951)$.

Результат

$$x = (-2.9874, -4.6060, -3.0138, -4.4951), \quad f(x) = -39.9673$$

```
def result():
        count = 1000
        y = 0
        x_vec = []
        res_fun = []
        res = []

        for i in range(4):
                x_plus = np.concatenate((x_0, [y])).copy()
                x_plus[i] = x_plus[i] + r
                x_vec.append(x_plus)

        for i in range(4):
                x_minus = np.concatenate((x_0, [y])).copy()
                x_minus[i] = x_minus[i] - r
                x_vec.append(x_minus)

        for x in x_vec:
                res.append(Newton(count, x))

        for x in res:
                res_fun.append(f(A, x[:4], b))
```