Министерство высшего образования и науки Российской Федерации

Национальный научно-исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №2

по дисциплине

«Вычислительная математика»

Вариант 11

Работу выполнил:

Макеев Роман Ильич

Группа P3208

Преподаватель:

Машина Екатерина Алексеевна

Санкт-Петербург

2024

Цель работы:

Изучить численные методы решения нелинейных уравнений и их систем, найти корни заданного нелинейного уравнения/системы нелинейных уравнений, выполнить программную реализацию методов.

Порядок выполнения работы:

1. Решение нелинейного уравнения

$$4.45x^3 + 7.81x^2 - 9.62x - 8.17$$

- Уточнение правого, левого и центрального корней уравнения методами половинного деления, хорд и простой итерации соответсвенно с точность до $\epsilon = 10^{-2}$

2. Решение системы нелинейных уравнений методом Ньютона

$$\begin{cases} tg(xy + 0.2) = x^2 \\ x^2 + 2y^2 = 1 \end{cases}$$

3. Программная реализация решения нелинейных уравнений и систем

- Для уравнений методами половинного деления, секущих и простой итерации

- Для систем методом простой итерации

Рабочие формулы используемых методов:

- Уточнение корня методом половинного деления

$$x_i = \frac{a_i + b_i}{2}$$

- Уточнение корня методом хорд

$$x_i = a_i - \frac{b_i - a_i}{f(b_i) - f(a_i)} f(a_i)$$

- Уточнение корня методом простой итерации

$$x_{i+1} = \phi(x_i)$$

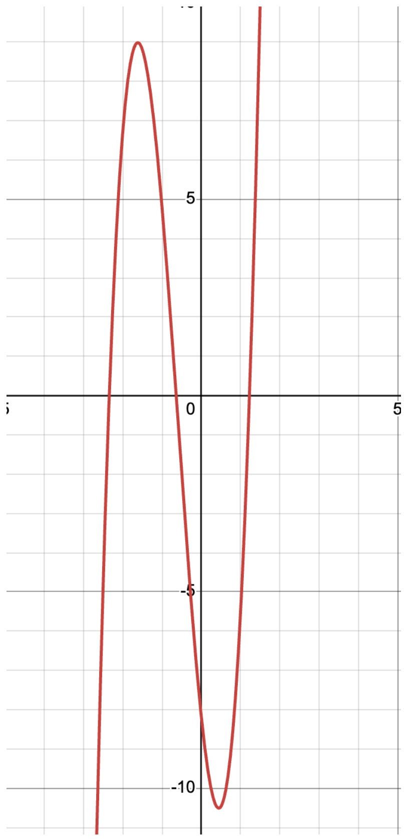
- Достаточное условие сходимости метода простой итерации

$$|\phi'(x)_{[a;b]}| < 1$$

Вычислительная часть:

1. Решение уравнения

$$4.45x^3 + 7.81x^2 - 9.62x - 8.17$$



- Нахождение правого корня методом половинного деления

№ шага	a	b	x	$f(a)$	$f(b)$	$f(x)$	$\|a-b\|$
1	1	2	1.5	-5.53	39.43	9.991	1
2	1	1.5	1.25	-5.53	9.991	0.699	0.5
3	1	1.25	1.125	-5.53	0.699	-2.772	0.25
4	1.125	1.25	1.188	-2.772	0.699	-1.115	0.125
5	1.188	1.25	1.219	-1.115	0.699	-0.231	0.062
6	1.219	1.25	1.235	-0.231	0.699	0.244	0.031
7	1.219	1.235	1.227	-0.231	0.244	0.0049	0.016
8	1.219	1.227	1.223	-0.231	0.0049	-0.113	0.008

$$x_3 \approx 1.22$$

- Нахождение левого корня методом хорд

№ шага	a	b	x	$f(a)$	$f(b)$	$f(x)$	$\lvert x_{k+1} - x_k \rvert$
1	-3	-2	-2.187	-29.17	6.71	3.675	-
2	-3	-2.187	-2.278	-29.17	3.675	1.669	0.091
3	-3	-2.278	-2.317	-29.17	1.669	0.695	0.039
4	-3	-2.317	-2.333	-29.17	0.695	0.278	0.016
5	-3	-2.333	-2.339	-29.17	0.278	0.108	0.006

$$x_1 \approx -2.34$$

- Нахождение центрального корня методом простой итерации

Для уравнения $4.45x^3 + 7.81x^2 - 9.62x - 8.17$:

$$\phi(x) = 0.463x^3 + 0.812x^2 - 0.849$$

$$\phi'(x) = 1.389x^2 + 1.624x$$

$$a = -1 \quad b = 0$$

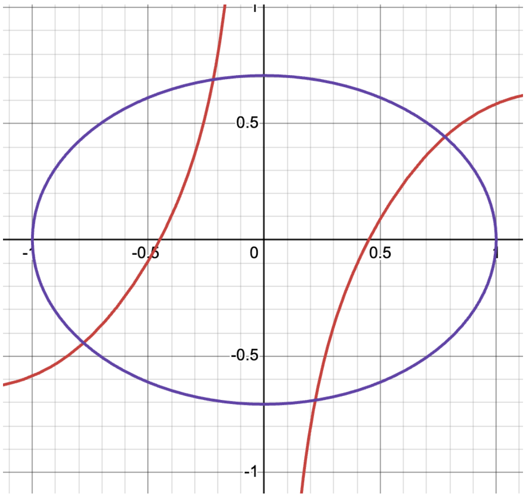
$max(\lvert\phi'(x)_{[a;b]}\rvert) = \phi'(-1) = 0.235$ => сходится

№ итерации	x_k	x_{k+1}	$f(x_{k+1})$	$\lvert x_{k+1} - x_k \rvert$
1	-1	-0.5	-1.964	0.5
2	-0.5	-0.704	0.922	0.204
3	-0.704	-0.608	-0.43	0.096
4	-0.608	-0.653	0.204	0.045
5	-0.653	-0.632	-0.095	0.021
6	-0.632	-0.642	0.044	0.01
7	-0.642	-0.637	-0.022	0.005

$$x_1 \approx -0.64$$

2. Решение системы уравнений

$$\begin{cases} tg(xy + 0.2) = x^2 \\ x^2 + 2y^2 = 1 \end{cases}$$



4 решения

$$\begin{cases} tg(xy+0.2) - x^2 = 0 \\ x^2 + 2y^2 - 1 = 0 \end{cases} \qquad \frac{\delta g}{\delta x} = 2x \quad \frac{\delta g}{\delta y} = 4y$$

$$\frac{\delta f}{\delta x} = y \cdot tg(xy+0.2)^2 + y - 2x \qquad \frac{\delta f}{\delta y} = x \cdot tg(xy+0.2)^2 + x$$

Итеративно решаем систему:

$$\begin{cases} (y \cdot tg(xy+0.2)^2 + y - 2x) \cdot \Delta x + (x \cdot tg(xy+0.2)^2 + x) \cdot \Delta y = -tg(xy+0.2) + x^2 \\ 2x \cdot \Delta x + 4y \cdot \Delta y = -x^2 - 2y^2 + 1 \end{cases}$$

Решение 1 (правая верхняя точка):

$$\begin{cases} x_0 = 1 \\ y_0 = 0.5 \end{cases} \quad \begin{cases} \Delta x_0 = -0.205 \\ \Delta y_0 = -0.045 \end{cases} \quad \begin{cases} x_1 = 0.795 \\ y_1 = 0.455 \end{cases}$$

$$\begin{cases} \Delta x_1 = -0.016 \\ \Delta y_1 = -0.011 \end{cases} \quad \begin{cases} x_2 = 0.779 \\ y_2 = 0.444 \end{cases} \quad \begin{cases} \Delta x_2 = 0 \\ \Delta y_2 = 0 \end{cases}$$

$$(0.779; 0.444)$$

Решение 2 (левая верхняя точка):

$$\begin{cases} x_0 = 0 \\ y_0 = 1 \end{cases} \quad \begin{cases} \Delta x_0 = -0.195 \\ \Delta y_0 = -0.25 \end{cases} \quad \begin{cases} x_1 = -0.195 \\ y_1 = 0.75 \end{cases}$$

$$\begin{cases} \Delta x_1 = -0.024 \\ \Delta y_1 = -0.057 \end{cases} \quad \begin{cases} x_2 = -0.219 \\ y_2 = 0.693 \end{cases} \quad \begin{cases} \Delta x_2 = 0 \\ \Delta y_2 = 0.003 \end{cases}$$

$$(-0.219; 0.696)$$

Решение 3 (левая нижняя точка):

$$\begin{cases} x_0 = -1 \\ y_0 = -0.5 \end{cases} \quad \begin{cases} \Delta x_0 = 0.205 \\ \Delta y_0 = 0.045 \end{cases} \quad \begin{cases} x_1 = -0.795 \\ y_1 = -0.455 \end{cases}$$

$$\begin{cases} \Delta x_1 = 0.016 \\ \Delta y_1 = 0.011 \end{cases} \quad \begin{cases} x_2 = -0.779 \\ y_2 = -0.444 \end{cases} \quad \begin{cases} \Delta x_2 = 0 \\ \Delta y_2 = 0 \end{cases}$$

$$(-0.779; -0.444)$$

Решение 4 (правая нижняя точка):

$$\begin{cases} x_0 = 0 \\ y_0 = -0.5 \end{cases} \quad \begin{cases} \Delta x_0 = 0.389 \\ \Delta y_0 = -0.25 \end{cases} \quad \begin{cases} x_1 = 0.389 \\ y_1 = -0.75 \end{cases}$$

$$\begin{cases} \Delta x_1 = -0.145 \\ \Delta y_1 = 0.055 \end{cases} \quad \begin{cases} x_2 = 0.244 \\ y_2 = -0.695 \end{cases} \quad \begin{cases} \Delta x_2 = -0.024 \\ \Delta y_2 = 0.005 \end{cases}$$

$$\begin{cases} x_3 = 0.22 \\ y_2 = -0.69 \end{cases} \quad \begin{cases} \Delta x_3 = 0 \\ \Delta y_3 = 0 \end{cases}$$

$$(0.22; -0.69)$$

Листинг программы:

Метод половинного деления:

```
def mid_div_method(eq, data: MethodData) -> MethodResul| None:
    counter = 0
    a: float = data.a
    b: float = data.b
    x: float = (a + b) / 2
    rising: bool = eq.get_res(a) < 0
    while abs(a - b) > data.e:
        f_x = eq.get_res(x)
        if (f_x < 0 and rising) or (f_x > 0 and not rising):
            a = x
        else:
            b = x

        x = (a + b) / 2
        counter += 1

        if counter > 1000:
            print('Too many iterations!')
            return

    return MethodResult(Point(x, eq.get_res(x)), counter)
```

Метод секущих:

```python
def secant_method(eq, data: MethodData) -> MethodResult | None:
    counter = 0
    x_last: float = data.a
    x: float = data.b
    f_xlast = eq.get_res(x_last)
    f_x = eq.get_res(x)
    while abs(x_last - x) > data.e:
        x_next = x - ((x - x_last) * f_x / (f_x - f_xlast))
        x_last = x
        x = x_next
        f_xlast = eq.get_res(x_last)
        f_x = eq.get_res(x)
        counter += 1

        if counter > 1000:
            print('Too many iterations!')
            return

    return MethodResult(Point(x, eq.get_res(x)), counter)
```

Метод простой итерации (одно уравнение):

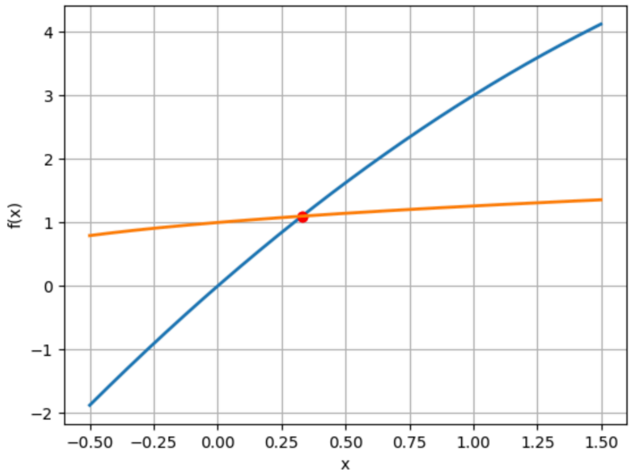
```python
def simple_it_method(eq, data: MethodData) -> MethodResult | None:
    phi = eq.create_phi_func(data.a, data.b)
    if phi is None:
        print("Can't solve it by this method")
        return

    x_last: float = data.a
    x: float = data.b
    counter: int = 0
    while abs(x_last - x) > data.e:
        x_next: float = phi(x)
        x_last = x
        x = x_next
        counter += 1

        if counter > 1000:
            print('Too many iterations!')
            return

    return MethodResult(Point(x, eq.get_res(x)), counter)
```

Метод простой итерации (система):

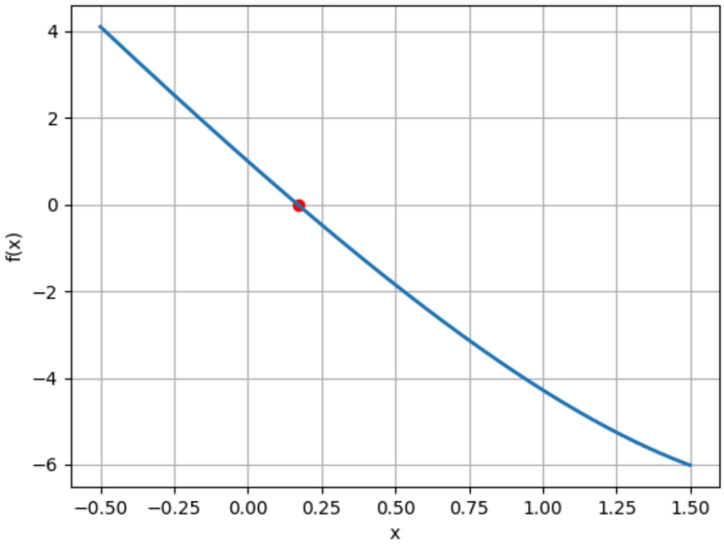
```python
def sys_simple_it_method(eq, data: MethodData) -> MethodResult | None:
    if not check_sys_conv(eq.phi1_data, eq.phi2_data, data):
        print("Can't solve it by this method")
        return None

    x_last: float = data.a
    x: float = data.b
    y: float = data.b_y
    counter: int = 0
    while abs(x_last - x) > data.e:
        x_last = x
        y_last = y
        x = eq.phi1_data.phi(x_last, y_last)
        y = eq.phi2_data.phi(x_last, y_last)
        counter += 1

        if counter > 1000:
            print('Too many iterations!')
            return

    return MethodResult(Point(x, y), counter)
```

Примеры и результаты работы программы:

```
Do you want to solve one equation (1) or system (2)? [1/2] -> 2
    1:
         +--
         | x^2 + 2y = 7x
        -+
         | y^3 - x = 1
         +--

    2:
         +--
         | sin(x) = 4 + x + y
        -+
         | 5y = x^3 + x + 1
         +--

Choose equation system: [1/2] -> 1
    1: Метод простых итераций
Choose method: [1] -> 1
Parse data for solving (a_x, b_x, a_y, b_y, e) from file? [n/filename] -> n
Input 5 float numbers joined spaces -> 0 1 1 2 1E-10
Point(0.3297147865784828, 1.099645832712799) It=22
```




```
Do you want to solve one equation (1) or system (2)? [1/2] -> 1
    1: 3x^3 + 3 = 8x
    2: e^x = 7x
    3: 10log(x^x) = x - 1
Choose equation: [1/2/3] -> 2
    1: Метод половинного деления
    2: Метод секущих
    3: Метод простых итераций
Choose method: [1/2/3] -> 2
Parse data for solving (a, b, e) from file? [n/filename] -> n
Input 3 float numbers (a, b, e) joined spaces -> 0 1 1E-5
Point(0.16919260528932212, -8.334393397646522e-09) It=4
```



Выводы:

В этой работе я изучил численные методы решения нелинейных уравнений и их систем, сделал программную реализацию нескольких методов.