РГР 1 по функциональному анализу

Выполнил студент

группы М8О-307Б-22

Кострюков Евгений Сергеевич

Задание

Докажите, что приведённое ниже отображение $T\colon C[0;1] \to C[0;1]$ (либо его степень) является сжимающим. Определите число итераций, необходимое для поиска неподвижной точки этого отображения с точностью $\varepsilon \in \{10^{-1}, 10^{-2}, 10^{-3}\}$ с помощью метода сжимающих отображений. С помощью вычислительной техники постройте график функции, являющейся неподвижной точкой отображения T. Проверьте результаты при различных значениях ε и различных начальных приближениях в методе сжимающих отображений. Продемонстрируйте несколько графиков, получающихся при промежуточных вычислениях.

Номер в списке – 15. Группа 307 => Вариант 5. l = 10, k = 7

5) $T(x) = \begin{cases} \frac{1}{1+k}x(3t) + \frac{l}{2}, & 0 \leqslant t \leqslant \frac{1}{3}; \\ f(t), & \frac{1}{3} < t < \frac{2}{3}; \\ \frac{1}{1+k}x(3t-2) + \frac{l}{2}, & \frac{2}{3} \leqslant t \leqslant 1, \end{cases}$ где $f(t)$ — уравнение ломаной, проходящей через точки $\left(\frac{4}{9}, 1\right)$ и $\left(\frac{5}{9}, -1\right)$, такой, что $T(x)$ — непрерывная функция;

Решение

$$T(x)=\begin{cases}\frac{1}{8}x(3t)+5, & 0\leqslant t\leqslant \frac{1}{3};\\ f(t), & \frac{1}{3}<t<2/3;\\ \frac{1}{8}x(3t-2)+5 & 2/3\leqslant t\leqslant 1,\end{cases}$$

где $f(t)$ – уравнение ломаной, проходящей через точки $(\frac{4}{9},1)$ и $(\frac{5}{9},-1)$, такой, что $T(x)$ – непрерывная функция.

$$\rho(x,y)=\max_{t\in[0,1]}|x(t)-y(t)|$$

$$\rho(T(x),T(y))=\max_{t\in[0,1]}|T(x)(t)-T(y)(t)|=$$

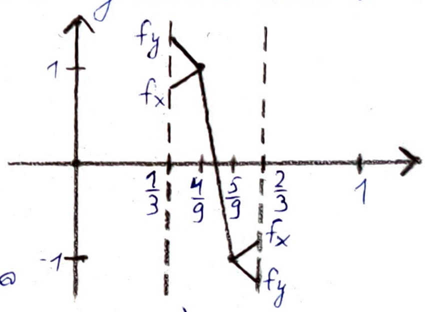
$$=\max\begin{pmatrix}\max\limits_{t\in[0;\frac{1}{3}]}\left|\frac{1}{8}x(3t)+5-\frac{1}{8}y(3t)-5\right|,\\ \max\limits_{t\in(\frac{1}{3};\frac{2}{3})}|f_x(t)-f_y(t)|,\\ \max\limits_{t\in[\frac{2}{3};1]}\left|\frac{1}{8}x(3t-2)+5-\frac{1}{8}y(3t-2)-5\right|\end{pmatrix}$$

Рассмотрим каждое выражение:

1) $\max\limits_{t[0;\frac{1}{3}]}\left|\frac{1}{8}x(3t)+5-\frac{1}{8}y(3t)-5\right|=\max\limits_{t[0;1/3]}\left|\frac{1}{8}x(3t)-\frac{1}{8}y(3t)\right|=\left|\begin{matrix}\text{Замена}\\ v=3t\end{matrix}\right|=$

$=\max\limits_{v\in[0,1]}\left|\frac{1}{8}x(v)-\frac{1}{8}y(v)\right|=\frac{1}{8}\max\limits_{v\in[0,1]}|x(v)-y(v)|=\frac{1}{8}\rho(x,y)$

2) $f_x(t)$ и $f_y(t)$ – уравнение ломаных для x и y соответственно.



Заметим, что функции $f_x(t)$ и $f_y(t)$ одинаковы на участке $[\frac{4}{9};\frac{5}{9}]$. По графику можно увидеть, что максимальное расстояние между функциями находится в одной из двух точек: $\frac{1}{3}$ или $\frac{2}{3}$. Это значит, что

$$\max_{t\in(\frac{1}{3};\frac{2}{3})}|f_x(t)-f_y(t)|\leqslant\max\begin{pmatrix}\max\limits_{t\in[0;\frac{1}{3}]}\left|\frac{1}{8}x(3t)+5-\frac{1}{8}y(3t)-5\right|,\\ \max\limits_{t\in[\frac{2}{3};1]}\left|\frac{1}{8}x(3t-2)+5-\frac{1}{8}y(3t-2)-5\right|\end{pmatrix},$$ а значит

это значение можно не рассматривать при поиске $\rho(T(x),T(y))$.

3) Решаем аналогично пункту 1:

$\max\limits_{t\in[\frac{2}{3};1]}\left|\frac{1}{8}x(3t-2)+5-\frac{1}{8}y(3t-2)-5\right|=\left|\begin{matrix}\text{Замена}\\ v=3t-2\end{matrix}\right|=\frac{1}{8}\max\limits_{v\in[0,1]}|x(v)-y(v)|=\frac{1}{8}\rho(x,y)$

Получаем: $\rho(T(x),T(y))=\max\begin{pmatrix}\frac{1}{8}\rho(x,y),\\ \frac{1}{8}\rho(x,y)\end{pmatrix}=\frac{1}{8}\rho(x,y)\Rightarrow \alpha=\frac{1}{8}\Rightarrow$

$\Rightarrow$ Отображение $T(x)$ сжимающее.

В методе сжимающих отображений количество итераций, необходимых для достижения требуемой точности, зависит от выбранной точки начального приближения.

Возьмём в качестве начального приближения точку x_0. Тогда $x_1 = T(x_0)$.

$$\rho(x_n, x^*) \leq \frac{\alpha^n}{1-\alpha}\rho(x_0, x_1) \leq \mathcal{E} \Rightarrow \frac{\alpha^n}{1-\alpha}\rho(x_0,x_1) \leq \mathcal{E} \Rightarrow \alpha^n \leq \frac{(1-\alpha)\mathcal{E}}{\rho(x_0,x_1)} \Rightarrow$$

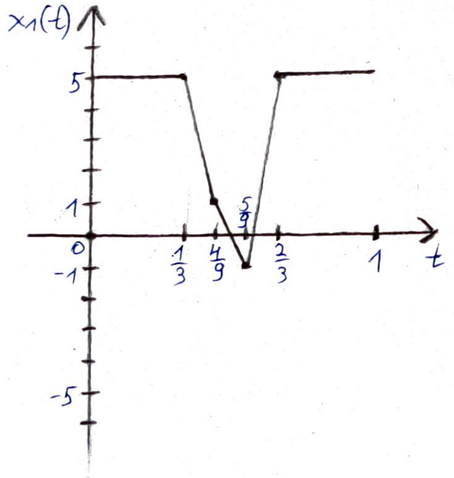
$$\Rightarrow n \geq \log_\alpha \frac{(1-\alpha)\mathcal{E}}{\rho(x_0,x_1)}.$$

Тогда $n = \left\lceil \log_\alpha \frac{(1-\alpha)\mathcal{E}}{\rho(x_0,x_1)} \right\rceil$ - минимальное число итераций.

Найдём n для случайного начального приближения и $\mathcal{E} \in \{10^{-1}, 10^{-2}, 10^{-3}\}$

Возьмём $x_0 = 0$ (константная функция).

$$x_1 = \begin{cases} \frac{1}{8}x_0(3t)+5, & t \in [0; \frac{1}{3}] \\ f(t), & t \in (\frac{1}{3}; \frac{2}{3}) \\ \frac{1}{8}x_0(3t-2)+5 & t \in [\frac{2}{3}; 1] \end{cases} = \begin{cases} 5, & t \in [0; \frac{1}{3}] \\ 17-36t, & t \in [\frac{1}{3}; \frac{4}{9}] \\ 9-18t, & t \in [\frac{4}{9}; \frac{5}{9}] \\ -31+54t, & t \in [\frac{5}{9}; \frac{2}{3}] \\ 5 & t \in [\frac{2}{3}; 1] \end{cases}$$



$\rho(x_0, x_1) = 5$

При $\mathcal{E} = 10^{-1}$: $n = \left\lceil \log_{\frac{1}{8}} \frac{\frac{7}{8}\cdot 10^{-1}}{5} \right\rceil \approx \lceil 1{,}95 \rceil = 2$

При $\mathcal{E} = 10^{-2}$: $n \approx \lceil 3{,}05 \rceil = 4$

При $\mathcal{E} = 10^{-3}$: $n \approx \lceil 4{,}16 \rceil = 5$

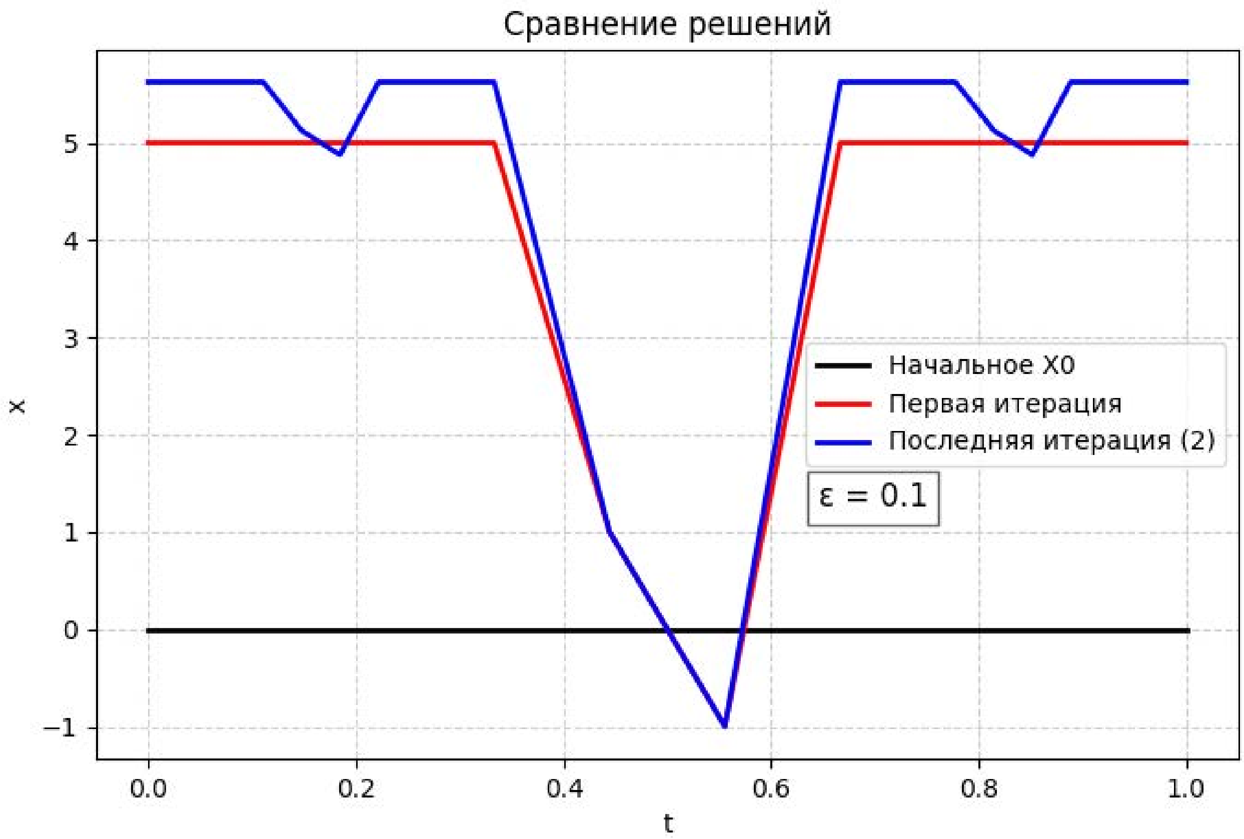
Код программы

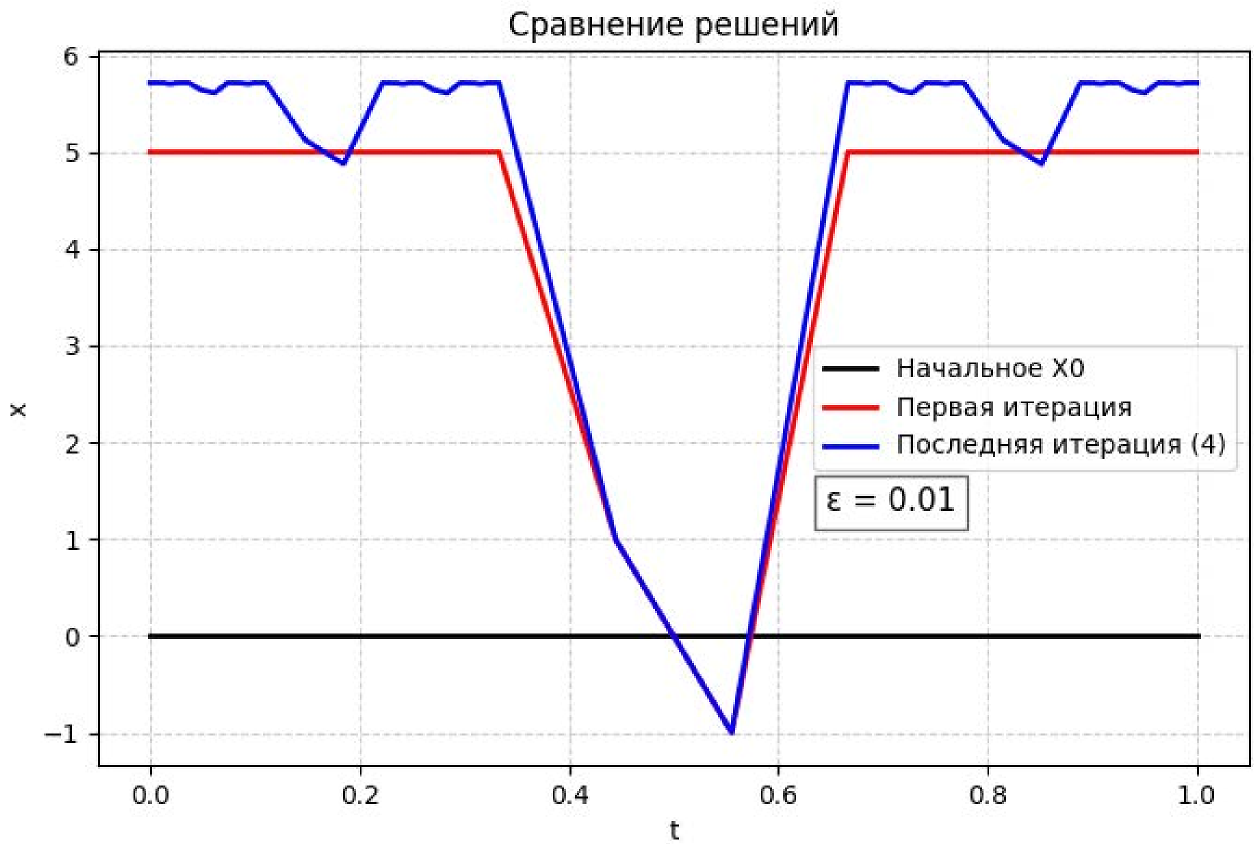
```python
import numpy as np
import matplotlib.pyplot as plt
import math
from typing import List

class SquashMapSolver:
    def __init__(self, density=1000, alpha=1/8):
        """
        Инициализирует класс.
        """
        self.density = density
        self.alpha = alpha
        self.T = np.linspace(0, 1, density)

    def apply_func(self, func: np.array, t: float) -> float:
        """
        Применяет функцию к заданному значению t, интерполируя по ближайшему индексу.
        """
        idx = np.searchsorted(self.T, t, side='right')
        return func[min(idx, len(func) - 1)]

    def F1(self, X: np.array, t: float) -> float:
        """
        Вычисляет F1(X, t) = 1/8 * x(3t) + 5.
        """
        return 1/8 * self.apply_func(X, 3*t) + 5

    def F2(self, X: np.array, t: float) -> float:
        """
        Вычисляет F2(X, t) = 1/8 * x(3t - 2) + 5.
        """
        return 1/8 * self.apply_func(X, 3*t - 2) + 5

    def between(self, x1: float, x2: float, p: float) -> float:
        """
        Линейно интерполирует значение между x1 и x2.
        """
        return x1 + (x2 - x1) * p

    def T_map(self, X: np.array) -> np.array:
        """
        Строит отображение T(x) для массива X.
        """
        T_X = np.zeros_like(X)
        for idx, t in enumerate(self.T):
            if t < 1/3:
                T_X[idx] = self.F1(X, t)
```

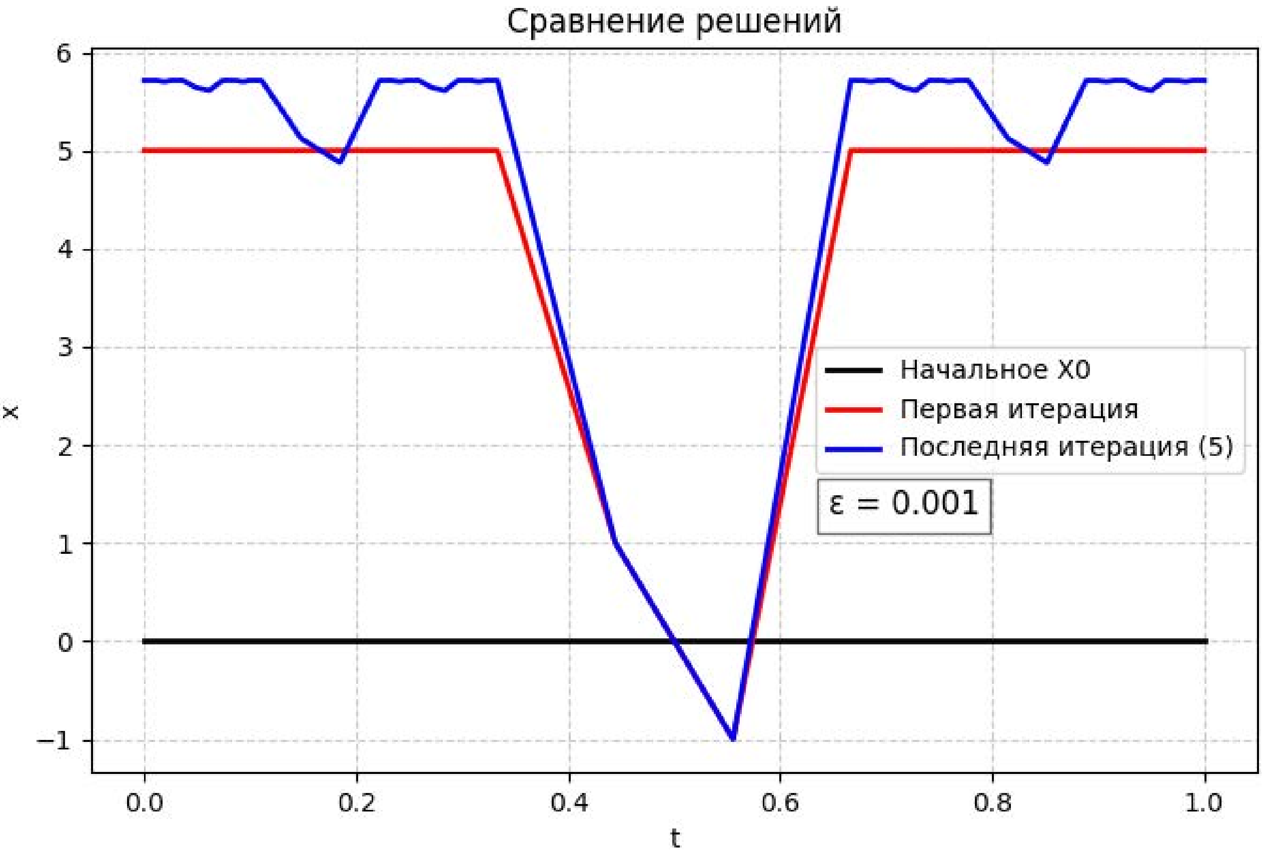
```python
            elif t < 2/3:
                if t < 4/9:
                    T_X[idx] = self.between(self.F1(X, 1/3), 1, (t - 1/3) / (1/9))
                elif t < 5/9:
                    T_X[idx] = self.between(1, -1, (t - 4/9) / (1/9))
                else:
                    T_X[idx] = self.between(-1, self.F2(X, 2/3), (t - 5/9) / (1/9))
            else:
                T_X[idx] = self.F2(X, t)
        return T_X

    def distance(self, X: np.array, Y: np.array) -> float:
        """
        Вычисляет максимальное расстояние между массивами X и Y.
        """
        return np.max(np.abs(X - Y))

    def iterations_count(self, d0: float, eps: float) -> int:
        """
        Вычисляет необходимое количество итераций метода сжимающего отображения.
        """
        return int(np.ceil(math.log((1 - self.alpha) * eps / d0, self.alpha)))

    def squash_map_method(self, X0: np.array, eps: float) -> tuple[np.array, int]:
        """
        Выполняет метод сжимающего отображения для X0 с точностью eps.
        Возвращает найденную фиксированную точку и количество итераций.
        """
        X1 = self.T_map(X0)
        d0 = self.distance(X0, X1)
        iter_count = self.iterations_count(d0, eps)
        X_I = np.copy(X1)
        for _ in range(iter_count - 1):
            X_I = self.T_map(X_I)
        return X_I, iter_count

    def plot_graph(self, Xs: List[np.array], labels: List[str], colors: List[str], eps: float, title: str):
        """
        Строит график для переданных массивов Xs.
        """
        plt.figure(figsize=(8, 5))
        for X, label, color in zip(Xs, labels, colors):
            plt.plot(self.T, X, label=label, color=color, linewidth=2)
        plt.xlabel("t")
        plt.ylabel("x")
        plt.title(title)
        plt.legend()
        plt.grid(True, linestyle='--', alpha=0.6)
        plt.text(0.632, 0.355, f"ε = {eps}", transform=plt.gca().transAxes, fontsize=12, bbox=dict(facecolor='white', alpha=0.6))
        plt.show()

def main():
    solver = SquashMapSolver()

    # Задание точности
    eps = 1e-2

    # Задание X0 = 0
    X0 = np.zeros_like(solver.T)  # Или np.copy(solver.T) для X0 = t

    X1 = solver.T_map(X0)
    X_final, iter_count = solver.squash_map_method(X0, eps)

    solver.plot_graph([X0, X1, X_final],
                      labels=["Начальное X0", "Первая итерация", f"Последняя итерация ({iter_count})"],
                      colors=["black", "red", "blue"],
                      eps=eps,
                      title="Сравнение решений")

if __name__ == '__main__':
    main()
```

Вывод программы и графики

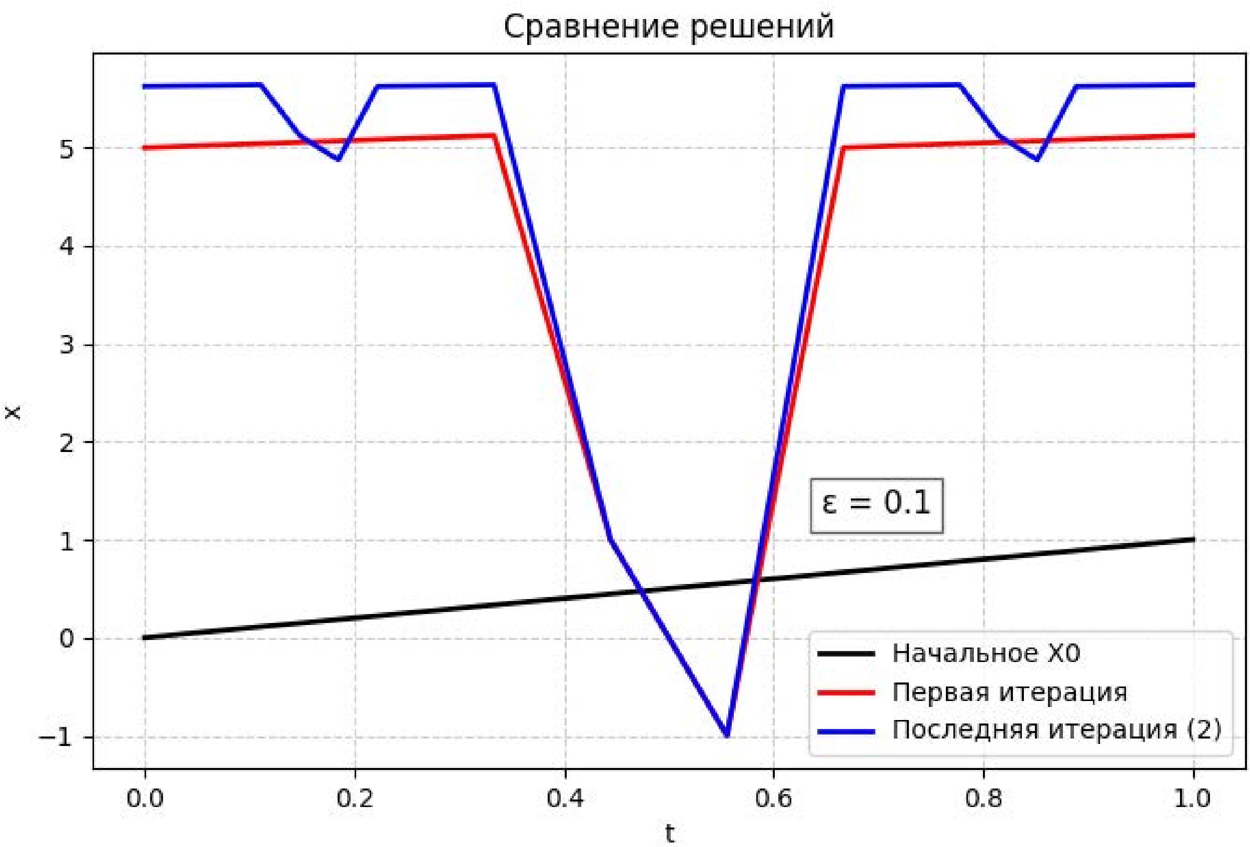
Рассмотрим графики неподвижной точки при начальном приближении x0 = 0 с разными значениями ε. Для сравнения добавим график, полученный на первой итерации метода сжимающего отображения.

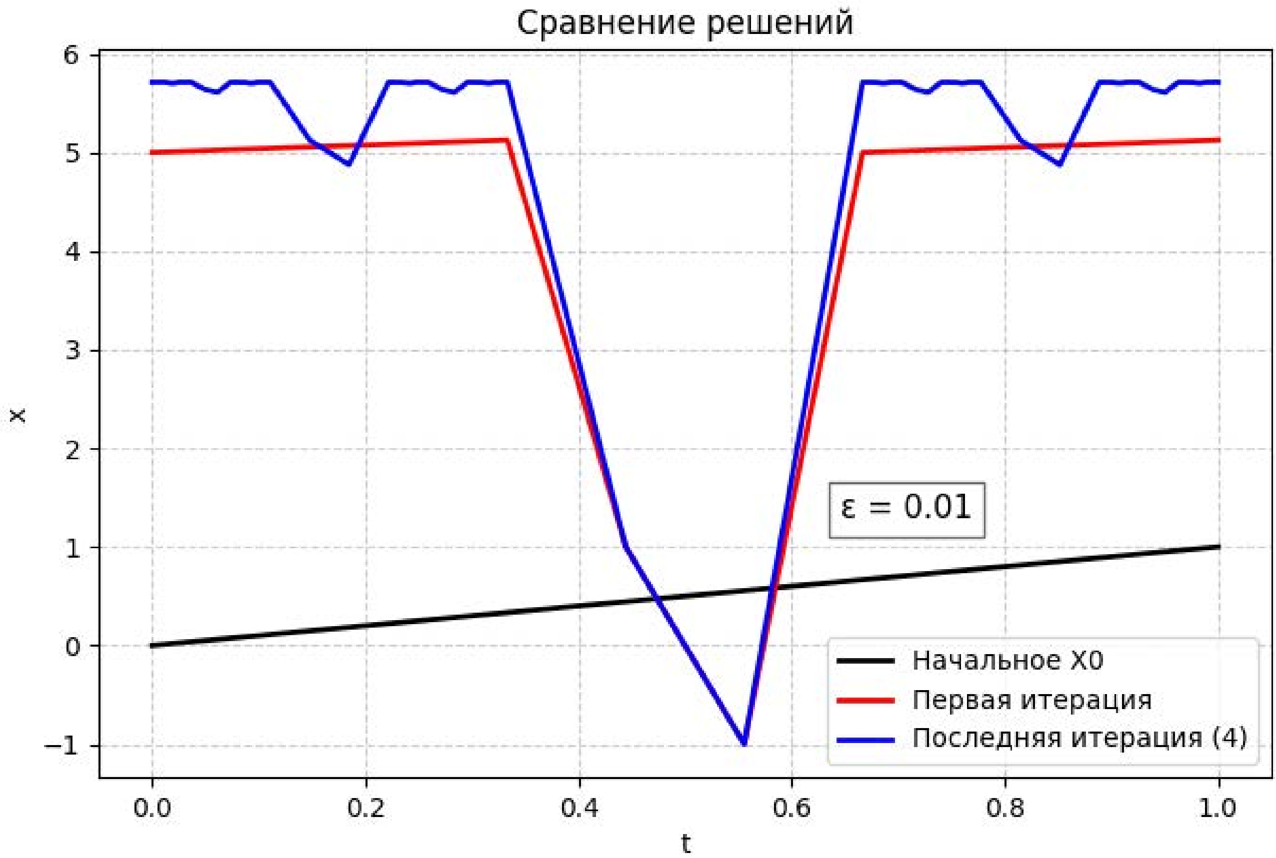


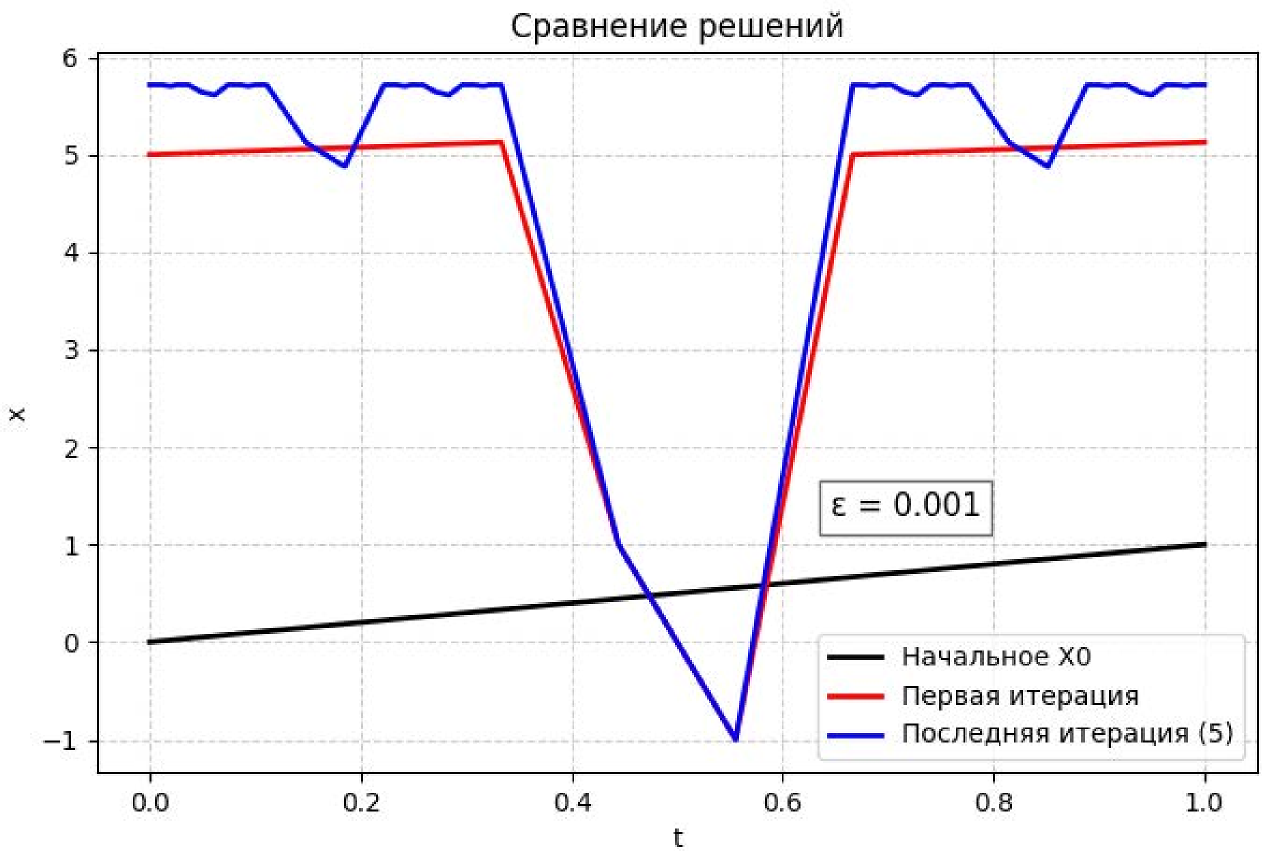




Теперь рассмотрим графики для неподвижной точки x0 = t:




Сравнение решений
x
t
ε = 0.01
Начальное X0
Первая итерация
Последняя итерация (4)
6
5
4
3
2
1
0
−1
0.0
0.2
0.4
0.6
0.8
1.0


Сравнение решений
x
t
ε = 0.001
Начальное X0
Первая итерация
Последняя итерация (5)
6
5
4
3
2
1
0
−1
0.0
0.2
0.4
0.6
0.8
1.0