НИУ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Вычислительная математика»

**Отчет**

По лабораторной работе №2 «Численное решение нелинейных уравнений и систем»

Вариант 27

Выполнил:

*студент группы P32131*

*Овсянников Роман Дмитриевич*

Преподаватель:

*Малышева Татьяна Алексеевна*

Санкт-Петербург,

2023 г.

**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:**

*Вычислительная реализация****:***

1. Отделить корни заданного нелинейного уравнения графически (вид уравнения

представлен в табл. 6)

2. Определить интервалы изоляции корней.

3. Уточнить корни нелинейного уравнения (см. табл. 6) с точностью ε=10^-2.

4. Используемые методы для уточнения каждого из 3-х корней многочлена представлены в таблице 7.

5. Вычисления оформить в виде таблиц (1-5), в зависимости от заданного метода.

Для всех значений в таблице удержать 3 знака после запятой.

5.1 Для метода половинного деления заполнить таблицу 1.

5.2 Для метода хорд заполнить таблицу 2.

5.3 Для метода Ньютона заполнить таблицу 3.

5.4 Для метода секущих заполнить таблицу 4.

5.5 Для метода простой итерации заполнить таблицу 5.

6. Заполненные таблицы отобразить в отчете.

*Программная реализация:*

Для нелинейных уравнений:

1. Все численные методы (см. табл. 8) должны быть реализованы в виде отдельных подпрограмм/методов/классов.

2. Пользователь выбирает уравнение, корень/корни которого требуется вычислить (3-5 функций, в том числе и трансцендентные), из тех, которые предлагает

программа.

3. Предусмотреть ввод исходных данных (границы интервала/начальное приближение к корню и погрешность вычисления) из файла или с клавиатуры по выбору конечного пользователя.

4. Выполнить верификацию исходных данных. Необходимо анализировать наличие корня на введенном интервале. Если на интервале несколько корней или

они отсутствуют – выдавать соответствующее сообщение. Программа должна

реагировать на некорректные введенные данные.

5. Для методов, требующих начальное приближение к корню (методы Ньютона,

секущих, хорд с фиксированным концом), выбор начального приближения (а

или b) вычислять в программе.

6. Для метода простой итерации проверять достаточное условие сходимости метода на введенном интервале.

7. Предусмотреть вывод результатов (найденный корень уравнения, значение

функции в корне, число итераций) в файл или на экран по выбору конечного

пользователя.

8. Организовать вывод графика функции, график должен полностью отображать

весь исследуемый интервал (с запасом).

Для систем нелинейных уравнений:

1. Пользователь выбирает предлагаемые программой системы двух нелинейных

уравнений (2-3 системы).

2. Организовать вывод графика функций.

3. Начальные приближения ввести с клавиатуры.

4. Для метода простой итерации проверить достаточное условие сходимости.

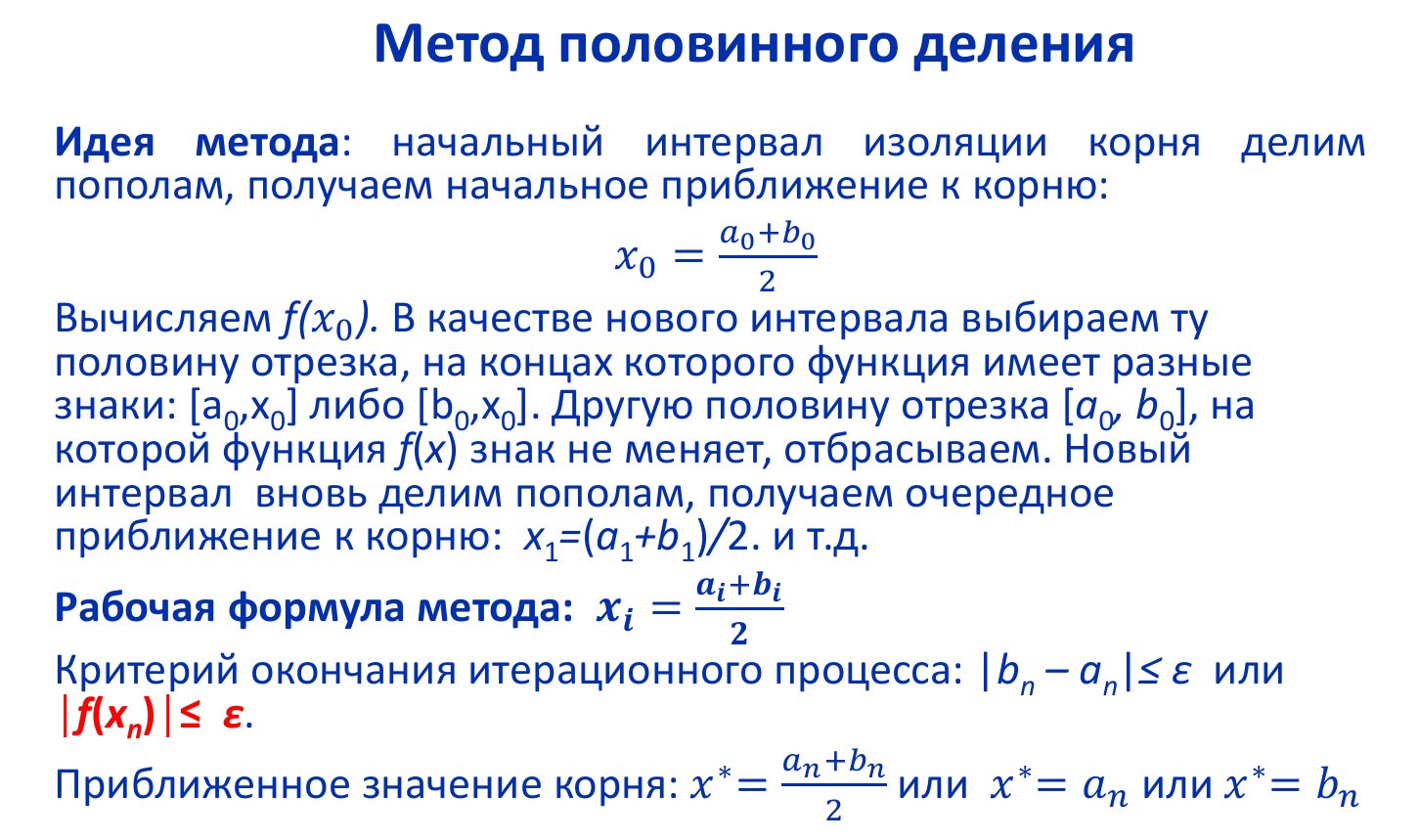
5. Организовать вывод вектора неизвестных: 𝑥1, 𝑥2.

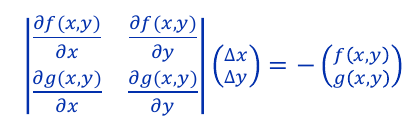
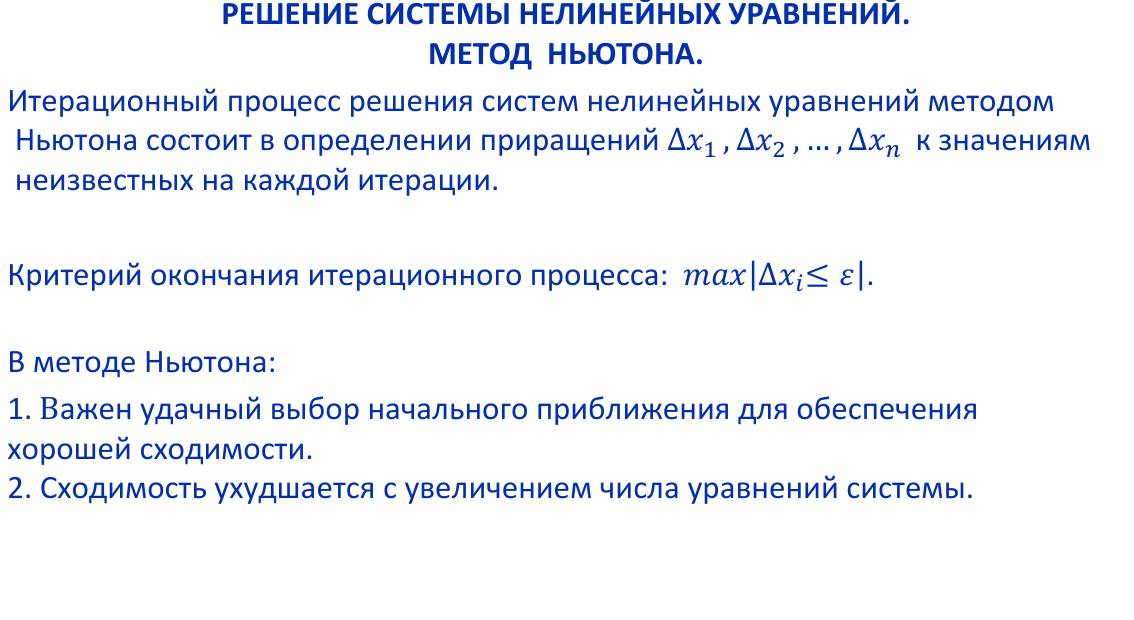
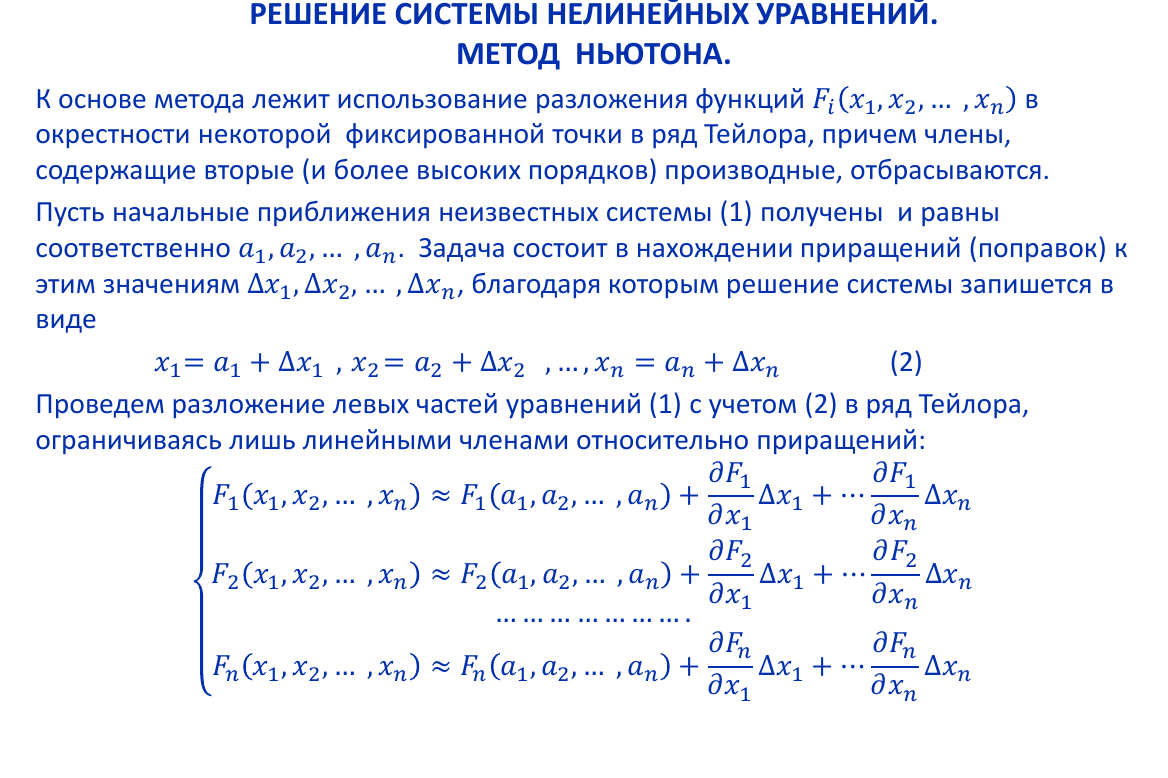
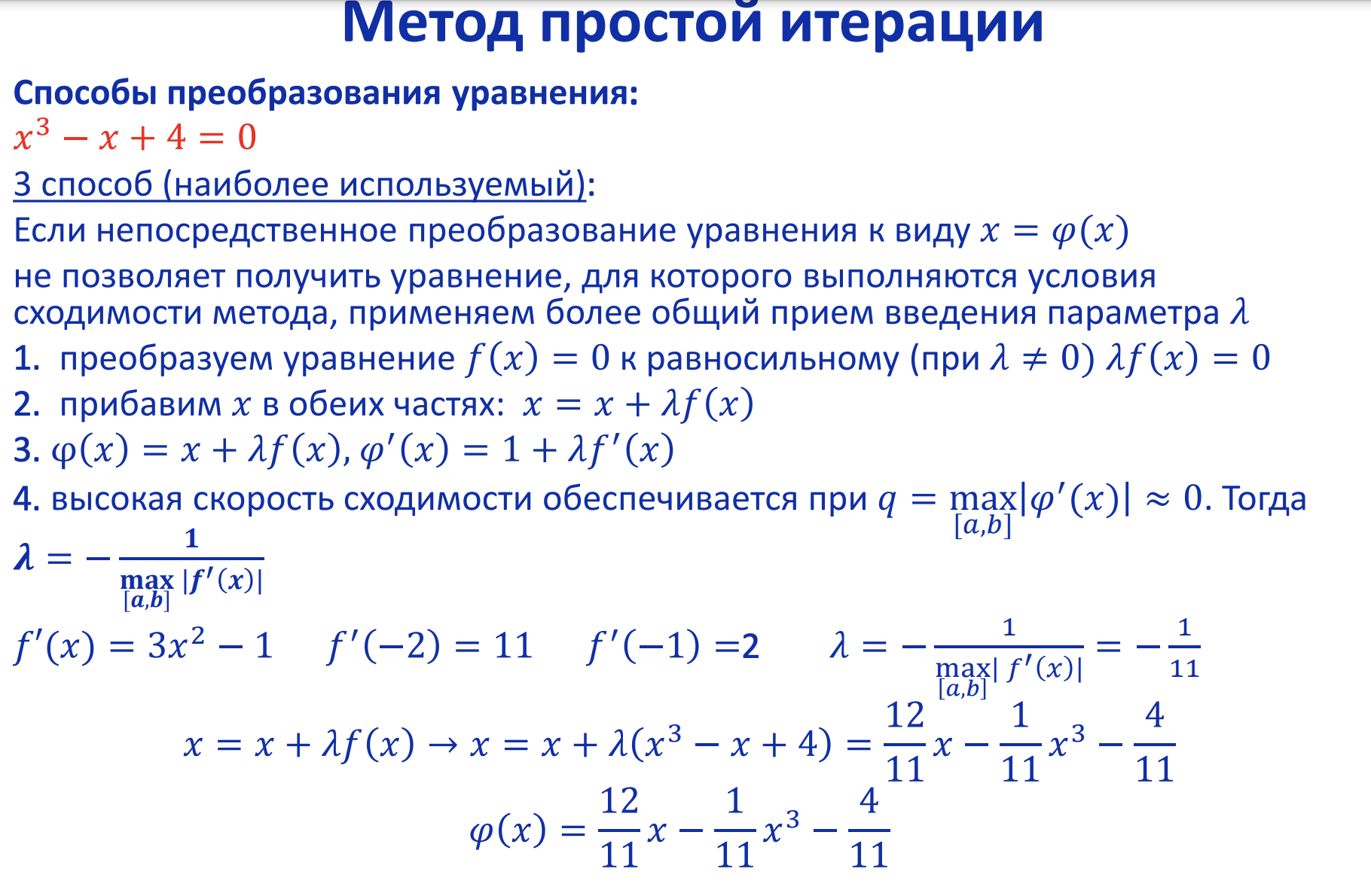
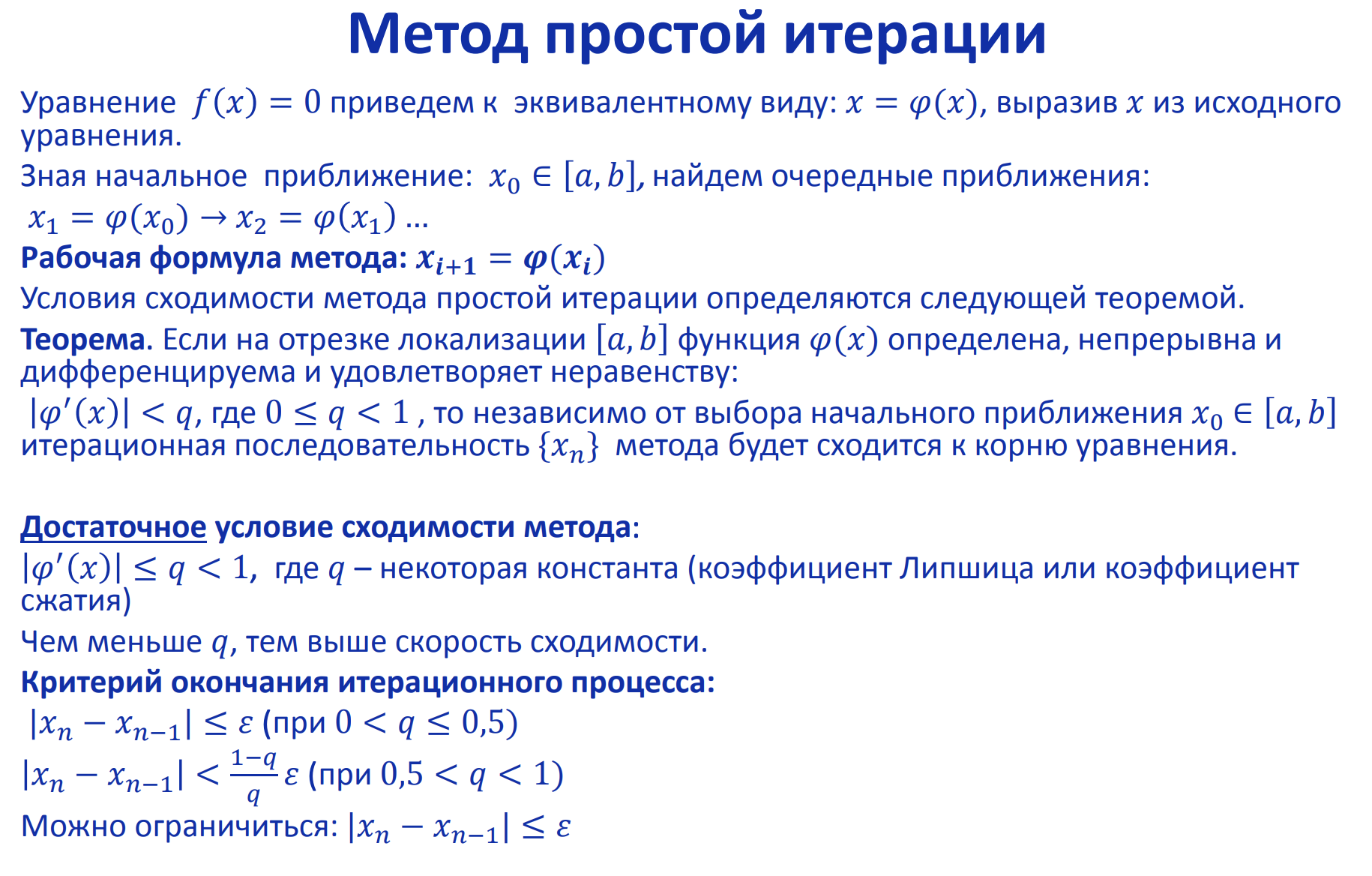
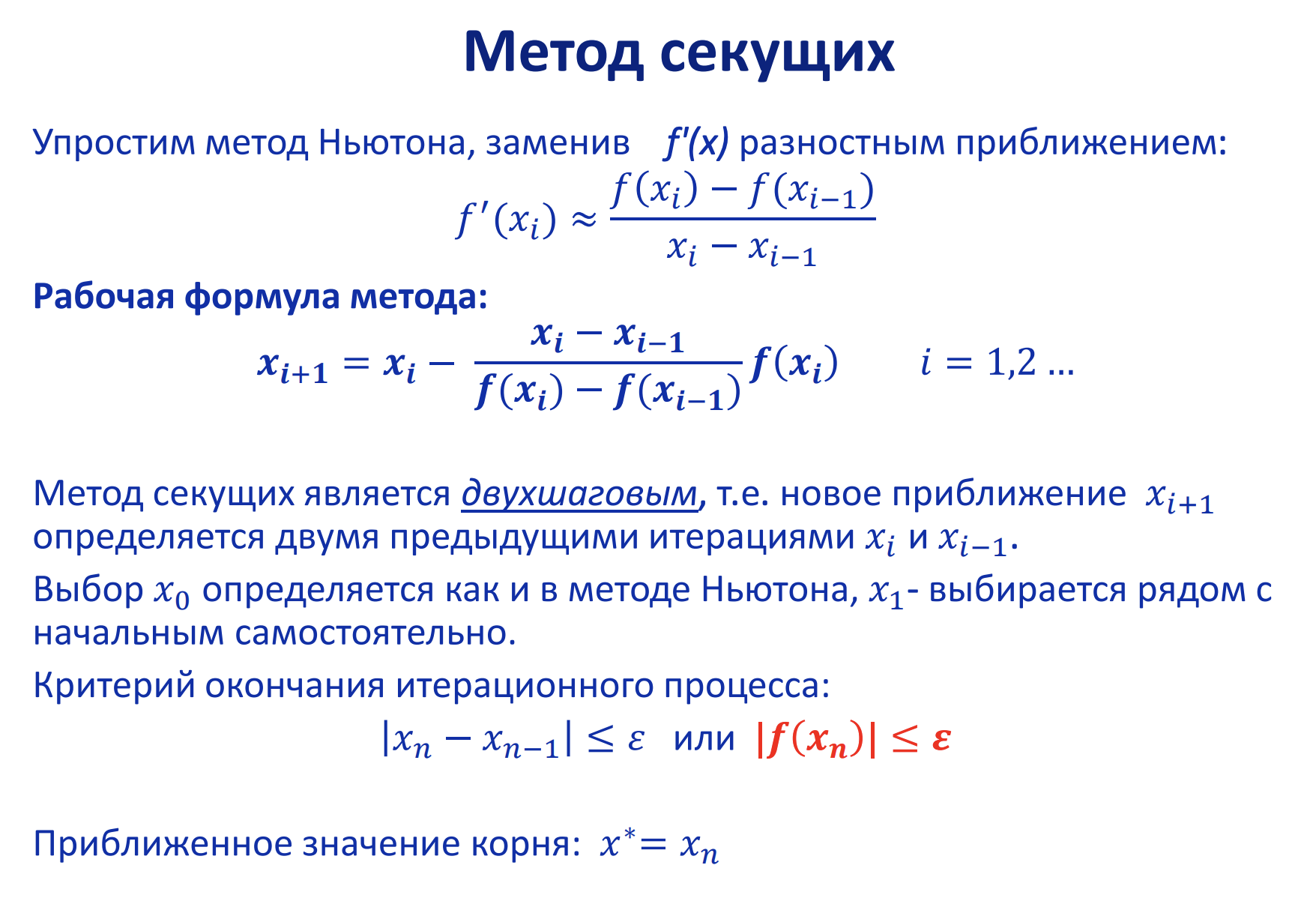
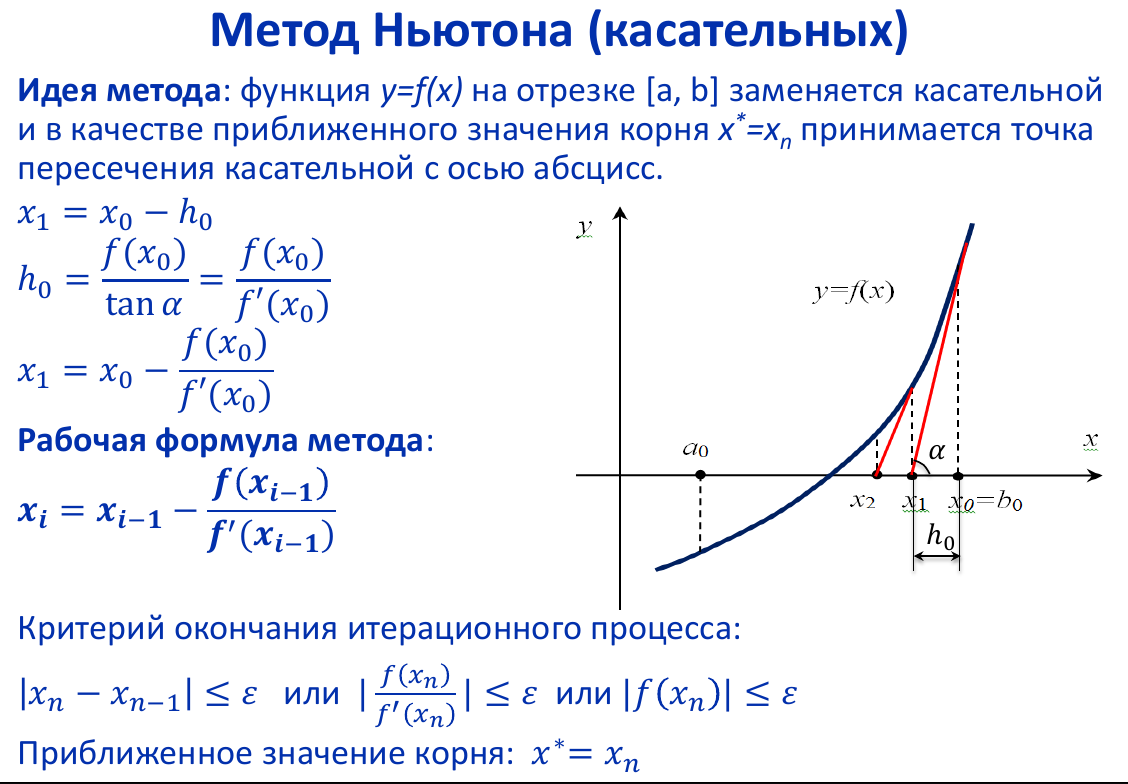
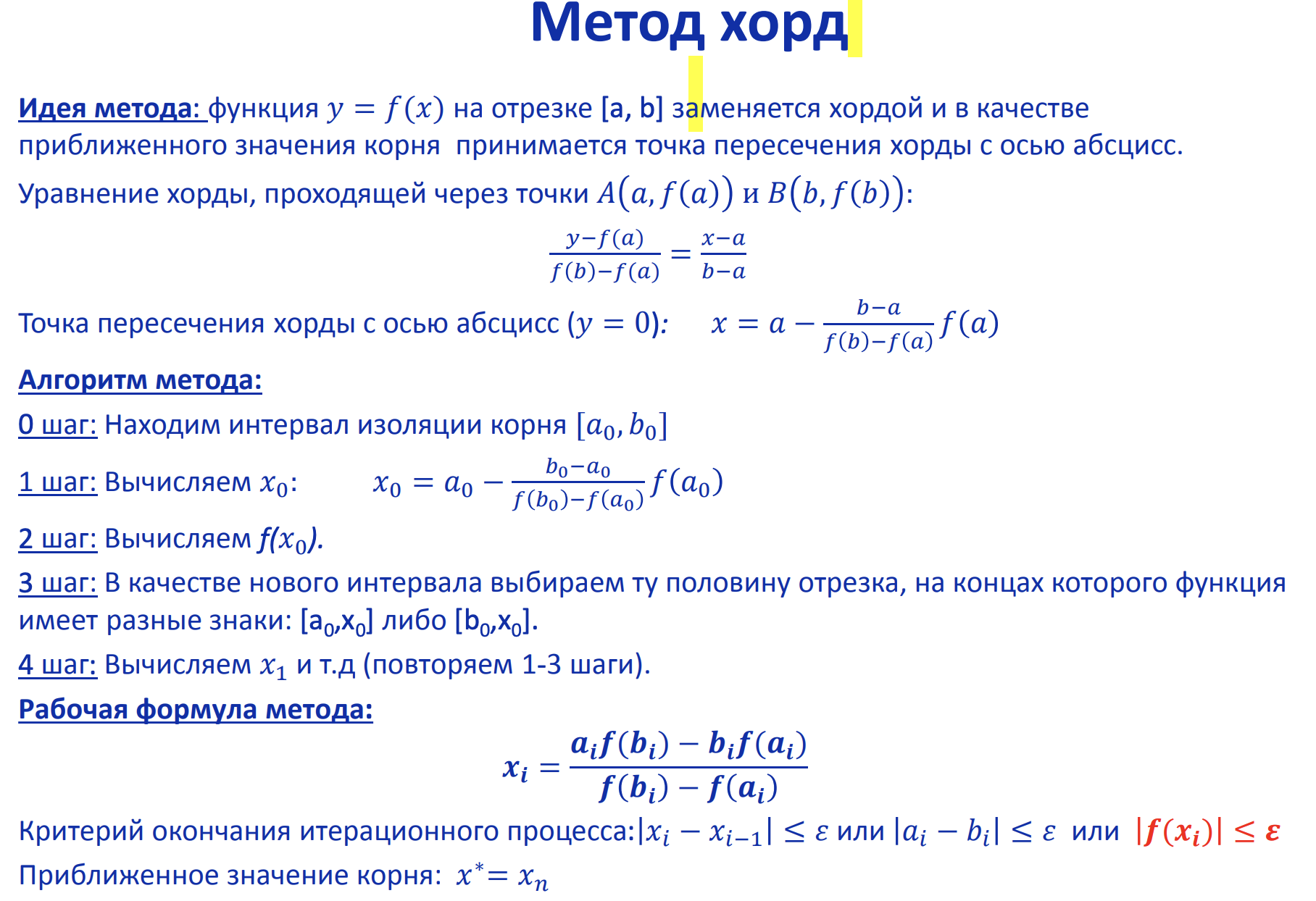
6. Организовать вывод количества итераций, за которое было найдено решение.

7. Организовать вывод вектора погрешностей: |𝑥𝑖^(𝑘) − 𝑥𝑖^(𝑘−1)|

8. Проверить правильность решения системы нелинейных уравнений.

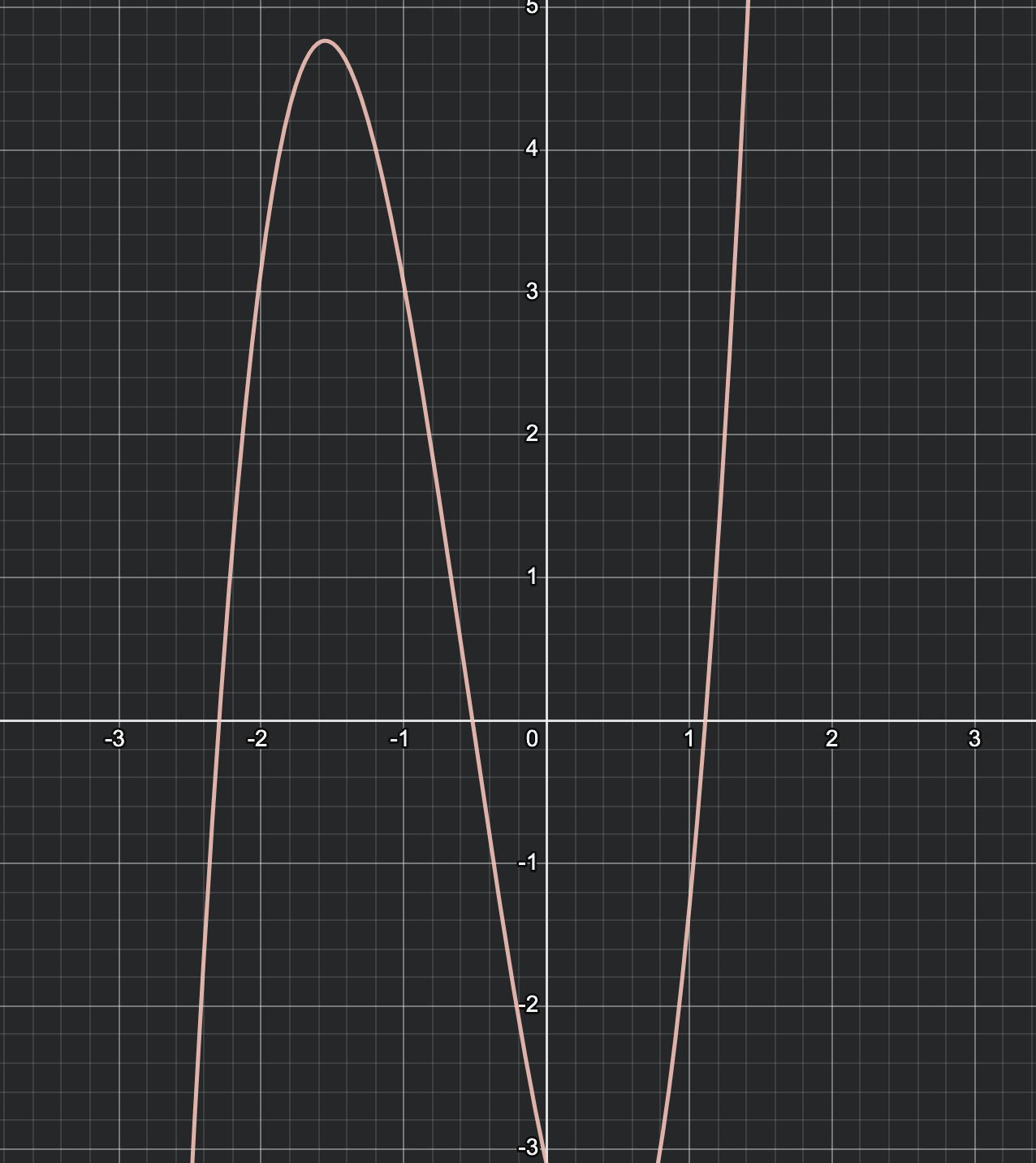
**Используемые формулы/методы:**

****

****

**Вычислительная реализация:**

Функция: 2,335𝑥^3 + 3,98𝑥^2 − 4,52𝑥 −3,11



Крайний правый корень [1, 2]

Метод половинного деления

n=8 - число итераций для точности 10^-2

Уточнение корня уравнения методом половинного деления

| № шага | a | b | x | f(a) | f(b) | f(x) | |a-b| |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 2 | 1.5 | -1.315 | 22.45 | 6.946 | 1 |
| 2 | 1 | 1.5 | 1.25 | -1.315 | 6.946 | 2.019 | 0.5 |
| 3 | 1 | 1.25 | 1.125 | -1.315 | 2.019 | 0.167 | 0.25 |
| 4 | 1 | 1.125 | 1.0625 | -1.315 | 0.167 | -0.619 | 0.125 |
| 5 | 1.0625 | 1.125 | 1.09375 | -0.619 | 0.167 | -0.237 | 0.0625 |
| 6 | 1.09375 | 1.125 | 1.1094 | -0.237 | 0.167 | -0.038 | 0.03125 |
| 7 | 1.1094 | 1.125 | 1.1172 | -0.038 | 0.167 | 0.064 | 0.0156 |
| 8 | 1.1094 | 1.1172 | **1.1133** | -0.038 | 0.064 | **0.013** | 0.0078 |

Правый корень уравнения: 1.1133 (точность 0.01)

Крайний левый корень [-3, -2]  
Метод Ньютона

Уточнение корня уравнения методом Ньютона

| № итерации | *xk* | *f*(*xk* ) | *f '*(*xk*) | *xk*+1 | │*xk* − *xk*+1│ |
| --- | --- | --- | --- | --- | --- |
| 1 | -3 | -16.775 | 34.645 | -2.5158 | 0.4842 |
| 2 | -2.5158 | -3.7287 | 19.7907 | -2.3274 | 0.1884 |
| 3 | -2.3274 | -0.4687 | 14.8984 | -2.2959 | 0.03146 |
| 4 | **-2.2959** | -0.01212 | 14.13 | -2.295042 | **0.000858** |

Левый корень уравнения: -2.2959 (точность 0.01)

Центральный корень [-1, 0]

Метод простой итерации

Уточнение корня уравнения методом простой итерации

| № итерации | *xk* | *xk*+1 | *f*(*xk+1* ) | │*xk* − *xk*+1│ |
| --- | --- | --- | --- | --- |
| 1 | -1 | -0.3242 | -1.3056 | 0.6758 |
| 2 | -0.3242 | -0.613 | 0.6188 | 0.2888 |
| 3 | -0.613 | -0.4762 | -0.3074 | 0.1368 |
| 4 | -0.4762 | -0.5442 | 0.1519 | 0.068 |
| 5 | -0.5442 | -0.5106 | -0.07554 | 0.0336 |
| 6 | -0.5106 | **-0.5273** | 0.03749 | 0.0167 |

Центральный корень уравнения: -0.5273 (точность 0.01)

2.335x^3+3.98x^2 - 4.52x -3.11

**Программная реализация:**

Решение нелинейных уравнений:

Метод хорд

Метод секущих

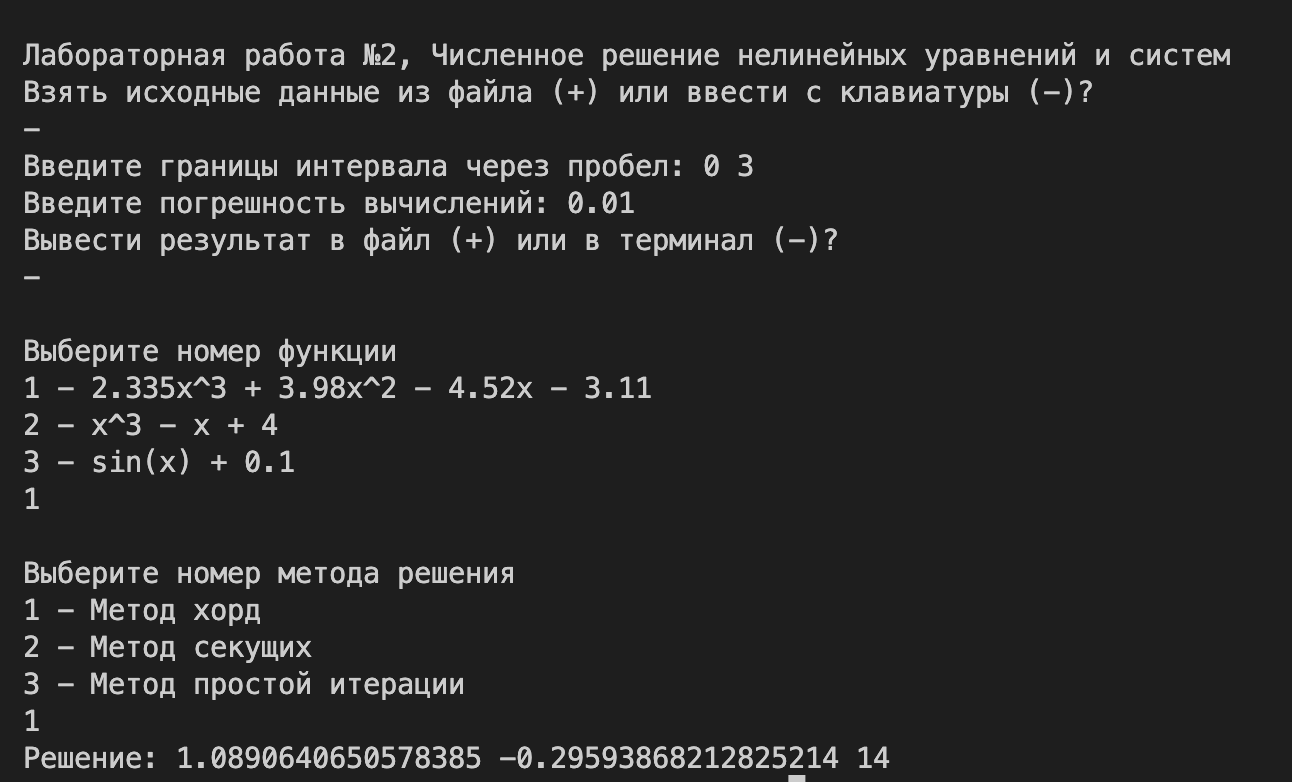
Метод простой итерации

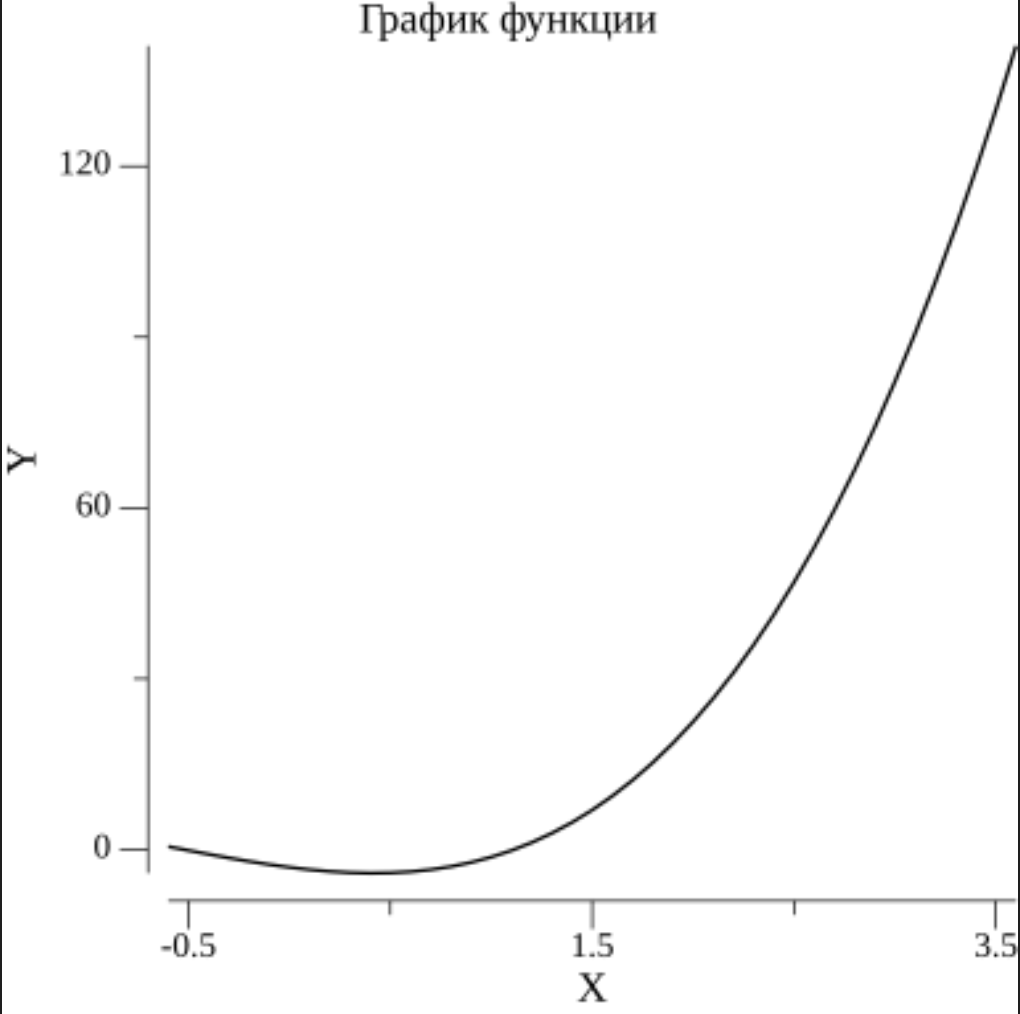
Решение систем нелинейных уравнений:

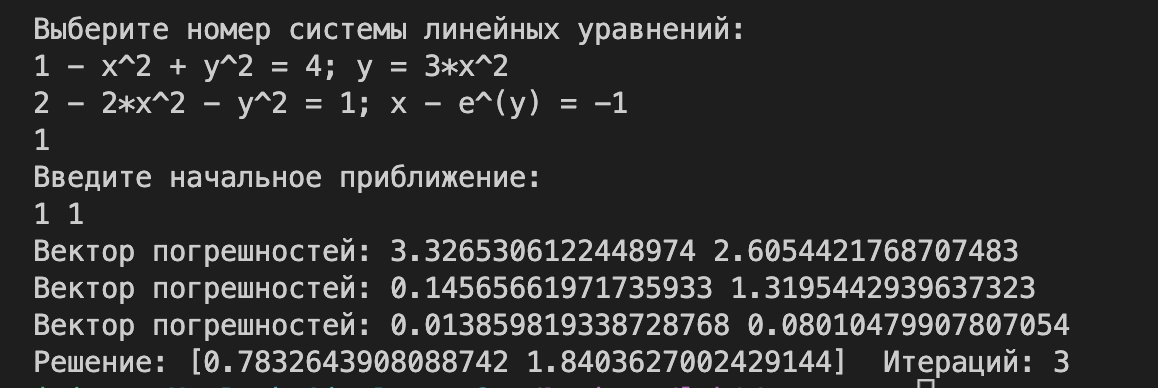
Метод Ньютона

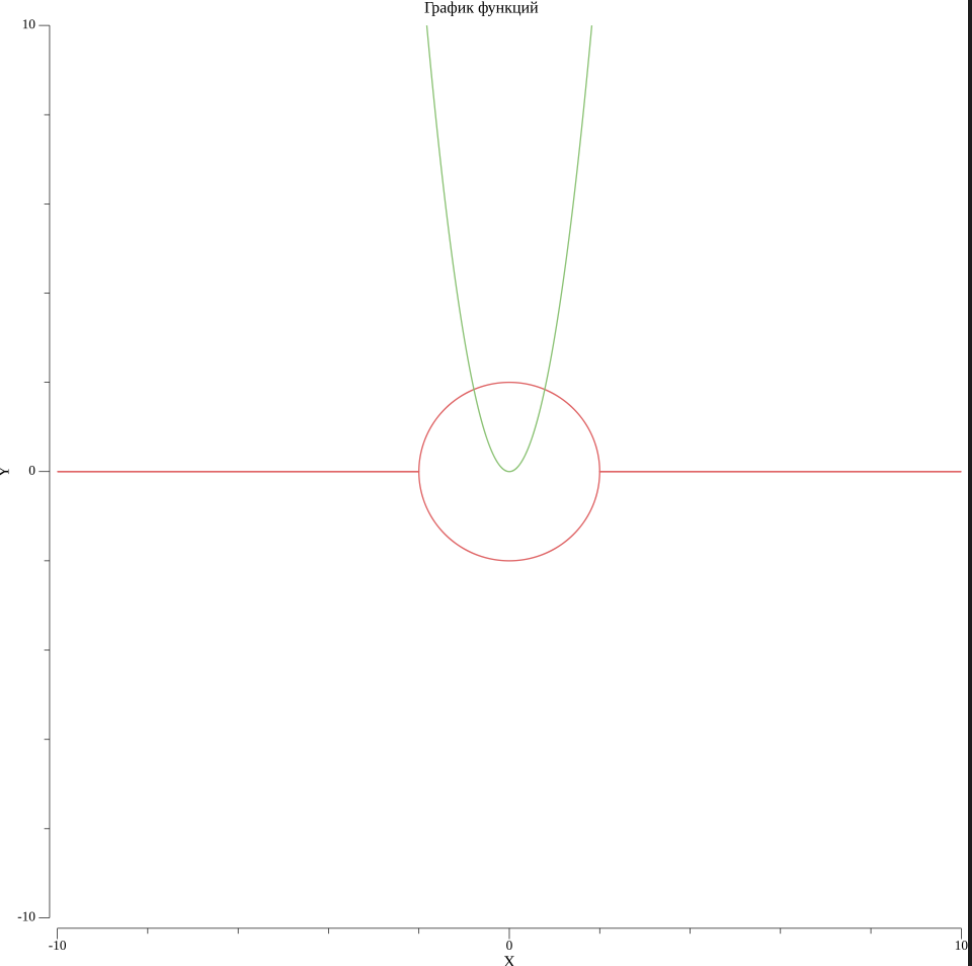
Код программы: <https://github.com/Ja1rman/Computational-Mathematics/tree/main/lab2>

**Примеры работы программы:**

****

****

****



**Вывод:**

Во время выполнения лабораторной работы я познакомился с несколькими методами, позволяющими решать нелинейные уравнения и системы нелинейных уравнений. Все методы довольно легко программируются и дают высокую точность и быструю сходимость при удачном выборе начального приближения.