Федеральное автономное образовательное учреждение
высшего образования

«Национальный исследовательский университет «Высшая школа экономики»»

Отчет по лабораторной работе 3

Решение систем алгебраических уравнений итерационными методами

Вариант 10: задачи 4.1.10, 4.5.3, 5.1.10, 5.4.4

Выполнил:

Студент группы БПМ-211
Ляхов Артём Андреевич

Преподаватель:

Брандышев Петр Евгеньевич

Март 2024 г.

Содержание

1 Задача 4.1.10 Решение системы нелинейных уравнений методом Ньютона

1.1 Формулировка задачи

Требуется, используя метод Ньютона, найти с точностью $\varepsilon = 10^{-6}$ все корни системы нелинейных алгебраических уравнений:

$$\begin{cases} f_1(x_1, x_2) = 0 \\ f_2(x_1, x_2) = 0 \end{cases} \quad \begin{cases} \sin(x_1 + x_2) - x_2 - 1.5 = 0 \\ x_1 + \cos(x_2 - 0.5) - 0.5 = 0 \end{cases} \tag{1}$$

1.2 Теоретический материал

Пусть дана система из n нелинейных уравнений с n неизвестными:

$$\begin{cases} f_1(x_1, \ldots, x_n) = 0 \\ f_2(x_1, \ldots, x_n) = 0 \\ \ldots \\ f_n(x_1, \ldots, x_n) = 0 \end{cases} \tag{2}$$

где $f_i(x_1, x_2, \ldots x_n)$, $i = 1, \ldots, n$ - непрерывно дифференцируемые в некоторой области $G \subset \mathbb{R}^n$ функции.

Обозначим за $F(x) = (f_1(x) \ \ldots \ f_n(x))^T$, $x \in \mathbb{R}^n$. Пусть $x^{(k)}$ - k-ое приближение решения системы 2. Построим линейную аппроксимацию $F(x)$ в окрестности точки x^k:

$$F(x^{(k+1)}) \approx F(x^{(k)}) \ + \ W(x^{(k)})(x^{(k+1)} - x^{(k)}) \tag{3}$$

где W - матрица Якоби:

$$W = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \ldots & \frac{\partial f 1}{\partial x_n} \\ \ldots & \ldots & \ldots \\ \frac{\partial f_n}{\partial x_1} & \ldots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

Решим систему 3 относительно x^{k+1}, в результате чего получим итеративный метод, называемый *методом Ньютона*:

$$x^{(k+1)} = x^{(k)} - W^{-1}(x^{(k)})F(x^{(k)}) \tag{4}$$

Вычисление обратной матрицы - сложная вычислительная задача. Для того, чтобы избежать этого введём обозначение $y^{(k)} = W^{-1}(x^{(k)})F(x^{(k)})$, тогда $y^{(k)}$ в виде решения системы линейных уравнений:

$$W(x^{(k)}) \cdot y^{(k)} = F(x^{(k)})$$

Тогда шаг метода Ньютона можно переписать в следующем виде:

$$\begin{cases} W(x^{(k)}) \cdot y^{(k)} = F(x^{(k)}) \\ x^{(k+1)} = x^{(k)} - y^{(k)} \end{cases} \tag{5}$$

1.3 Порядок решения задачи

1. Используя встроенные функции, локализуем корни системы уравнений графически.
2. Напишем функцию, вычисляющую корень системы двух нелинейных уравнений по методу Ньютона с точностью ε.
3. Используя вышеуказанную функцию, вычислим все корни заданной системы с точностью ε.
4. Найдём корни системы, используя встроенную функцию, и сравним с результатом полученным в третем пункте.

В качестве критерия окончания итеративного процесса чаще всего использую условие $||x^{(k+1)} - x^{(k)}|| < \varepsilon$.

При этом если $F(x) \in C^2(G)$, то имеет место *квадратичная сходимость* метода Ньютона.

1.4 Результаты

1.4.1 Графическая локализация корней

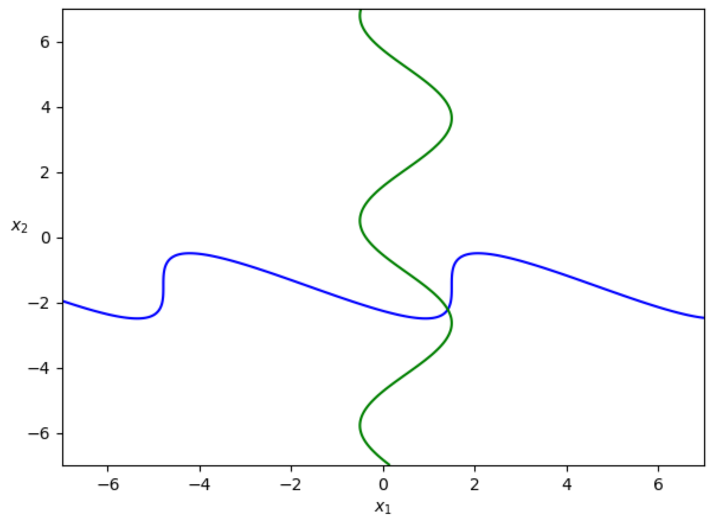

Рис. 1: Графическая локализация корня системы уравнений 1. Синим изображена линия уровня $f_1(x) = 0$, зелёным - линия уровня $f_2(x) = 0$.

1.4.2 Вычисление корней

Начальная точка	Решение методом Ньютона	Решение SciPy
(-2.0, 0.0)	(1.41423304, -2.22440735)	(1.41423304, -2.22440735)
(1.0, -1.5)	(1.41423304, -2.22440735)	(1.41423304, -2.22440735)

Таблица 1: Результаты вычисления корней с помощью реализованного метода и встроенной функции для двух начальных точек. Для первой точки метод Ньютона сошёлся за *7 итераций*, для второй - за *5 итераций*. Для обеих начальных точек абсолютная погрешность решений составила порядка $1.49 \cdot 10^{-9}$.

Вывод: как мы видим, реализованная нами программа находит решение с точностью, не превосходящей машинное эпсилон.

2 Задача 4.5.3 Нахождение ближайшей точки к поверхности

2.1 Формулировка задачи

Даны координаты точек P_1, P_2, P_3:

$$\begin{cases} P_1 = (15.5,\ 6.4,\ 12.162) \\ P_2 = (8.22,\ 5.879,\ 9.122) \\ P_3 = (14.531,\ 3.464,\ 5.375) \end{cases}$$

и уравнение поверхности S:

$$\frac{(x_1)^2}{a_1} + \frac{(x_2)^2}{a_2} = 2x_3$$

где $a_1 = 8.5 - 3 \times 0.25 = 7.75$, $a_2 = 2.3 + 3 \times 0.3 = 3.2$.

Необходимо определить ближайшую к поверхности точку и наиболее удалённую от поверхности точку. После этого построить на одном чертеже точечный график поверхности S и точки P_1, P_2, P_3.

2.2 Теоретическая часть

2.2.1 Метод Ньютона для задачи оптимизации

Рассмотрим задачу безусловной оптимизации:

$$\min_{x \in \mathbb{R}^n} f(x)$$

где $f(x)$ - дважды непрерывно-дифференцируемая функция.

Пусть x^k - k-ое приближение точки минимума. Воспользуемся разложением функции $f(x)$ в окрестности точки x^k:

$$f(x^k + h) \approx f(x^k) + f'(x^k)h + \frac{1}{2}h^T \cdot f''(x^k)h \tag{6}$$

где $f'(x^k)$ - градиент, $f''(x^k)$ - гессиан функции $f(x)$ в точке x^k.

Поскольку мы хотим найти h такое, что 6 достигает минимума, то мы можем найти градиент функции $f(x^k + h)$ по h и прировнять его к нулю. Положив, вместе с тем $x^{k+1} = x^k + h$ мы приходим к соотношению:

$$x^{k+1} = x^k - \left[f''(x^k)\right]^{-1} f'(x^k) \tag{7}$$

Как и в предыдущем случае, мы не будем вычислять явно обратную матрицу, вместо этого к итерации метода добавим решение системы уравнений:

$$\begin{cases} f''(x^k) \cdot y^k = f'(x^k) \\ x^{k+1} = x^k - y^k \end{cases} \tag{8}$$

Вместе с этим сделаем замечание, что если $f(x)$ не выпукла, то метод Ньютона может сходиться к локальному минимуму, поэтому в дальнейшем при решении мы будем использовать несколько начальных точек x^0.

2.2.2 Расстояние от точки до поверхности

Пусть $p = (p_1, p_2, p_3)$, $q = (q_1, q_2, q_3)$, $p, q \in \mathbb{R}^3$, введём функцию H как:

$$H(p,q) = \sum_{i=1}^{3} (p_i - q_i)^2 = \rho^2(p,q) = ||p-q||_2^2$$

Задачу нахождения расстояния от фиксированной точки до поверхности можно сформулировать в виде задачи условной оптимизации:

$$\rho(P_i, S) = \min_{x \in S} \rho(P_i, x)$$

Вместе с тем, минимум функции ρ достигается тогда и только тогда, когда достигается минимум функции H, то нам необходимо решить задачу оптимизации для функции H:

$$\min_{x \in S} H(P_i, x)$$

Однако в нашем случае эта задача очень просто сводится к задаче *безусловной* оптимизации. Для этого достаточно параметризовать поверхность S:

$$x = x(u,\theta) = \begin{cases} x_1 = \sqrt{a_1} \cdot u \cdot \cos(\theta) \\ x_2 = \sqrt{a_2} \cdot u \cdot \sin(\theta) \\ x_3 = 0.5 \cdot u^2 \end{cases}$$

В этом случае параметры u и θ могут принимать любые значения. Следовательно, мы можем переформулировать задачу нахождения расстояния от фиксированной точки P_i до поверхности S в виде задачи безусловной оптимизации:

$$\rho(P_i, S) = \min_{u,\theta} \rho(P_i, x(u, \theta))$$

А как мы поняли до этого, такая задача эквивалентна решению оптимизационной задачи для функции H:

$$\min_{u,\theta} H(P_i, x(u, \theta))$$

2.3 Порядок решения задачи

1. Напишем функцию, которая будет с помощью метода Ньютона минимизировать целевую функцию.

2. С помощью функции из предыдущего пункта и найдём расстояния от точек P_i до поверхности S. Чтобы исключить попадание в локальный минимум, будем производить перебор по сетке множества начальных точек.

3. Графически изобразим поверхность P и точки P_1, P_2, P_3 на одном графике.

2.4 Результаты

2.4.1 Результаты вычислений

Точка	Расстояние до S
$P_1 = (15.5, 6.4, 12.162)$	3.6474
$P_2 = (8.22, 5.879, 9.122)$	0.2754
$P_3 = (14.531, 3.464, 5.375)$	4.8716

Таблица 2: Результаты вычислений расстояний между поверхностью S и точками P_1, P_2, P_3 с использованием метода Ньютона.

Вывод: таким образом, наименее удалённой от S точкой оказалась P_2, а наиболее удалённой - P_3.

2.4.2 Изображение поверхности и точек

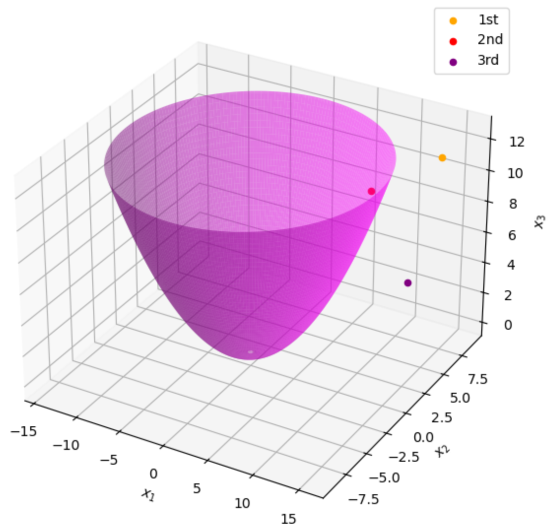

Рис. 2: Графическое изображение точек $P_i,\ i \in \{1, 2, 3\}$ и поверхности S.

3 Задача 5.1.10 Нахождение решения СЛАУ методом Зейделя

3.1 Формулировка задачи

Дана системы $Ax = b$. Требуется найте решение данной системы с помощью метода Гаусса. Выполнить 10 итераций по методу Зейделя. После этого, принимая решение с помощью метода Гаусса за точное, найти абсолютную погрешность итерационного решения.

$$A = \begin{pmatrix} 49.5 & 12.52 & 16.12 & 19.80 \\ 0 & 27.1 & 1.64 & 23.76 \\ 12.87 & 11.52 & 40 & -14.85 \\ 0 & 4.32 & 0.12 & 6.27 \end{pmatrix} \qquad b = \begin{pmatrix} -92.98 \\ 25.46 \\ -26.76 \\ -1.15 \end{pmatrix}$$

3.2 Теоретический материал

Систему уравнений вида $Ax = b$ можно преобразовать к виду $x = Bx+c$, который являтеся очень удобным для применения итерационных методов с помощью следующих соотношений:

$$\begin{cases} b_{ij} = -\frac{a_{ij}}{a_{ii}}, \quad i \neq j \\ b_{ij} = 0, \quad i = j \\ c_i = \frac{b_i}{a_{ii}} \end{cases} \tag{9}$$

В дальнейшем метод Зейделя сводится к итеративному приближению решения $Ax = b$ посредством соотношения

$$x^{k+1} = Bx^k + c, \tag{10}$$

где x^k - k-ое приближение решения.

Отметим при этом, что далеко не всегда данный итеративный метод сходится. Достаточным условием сходимости итерационных методов, которым мы будем пользоваться при решении, является условие $||B||_\infty < 1$.

3.3 Порядок решения задачи

1. Задать матрицу системы A и вектор правой части b. Найти решение системы $Ax = b$ с помощью метода Гаусса.

2. Преобразовать систему $Ax = b$ к виду $x = Bx + c$, удобному для итераций. Проверить выполнение достаточного условия сходимости итерационных методов $||B||_\infty < 1$.

3. Написать функцию, которая решает систему по методу Зейделя. Взять любое начальное приближение и сделать 10 итераций по этому методу. Найти абсолютную погрешность, принимая решения, полученное в пункте 1 за точное.

4. Взять другое начальное приближение и проделать ещё раз все вышеуказанные операции.

3.4 Результаты вычислений

x_1	x_2	x_3	x_4
-1.101	3	-2	-2.2121

Таблица 3: Решение системы $Ax = b$ с помощью метода Гаусса. Это решение принимается за точное.

x^0	x_1	x_2	x_3	x_4	ε
$(0, 0, 0, 0)$	-1.2549	2.7015	-1.7338	-1.9861	0.2985
$(1, 1, 1, 1)$	-1.1651	2.7979	-1.6921	-1.8861	0.3260

Таблица 4: Приближённые решения системы уравнений после 10 итераций по методу Зейделя. x^0 - вектор начального приближения, ε - абсолютная погрешность.

Вывод: Мы видим, что вне зависимости от начального приближения метод Зейделя сходится к точному решению, однако при этом скорость сходимости за 10 итераций для разных начальных приближений может отличаться.

4 Задача 5.4.4 Исследование зависимости решения СЛАУ итерационным методом от параметра

4.1 Формулировка задачи

Дана система уравнений $x = Bx + c$, где $B = B(t)$, $t = -1, -0.8, \ldots, 0.8, 1$ - параметр.

Необходимо:

1. Построить график (или гистрограмму) зависимости нормы $||B||_\infty$ от параметра t. По этому графику определить, при каких t выполнено достаточное условие сходимости итерационных методов.

2. Для наибольшего из значений t, при которых выполнено условие сходимости, решить с точностью $\varepsilon = 10^{-5}$ систему $x = Bx + c$.

Матрица B и вектор c задаются в виде:

$$B = \begin{pmatrix} -0.2 & \cos(3t) & 0.1 & 0.3 \\ 0.1 & 0.11 & 0.4 & -0.05 \\ 0.3 & 0.1 & \sin(3t) + \cos(2t) & 0.1 \\ 0.2 & -0.12 & 0.1 & 0.09 \end{pmatrix} \qquad b = \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \end{pmatrix}$$

4.2 Результаты

4.2.1 График зависимости нормы матрицы B

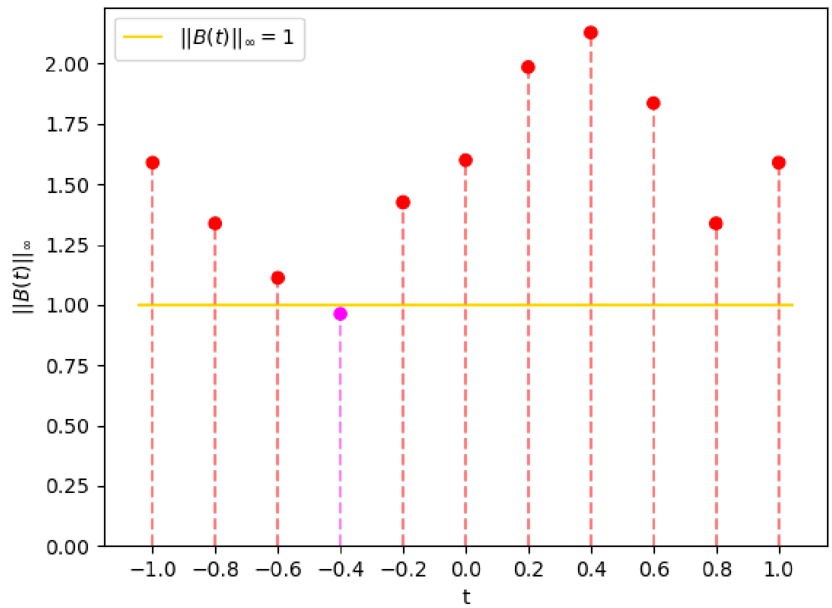

Рис. 3: График зависимости ∞-нормы матрицы B от параметра t.

Вывод: только для значения $t = -0.4$ выполнено достаточное условие сходимости итерационных методов. Таким образом, в дальнейшем мы будем искать решение системы для $t = -0.4$.

4.2.2 Решение системы

x_1	x_2	x_3	x_4
1.81394	2.26257	2.54025	3.67616

Таблица 5: Решение системы $x = B(t)x + c$ для $t = -0.4$. Для нахождения решения использовался метод Зейделя. В качестве критерия остановки использовалось условие $||x^{k+1} - x^k||_\infty < \frac{\varepsilon}{2}$, где $\varepsilon = 10^{-5}$ - точность.

5 Приложение

Все вычисления в рамках решения задач лабораторной работы проводились на языке Python (версия 3.11) с помощью пакета NumPy и пакета Autograd, который предоставляет набор инструментов для автоматического дифференцирования.

Код для проведения вычислений можно найти в прикреплённом вместе с отчётом ноутбуке lab3.ipynb.