РГР №1

Былькова Кристина Алексеевна

Группа М8О-308Б-22

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.

Вариант 2:

2) $T(x) = \begin{cases} \frac{1}{1+k}x(3t) - \frac{l}{2}, & 0 \leqslant t \leqslant \frac{1}{3}; \\ f(t), & \frac{1}{3} < t < \frac{2}{3}; \\ \frac{1}{1+k}x(3t-2), & \frac{2}{3} \leqslant t \leqslant 1, \end{cases}$ где $f(t)$ — аффинная функция такая, что $T(x)$ — непрерывная функция;

Решение:

$$T(x)=\begin{cases}\frac{1}{9}x(3t)-\frac{15}{2}, & 0\le t\le\frac{1}{3};\\ f(t), & \frac{1}{3}<t<\frac{2}{3};\ \text{где } f(t) - \\ \frac{1}{9}x(3t-2), & \frac{2}{3}\le t\le 1,\end{cases}$$

аффинная ф-я: $T(x)$ – непр. ф-я

$T: C[0;1]\to C[0;1]$

Покажем, что выражение в правой части яв-ся сжатием в $C[0;1]$:

для $f(t)$

$$\rho(g(x),g(y))=\max\Big\{\max_{t\in[0;\frac{1}{3}]}\Big|\frac{1}{9}x(3t)-\frac{15}{2}-\frac{1}{9}y(3t)+\frac{15}{2}\Big|,\ \ldots,$$

$3t=s,\ t\in[0;\frac{1}{3}]\Rightarrow s\in[0;1]$

$$\max_{t\in[\frac{2}{3};1]}\Big|\frac{1}{9}x(3t-2)-\frac{1}{9}y(3t-2)\Big|\Big\}\ (=)$$

$3t-2=s,\ t\in[\frac{2}{3};1]\Rightarrow s\in[0;1]$

$T(x)(t)=at+b$ $(f(t))$ (проверяем разрывы)

при $t=\frac{1}{3}$: $\frac{1}{9}x(3\cdot\frac{1}{3})-\frac{15}{2}=\frac{1}{9}x(1)-\frac{15}{2}$

при $t=\frac{2}{3}$: $\frac{1}{9}x(3\cdot\frac{2}{3}-2)=\frac{1}{9}x(0)$

$$a=\frac{T(x)(\frac{2}{3})-T(x)(\frac{1}{3})}{\frac{2}{3}-\frac{1}{3}}=\frac{0-(-\frac{15}{2})}{\frac{1}{3}}=22{,}5$$

$$b=T(x)(\tfrac{1}{3})-a\cdot\tfrac{1}{3}=-\tfrac{15}{2}-\tfrac{45}{2}\cdot\tfrac{1}{3}=-15$$

$$(=)\ \max_{s\in[0;1]}\Big|\frac{1}{9}x(s)-\frac{1}{9}y(s)\Big|=\frac{1}{9}\max_{s\in[0;1]}|x(s)-y(s)|,\ \alpha=\frac{1}{9}\Rightarrow$$

$\Rightarrow \alpha\in[0;1]\Rightarrow$ отображение сжимающее.

Для нахождения числа итераций для достижения точности ε (0,1 ; 0,01 ; 0,001) используем:

$$\|x_n - x^*\| \leq \frac{\alpha^n}{1-\alpha}\|x_0 - x_1\|$$

$$\frac{\alpha^n}{1-\alpha}\|x_0 - x_1\| \leq \varepsilon$$

$$\left(\frac{1}{9}\right)^n \leq \frac{8}{9}\varepsilon \Rightarrow n \geq \frac{\ln\left(\frac{8}{9}\varepsilon\right)}{\ln\left(\frac{1}{9}\right)}$$

1) $\varepsilon = 0{,}1$

$$n \geq \frac{\ln\left(\frac{8}{9}\cdot 0{,}1\right)}{\ln\left(\frac{1}{9}\right)} \approx 1{,}1 \Rightarrow n \geq 2$$

2) $\varepsilon = 0{,}01$

$$n \geq \frac{\ln\left(\frac{8}{9}\cdot 0{,}01\right)}{\ln\left(\frac{1}{9}\right)} \approx 2{,}15 \Rightarrow n \geq 3$$

3) $\varepsilon = 0{,}001$

$$n \geq \frac{\ln\left(\frac{8}{9}\cdot 0{,}001\right)}{\ln\left(\frac{1}{9}\right)} \approx 3{,}2 \Rightarrow n \geq 4$$

Код:

```
import numpy as np
import matplotlib.pyplot as plt

# Начальное приближение x0(t)
def x0(t):
    return t
    #return -7.5
    #return -4

# Система T(x)
def T(x, t):
    if 0 <= t <= 1/3:
        return (1/9) * x(3 * t) - 15/2
    elif 1/3 < t < 2/3:
        t1, t2 = 1/3, 2/3
        y1, y2 = T(x, t1), T(x, t2)  # значения на границах y1 = -15/2, y2 = 0
        a = (y2 - y1) / (t2 - t1)  # коэффициент наклона a = 22,5
        b = y1 - a * t1  # свободный член b = -15
        return a * t + b
    elif 2/3 <= t <= 1:
        return (1/9) * x(3 * t - 2)

# Итеративное применение оператора T
def find_fixed_point(x_init, epsilon=1e-6, max_iter=100):
    t_values = np.linspace(0, 1, 1000)  # Значения t
    x_current = x_init  # Текущая функция
    history = []  # Для хранения промежуточных итераций

    for i in range(max_iter):
        x_next_values = np.array([T(x_current, t) for t in t_values])  # Применяем
T
        history.append((t_values, x_next_values.copy()))  # Сохраняем итерацию

        # Проверяем сходимость (норма разницы между итерациями < эпсилон)
        if i > 0:
            diff = np.max(np.abs(x_next_values - history[-2][1]))
            print(f"Итерация {i}: diff = {diff:.10f}")  # Выводим разницу
            if diff < epsilon:
                print(f"Сходимость достигнута на итерации {i}")
                break

        x_current = lambda t: x_next_values[np.argmin(np.abs(t_values - t))]  #
Обновляем текущую функцию

    else:
```

```
        print("Достигнуто максимальное число итераций")

    return history  # Возвращаем историю итераций

def main():
    epsilon_values = [1e-1, 1e-2, 1e-3]

    for epsilon in epsilon_values:
        print(f"\nЗапуск для epsilon = {epsilon}")
        history = find_fixed_point(x0, epsilon=epsilon)

        plt.figure(figsize=(12, 6))

        # Левый подграфик: начальная и промежуточные итерации
        plt.subplot(1, 2, 1)

        # Добавляем начальную итерацию (нулевая итерация)
        t_values = history[0][0]
        x_init_values = np.array([x0(t) for t in t_values])
        plt.plot(t_values, x_init_values, label='Итерация 0', color='blue',
linestyle='--', linewidth=2)

        # Добавляем промежуточные итерации
        for i, (t_values, x_values) in enumerate(history[1:]):
            plt.plot(t_values, x_values, label=f'Итерация {i + 1}')

        plt.axvline(x=1/3, color='gray', linestyle='--', label='t = 1/3')
        plt.axvline(x=2/3, color='black', linestyle='--', label='t = 2/3')
        plt.title(f'epsilon = {epsilon}\nНачальная и промежуточные итерации')
        plt.xlabel('t')
        plt.ylabel('x(t)')
        plt.xticks(np.arange(0, 1.1, 0.1))
        plt.yticks(np.arange(-9, 1, 0.5))
        plt.xlim(0, 1)
        plt.grid(True)
        plt.legend()

        # Правый подграфик: финальная итерация
        plt.subplot(1, 2, 2)
        t_values, x_final = history[-1]
        plt.plot(t_values, x_final, label='Финальная итерация', color='red',
linewidth=2)
        plt.axvline(x=1/3, color='gray', linestyle='--', label='t = 1/3')
        plt.axvline(x=2/3, color='black', linestyle='--', label='t = 2/3')
        plt.title(f'epsilon = {epsilon}\nФинальная итерация')
        plt.xlabel('t')
        plt.ylabel('x(t)')
```

```
        plt.xticks(np.arange(0, 1.1, 0.1))
        plt.yticks(np.arange(-9, 1, 0.5))
        plt.xlim(0, 1)
        plt.grid(True)
        plt.legend()

        plt.tight_layout()
        plt.show()

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

Вывод:

$ python main.py

Запуск для epsilon = 0.1

Итерация 1: diff = 0.8580246914

Итерация 2: diff = 0.0953360768

Сходимость достигнута на итерации 2

Запуск для epsilon = 0.01

Итерация 1: diff = 0.8580246914

Итерация 2: diff = 0.0953360768

Итерация 3: diff = 0.0105846587

Итерация 4: diff = 0.0011760732

Сходимость достигнута на итерации 4

Запуск для epsilon = 0.001

Итерация 1: diff = 0.8580246914

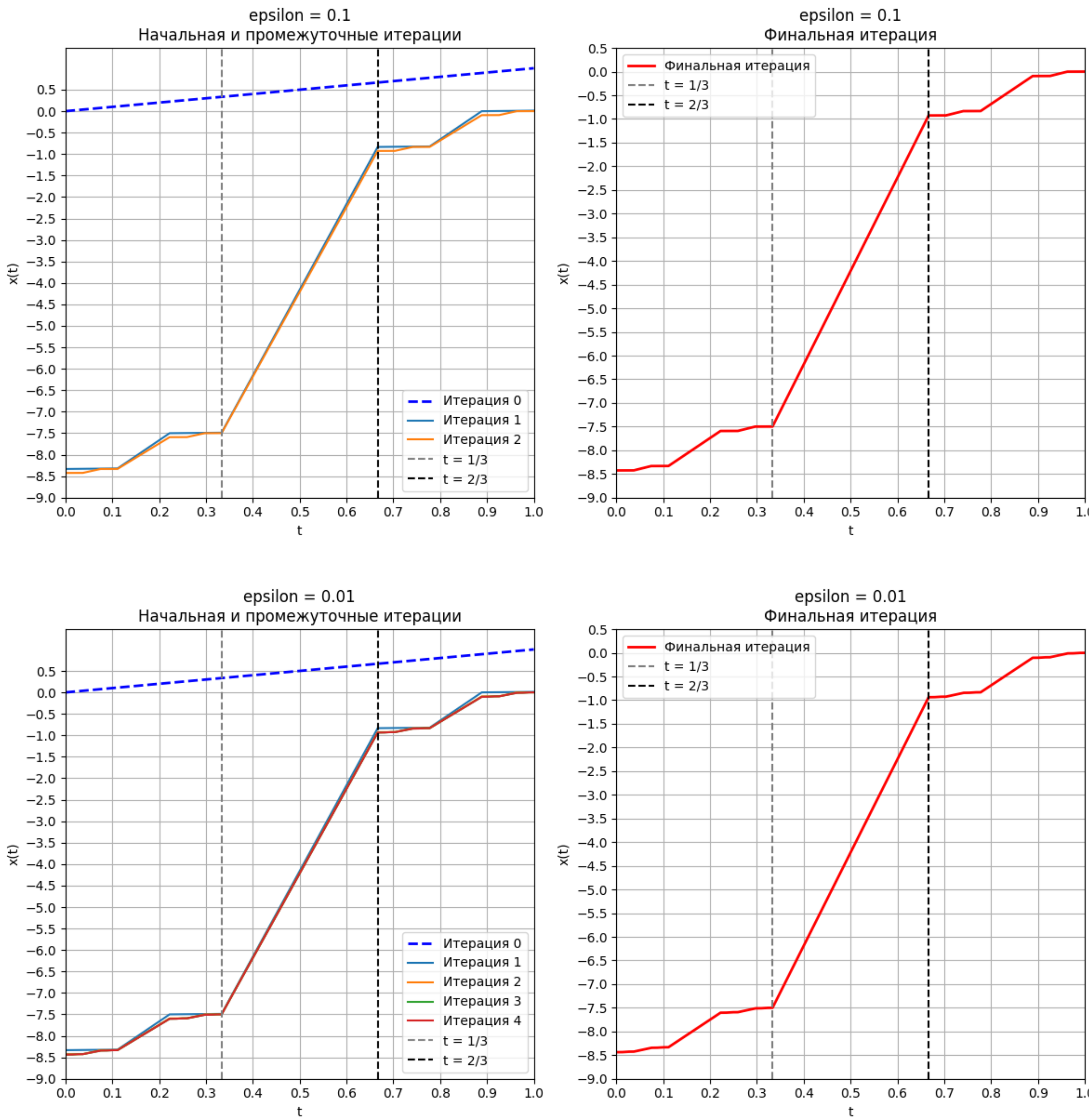
Итерация 2: diff = 0.0953360768

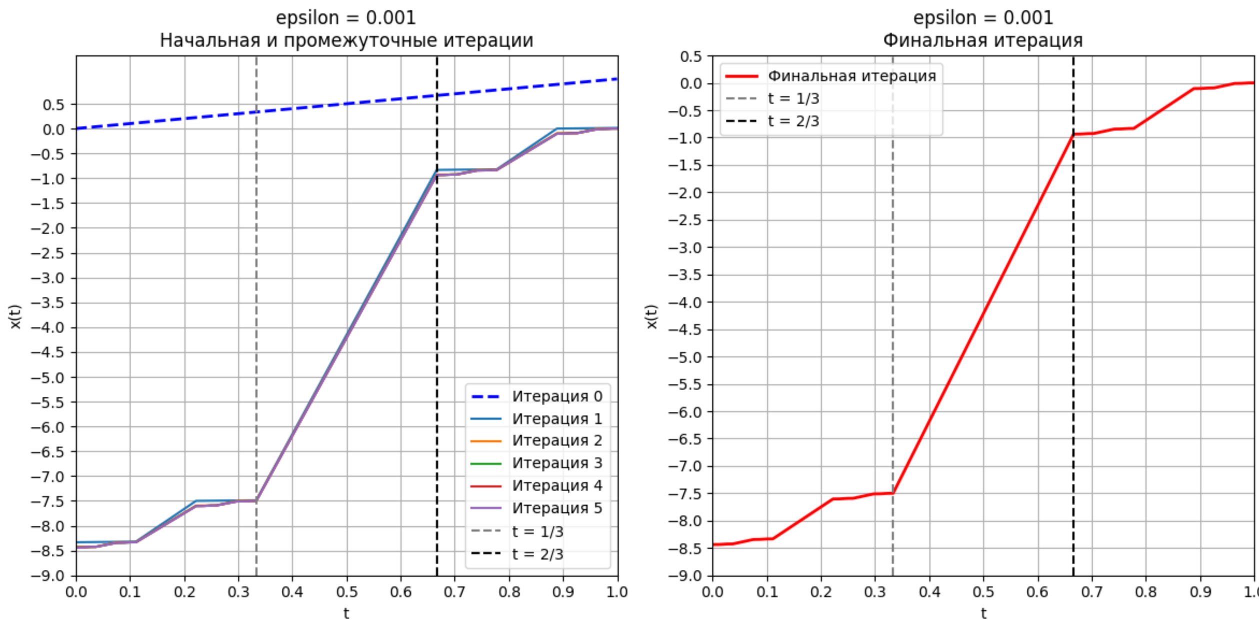
Итерация 3: diff = 0.0105846587

Итерация 4: diff = 0.0011760732

Итерация 5: diff = 0.0001306748

Сходимость достигнута на итерации 5


epsilon = 0.1
Начальная и промежуточные итерации
Итерация 0
Итерация 1
Итерация 2
t = 1/3
t = 2/3
x(t)
t
epsilon = 0.1
Финальная итерация
Финальная итерация
t = 1/3
t = 2/3
x(t)
t
epsilon = 0.01
Начальная и промежуточные итерации
Итерация 0
Итерация 1
Итерация 2
Итерация 3
Итерация 4
t = 1/3
t = 2/3
x(t)
t
epsilon = 0.01
Финальная итерация
Финальная итерация
t = 1/3
t = 2/3
x(t)
t



На левых подграфиках изображены начальная и промежуточные итерации, на правом – финальная итерация. Для epsilon = 0,1 сходимость достигается за 2 итерации, для epsilon = 0,01 – за 4 итерации, а для epsilon = 0,001 – за 5 итераций. Запуск производился для начального приближения = t.

Запуск для начального приближения = -7.5 (выбрано, так как данное значение близко к значению T(x) на интервале 0 <= t <= 1/3):

$ python main.py

Запуск для epsilon = 0.1

Итерация 1: diff = 0.7407407407

Итерация 2: diff = 0.0823045267

Сходимость достигнута на итерации 2

Запуск для epsilon = 0.01

Итерация 1: diff = 0.7407407407

Итерация 2: diff = 0.0823045267

Итерация 3: diff = 0.0091449474

Сходимость достигнута на итерации 3

Запуск для epsilon = 0.001

Итерация 1: diff = 0.7407407407

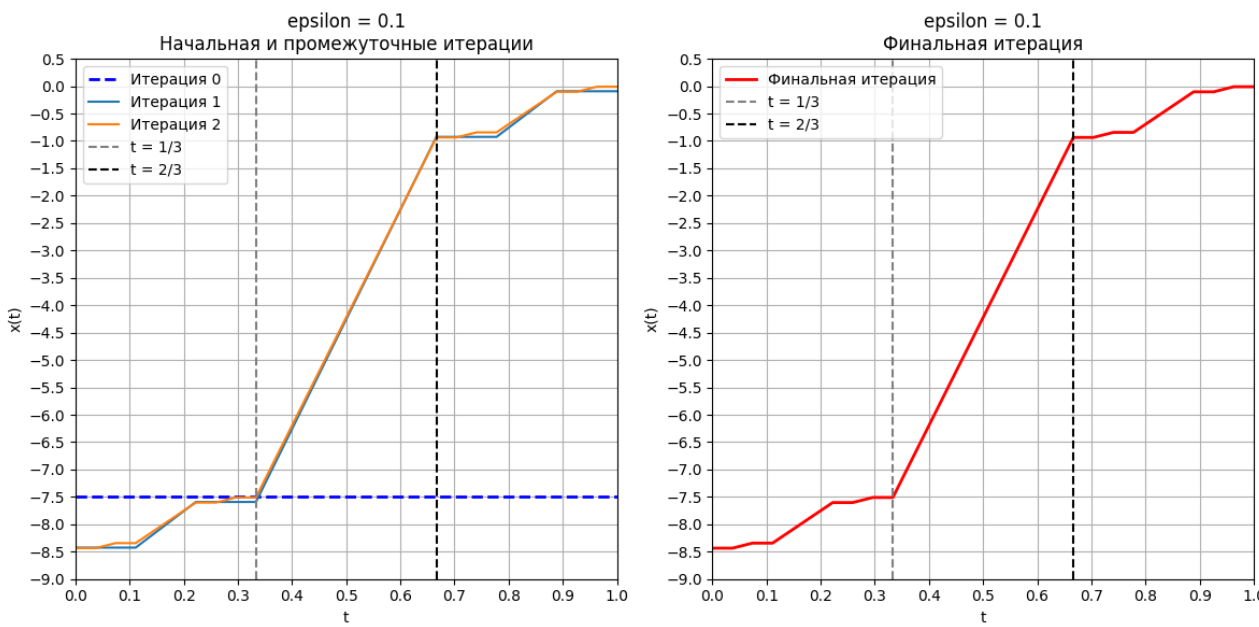
Итерация 2: diff = 0.0823045267

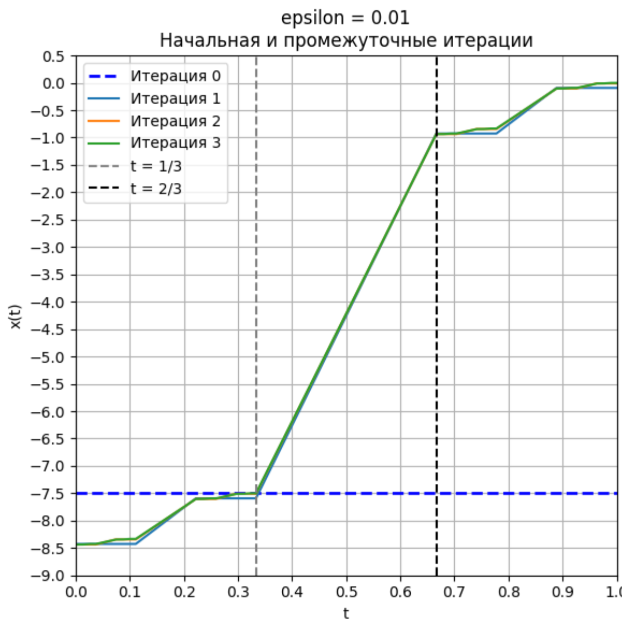
Итерация 3: diff = 0.0091449474

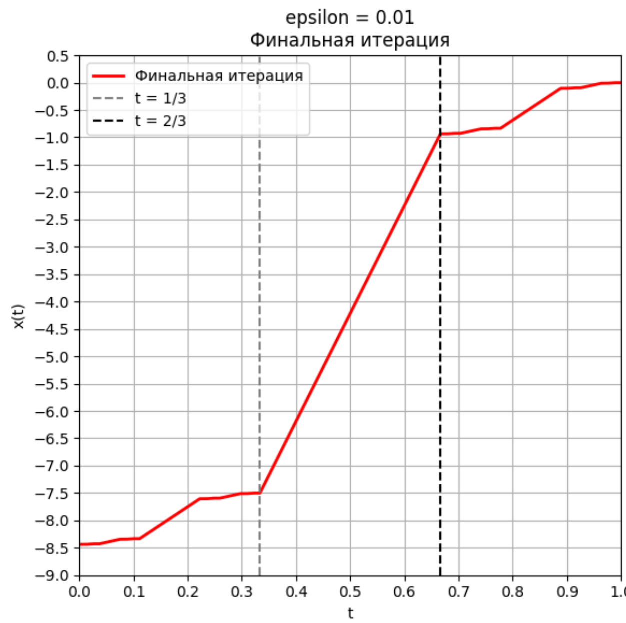
Итерация 4: diff = 0.0010161053

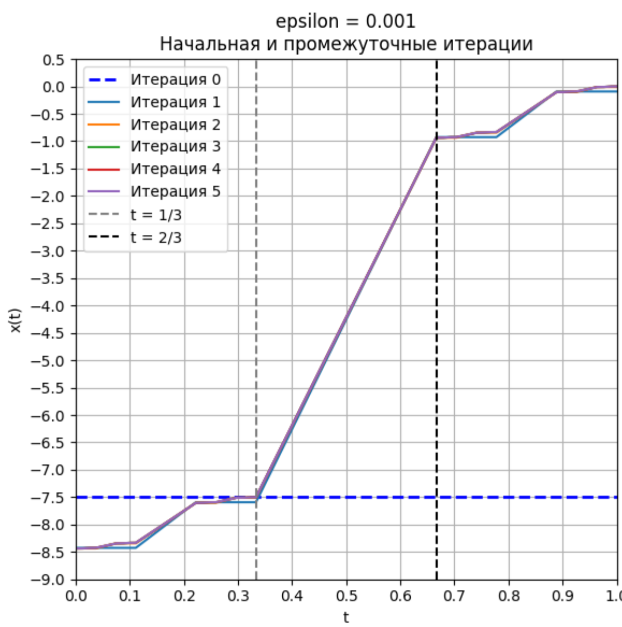
Итерация 5: diff = 0.0001129006

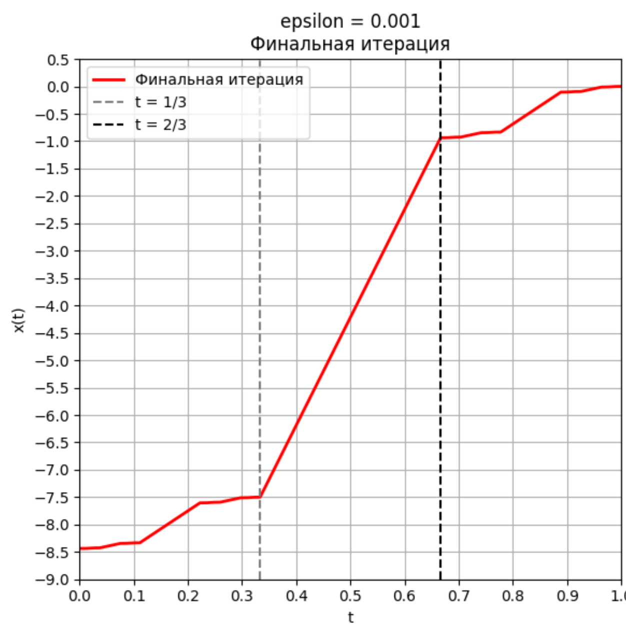
Сходимость достигнута на итерации 5











При данном начальном приближении для epsilon = 0,1 сходимость достигается за 2 итерации, для epsilon = 0,01 – уже за 3 итерации, а для epsilon = 0,001 – за 5 итераций.

Запуск для начального приближения = -4:

$ python main.py

Запуск для epsilon = 0.1

Итерация 1: diff = 0.4382716049

Итерация 2: diff = 0.0486968450

Сходимость достигнута на итерации 2

Запуск для epsilon = 0.01

Итерация 1: diff = 0.4382716049

Итерация 2: diff = 0.0486968450

Итерация 3: diff = 0.0054107606

Сходимость достигнута на итерации 3

Запуск для epsilon = 0.001

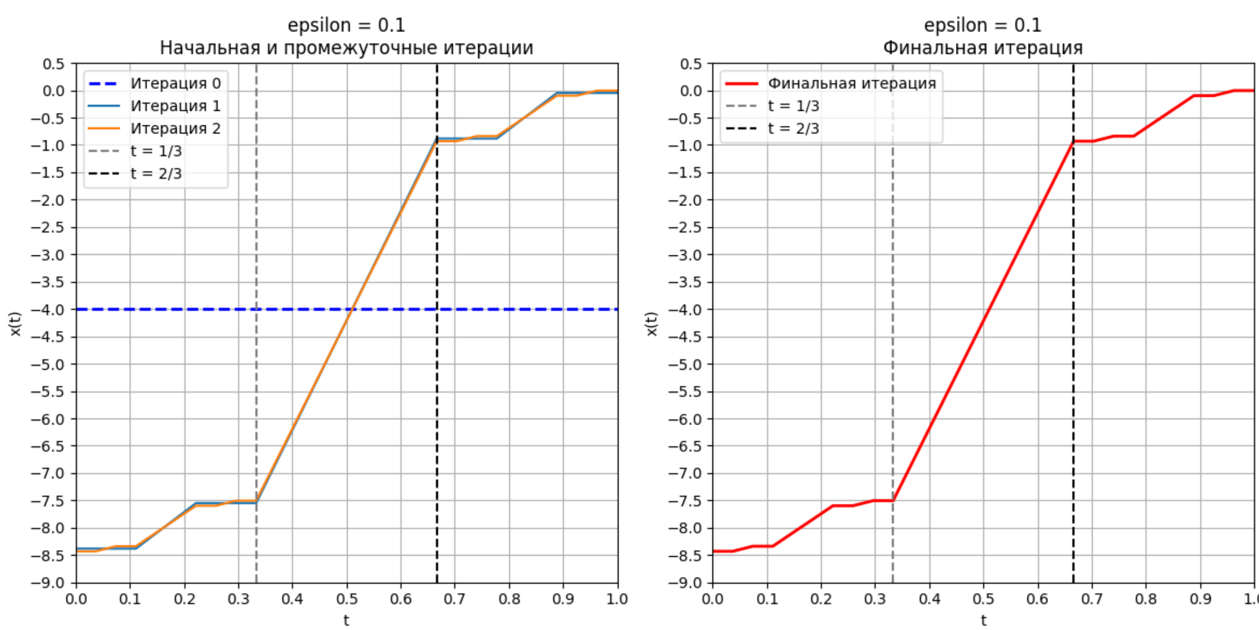
Итерация 1: diff = 0.4382716049

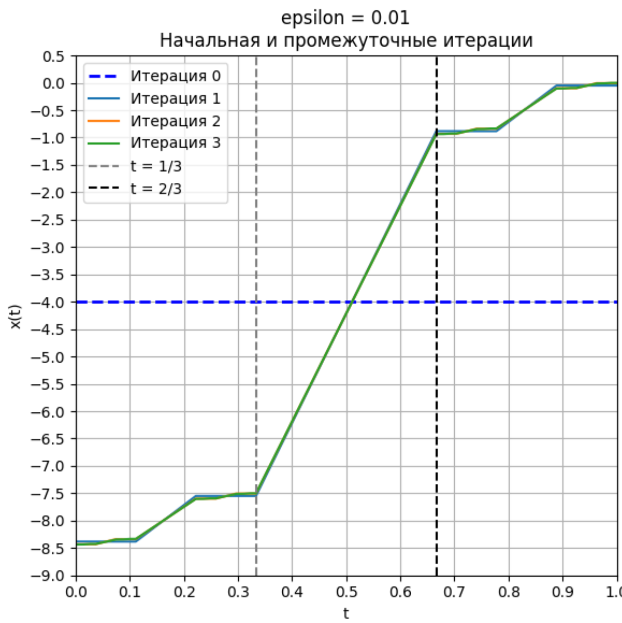
Итерация 2: diff = 0.0486968450

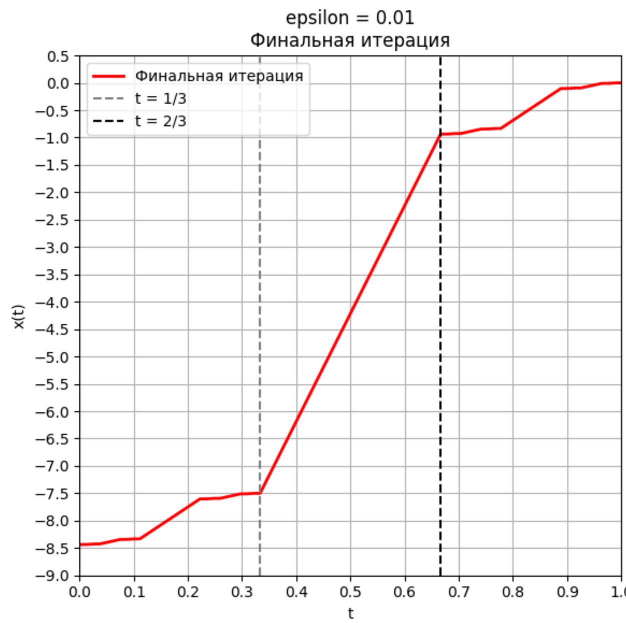
Итерация 3: diff = 0.0054107606

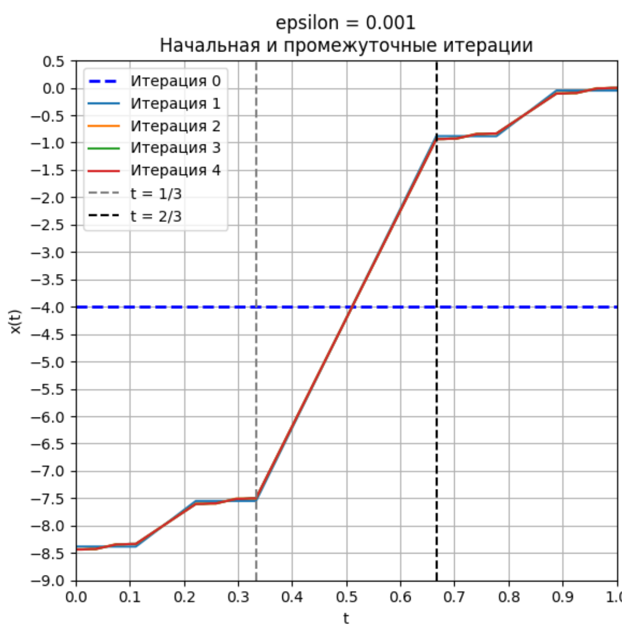
Итерация 4: diff = 0.0006011956

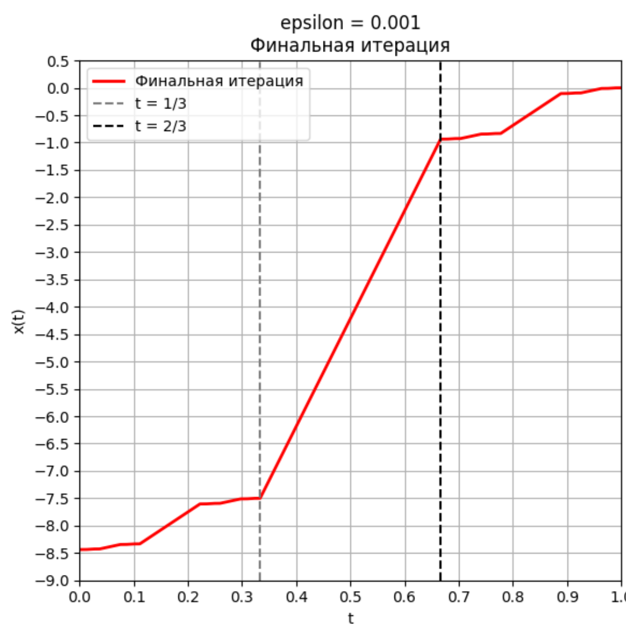
Сходимость достигнута на итерации 4











При данном начальном приближении для epsilon = 0,1 сходимость достигается за 2 итерации, для epsilon = 0,01 – за 3 итерации, а для epsilon = 0,001 – за 4 итераций, что является наилучшей скоростью сходимости (опираясь на теоретическое решение).

Таким образом, выбор начального приближения значительно влияет на скорость сходимости метода сжимающих отображений.