МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Кафедра информатики, математического и компьютерного моделирования

Лабораторная работа №2

Направление подготовки
01.03.02 «Прикладная математика и информатика»
профиль « Математическое и информационное обеспечение математической деятельности »

Выполнил студент
гр. Б9120-01.03.02
Агличеев А.О. ________
(ФИО) *(подпись)*

Проверил
Яковлев А.А. ________
(ФИО) *(подпись)*

« 15 » мая 2023 г.

г. Владивосток
2023

1 Постановка задачи

Найти минимум функции R^n:

$$f(x) = \frac{1}{2} \cdot x^T A x + b \cdot x$$

с условием $\|x - x_0\| \leq r$

Исходные данные:

A – произвольная симметрическая невырожденная матрица, $A \in \mathbb{R}^4$

b – произвольный ненулевой вектор, $b \in \mathbb{R}^4$

x_0 – произвольный начальный ненулевой вектор, $x \in \mathbb{R}^4$

$$A = \begin{pmatrix} 0.958769 & 1.007950 & 0.960282 & 1.005220 \\ 1.007950 & 1.113201 & 1.047806 & 1.080203 \\ 0.960282 & 1.047806 & 1.052557 & 1.013009 \\ 1.005220 & 1.080203 & 1.013009 & 1.065958 \end{pmatrix} b = \begin{pmatrix} 1.82801 \\ 1.89756 \\ 1.52039 \\ 1.29904 \end{pmatrix} x_0 = \begin{pmatrix} 1.29117 \\ 1.52013 \\ 1.98189 \\ 1.44639 \end{pmatrix} r = 5$$

2 Решение

Найдем функцию Лагранжа:

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

Найдем точки минимума. Для этого возьмем частную производную по x и приравняем ее к нулю:

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

Рассмотрим два случая:

1. Пусть $y = 0$

   $Ax + b = 0$, тогда $x_* = -A^{-1}b$, где x_* – «подозрительная» на минимум точка

   $$x_* = \begin{pmatrix} 23.24602137 \\ -77.78375858 \\ -18.34034689 \\ 71.05482568 \end{pmatrix} f(x_*) = -20.34363927$$

   Проверим, подходит ли данная точка под условие $\|x - x_0\| \leq r$:

   $$\left\| \begin{matrix} 21.95484357 \\ -79.30388908 \\ -20.32224649 \\ 69.60843178 \end{matrix} \right\| = 109.67884689 > r$$

   Условие не выполняется. Таким образом, найденная точка не подходит под ограничения и не будет рассматриваться при выборе итогового ответа.

2. Пусть $y > 0$

   Преобразуем L'_x и получим следующую систему из пяти уравнений:

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

   Для нахождения точек, подозрительных на оптимум, воспользуемся методом Ньютона:

   $$x_{k+1} = x_k - f'(x_k)^{-1} \cdot f(x_k),$$

   где x_k – пятимерный вектор неизвестных, составленный из элементов x и y

   $f(x_k)$ – левая часть, данной системы,

   $f'(x_k)$ – матрица Якоби данной системы уравнения.

   $$f'(x) = J = \begin{pmatrix} A + 2Iy & 2(x - x_0) \\ 2(x - x_0)^T & 0 \end{pmatrix}$$

Метод Ньютона будем запускать на нескольких начальных приближений, т.к. функция может иметь несколько оптимальных точек:

$$x_1 = \begin{pmatrix} -2.7088222 \\ 1.5201305 \\ 1.9818996 \\ 1.4463939 \\ 4 \end{pmatrix} x_2 = \begin{pmatrix} 5.2911778 \\ 1.5201305 \\ 1.9818996 \\ 1.4463939 \\ 4 \end{pmatrix} x_3 = \begin{pmatrix} 1.2911778 \\ -2.4798695 \\ 1.9818996 \\ 1.4463939 \\ 4 \end{pmatrix} x_4 = \begin{pmatrix} 1.2911778 \\ 5.5201305 \\ 1.9818996 \\ 1.4463939 \\ 4 \end{pmatrix}$$

$$x_5 = \begin{pmatrix} 1.2911778 \\ 1.5201305 \\ -2.0181004 \\ 1.4463939 \\ 4 \end{pmatrix} x_6 = \begin{pmatrix} 1.2911778 \\ 1.5201305 \\ 5.9818996 \\ 1.4463939 \\ 4 \end{pmatrix} x_7 = \begin{pmatrix} 1.2911778 \\ 1.5201305 \\ 1.9818996 \\ -2.5536061 \\ 4 \end{pmatrix} x_8 = \begin{pmatrix} 1.2911778 \\ 1.5201305 \\ 1.9818996 \\ 5.4463939 \\ 4 \end{pmatrix}$$

Условие выхода из цикла:

$$\|x_{k+1} - x_k\| \leq \varepsilon,$$

где $\varepsilon = 10^{-6}$

В результате получаем несколько точек x_i, подозрительных на оптимум:

i	Начальное приближение	x_i	y_i	$f(x_i)$
1	$\begin{pmatrix} -2.7088222 \\ 1.5201305 \\ 1.9818996 \\ 1.4463939 \\ 4 \end{pmatrix}$	$\begin{pmatrix} -0.73928961 \\ -0.99715835 \\ -1.18075373 \\ -0.68388375 \end{pmatrix}$	2.2349577	1.434714
2	$\begin{pmatrix} 5.2911778 \\ 1.5201305 \\ 1.9818996 \\ 1.4463939 \\ 4 \end{pmatrix}$	$\begin{pmatrix} -0.73928961 \\ -0.99715835 \\ -1.18075373 \\ -0.68388375 \end{pmatrix}$	−2.578362	1.434714

3	$\begin{pmatrix} 1.2911778 \\ -2.4798695 \\ 1.9818996 \\ 1.4463939 \\ 4 \end{pmatrix}$	$\begin{pmatrix} -0.73928961 \\ -0.99715835 \\ -1.18075373 \\ -0.68388375 \end{pmatrix}$	2.877857	1.434714
4	$\begin{pmatrix} 1.2911778 \\ -2.4798695 \\ 1.9818996 \\ 1.4463939 \\ 4 \end{pmatrix}$	$\begin{pmatrix} -0.73928961 \\ -0.99715835 \\ -1.18075373 \\ -0.68388375 \end{pmatrix}$	-3.457333	1.434714
5	$\begin{pmatrix} 1.2911778 \\ 1.5201305 \\ -2.0181004 \\ 1.4463939 \\ 4 \end{pmatrix}$	$\begin{pmatrix} -0.73928961 \\ -0.99715835 \\ -1.18075373 \\ -0.68388375 \end{pmatrix}$	3.281529	1.434714
6	$\begin{pmatrix} 1.2911778 \\ 1.5201305 \\ 5.9818996 \\ 1.4463939 \\ 4 \end{pmatrix}$	$\begin{pmatrix} -0.73928961 \\ -0.99715835 \\ -1.18075373 \\ -0.68388375 \end{pmatrix}$	-4.974945	1.434714
7	$\begin{pmatrix} 1.2911778 \\ 1.5201305 \\ 1.9818996 \\ -2.5536061 \\ 4 \end{pmatrix}$	$\begin{pmatrix} -0.73928961 \\ -0.99715835 \\ -1.18075373 \\ -0.68388375 \end{pmatrix}$	2.244119	1.434714
8	$\begin{pmatrix} 1.2911778 \\ 1.5201305 \\ 1.9818996 \\ -2.5536061 \\ 4 \end{pmatrix}$	$\begin{pmatrix} -0.73928961 \\ -0.99715835 \\ -1.18075373 \\ -0.68388375 \end{pmatrix}$	-2.869787	1.434714

Выясниим, в какой из данных точек функция принимает минимальное значение. Отбросим результаты, полученные при $y < 0$, и получим, что минимальное значение функции $f(x)$ при заданных ограничениях достигается в точке:

$$x_{min} = \begin{pmatrix} -0.73928961 \\ -0.99715835 \\ -1.18075373 \\ -0.68388375 \end{pmatrix}$$

Минимальное значение функции:

$$f_{min}(x) = 1.434714$$

Алгоритм содержится в приложении 1.

3 Приложение

```
import numpy as np


def f(x: np.ndarray) -> float:
  res = .5*x.transpose()@A@x + b.transpose()@x
  return res[0][0]


def lagrange_slae(x: np.ndarray) -> np.ndarray:
  return np.append((A + 2*np.eye(4)*y)@x + (b + 2*y*x_0), [[np.linalg.norm(x
    - x_0)**2 - r**2]], axis=0)


def jacobian(x: np.ndarray) -> np.ndarray:
  J_1_1 = A + 2*np.eye(4)*y
  J_1_2 = 2*(x - x_0)
  J_2_1 = J_1_2.transpose()
  J_2_2 = [[0]]
  J_1 = np.append(J_1_1, J_1_2, axis=1)
  J_2 = np.append(J_2_1, J_2_2, axis=1)
  return np.append(J_1, J_2, axis=0)


def newton(x_k: np.ndarray, epsilon=1e-6, max_iter=30):
  x_prev = x_k
  x_cur = x_prev - np.linalg.inv(jacobian(x_prev[0:-1])) @ lagrange_slae(
   x_prev[0:-1])
  it = 0
  while np.linalg.norm(x_cur[0:-1] - x_prev[0:-1]) > epsilon and it <
   max_iter:
```

```
    it += 1
    x_prev = x_cur
    x_cur = x_prev - np.linalg.inv(jacobian(x_prev[0:-1])) @ lagrange_slae(
   x_prev[0:-1])
  return x_cur


if __name__ == '__main__':
  A = np.loadtxt("a.txt", usecols=(range(4)))
  b = np.loadtxt("b.txt", usecols=(range(1)), ndmin=2)
  x_0 = np.loadtxt("x_0.txt", usecols=(range(1)), ndmin=2)
  r = 5
  a = 4
  y = 3
  sign = 1
  x_ = np.append(x_0, [[y]], axis=0)

x_star = -np.linalg.inv(A) @ b
f_in_x_star = f(x_star)

for i in range(8):
  sign = -sign
  x_k = x_.copy()
  x_k[i//2][0] += sign * a
  res = newton(x_k)
```