Отчёт по лабораторной работе №3

«Численные методы решения нелинейных уравнений»

Выполнили:

Адаменко С. С.

Гневнов А. Е.

Суворов Р.М.

Отчёт Адаменко С.С.

Тема: «Численные методы решения нелинейных уравнений»

Используемое оборудование: ПК, языки программирования: Python, HTML, CSS, SQL, JavaScript; используемые сторонние библиотеки: Django, Matpotlib, Numpy, SQLlite3; среда разработки Visual Studio Code.

Постановка задачи: Изучить численные методы деления отрезка пополам, касательных, хорд

Математическая модель:

1. Метод деления отрезка пополам

Метод можно использовать если: 1) $f(x)$ – непрерывна на $[a; b]$; 2) $f(x)$ принимает на концах отрезка разные знаки, то есть $f(a)f(b) < 0$

Алгоритм:

1) За $x0$ выбираем середину отрезка $[a; b]$, то есть $x0 = a+b /2$ Будем искать корень на одном из отрезков: $[a; x0]$ и $[x0; b]$. На концах этих отрезков функция $f(x)$ принимает разные знаки ($f(a)f(x0) < 0$ и $f(x0)f(b) < 0$).

2) Вычислить $f(a)$ и $f(x0)$.

3) Найти $f(a)f(x0)$ и проверить выполнение неравенства $f(a)f(x0) < 0$. Если неравенство верно, то корень принадлежит отрезку $[a; x0]$ и далее пункт 5.

4) Вычислить $f(x0)$ и $f(b)$. Найти $f(x0)f(b)$. Если $f(x0)f(b) < 0$, то корень принадлежит отрезку $[x0; b]$ и далее пункт 7.

5) Продолжаем сужать отрезок. Делим его ещё раз пополам и находим следующий предполагаемый корень $x1 = a + x0\ 2$

6) Проверяем выполнение неравенства $f(x1)f(x0) < 0$. Если верно, то корень $\epsilon[x0; x1]$

7) Опять сужаем отрезок. Делим его и находим ещё предполагаемый корень $x1 = b + x0/2$

8) Деление отрезка продолжаем до тех пор, пока его длина не будет превосходить заданную точность h.

2. Метод касательных

1. **Итерационный процесс:**
   - Формула итерации: $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$
   - Где x_k - текущее приближение, $f(x_k)$ - значение функции в точке x_k, а $f'(x_k)$ - производная функции в точке x_k.

Процесс останавливается, когда значение функции достаточно близко к нулю или достигнута заданная точность

3. Метод хорд

Теория метода хорд:

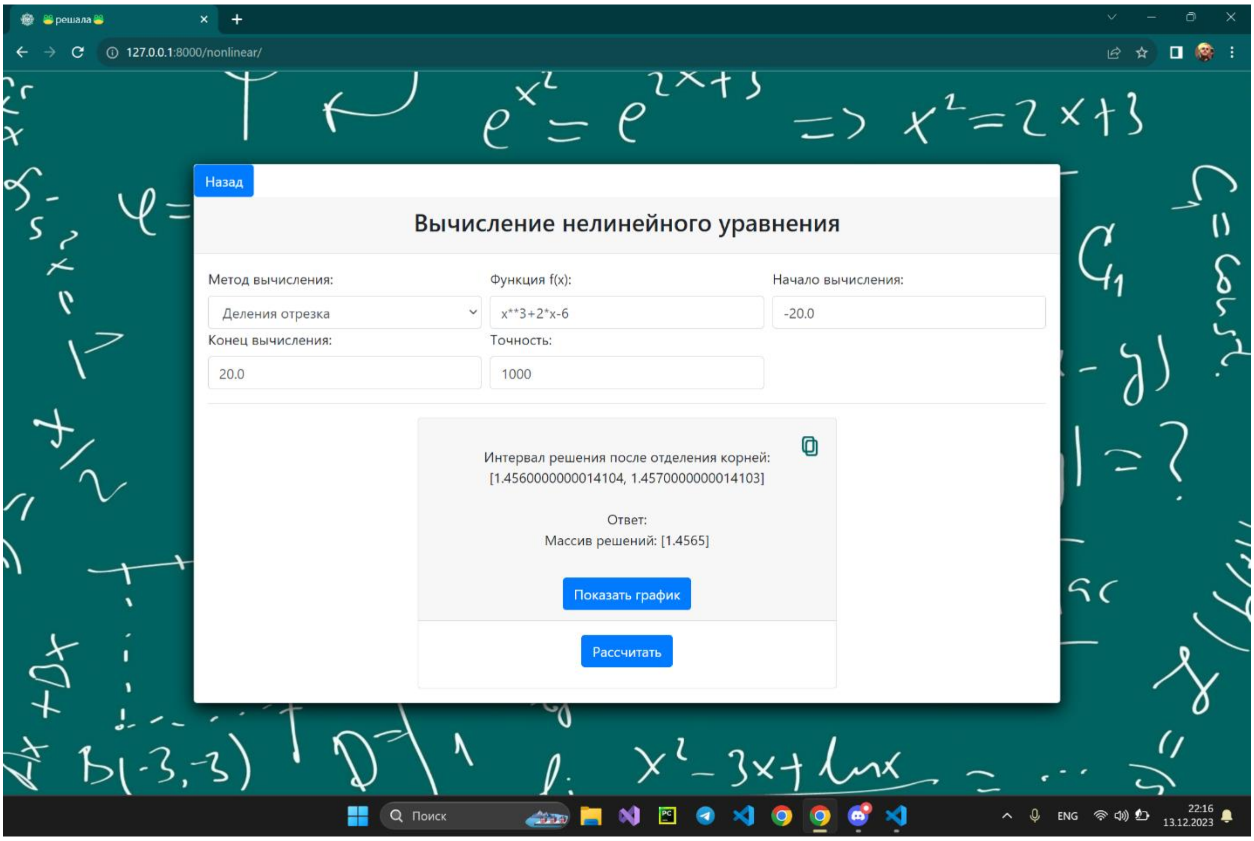
1. **Итерационный процесс:**
   - $x_{k+1} = x_k - \frac{f(x_k)\cdot(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$
   - Где x_k и x_{k-1} - текущее и предыдущее приближения, соответственно.
2. **Условие остановки:**
   - Остановить процесс, когда значение функции достаточно близко к нулю или достигнута заданная точность.

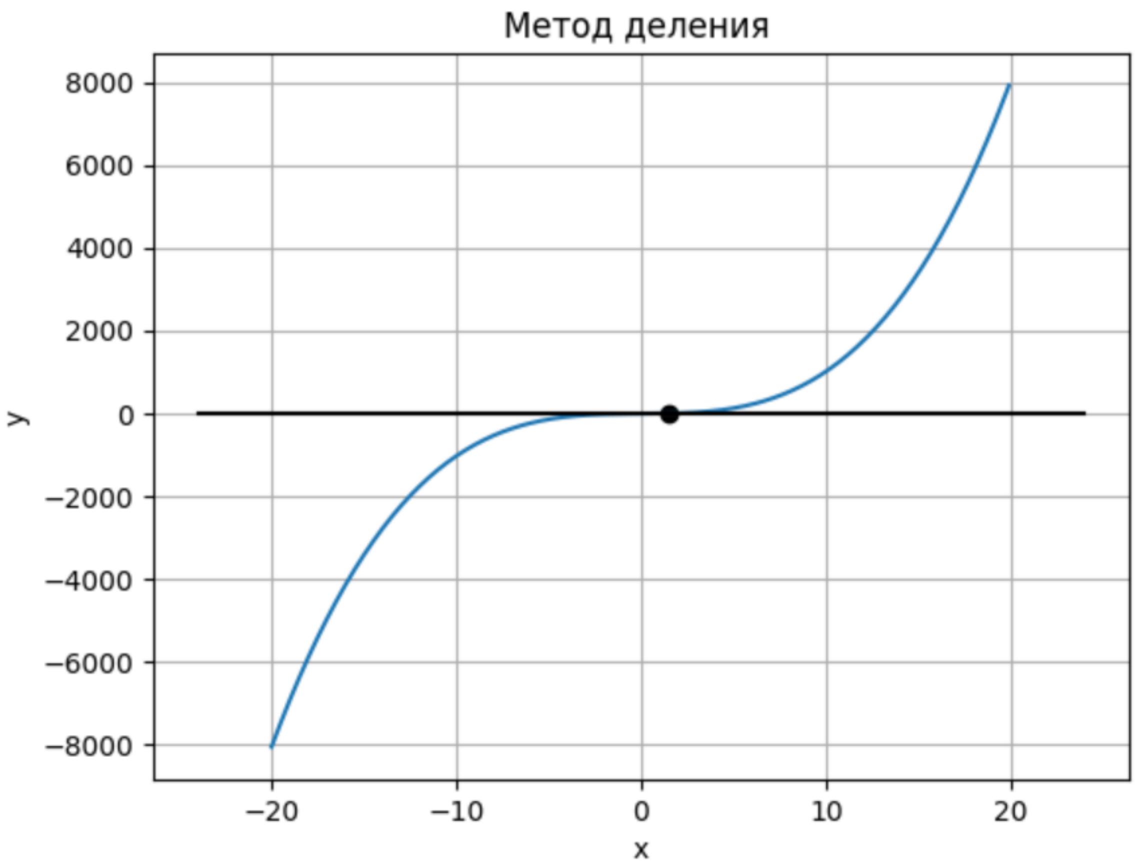
Код программы:

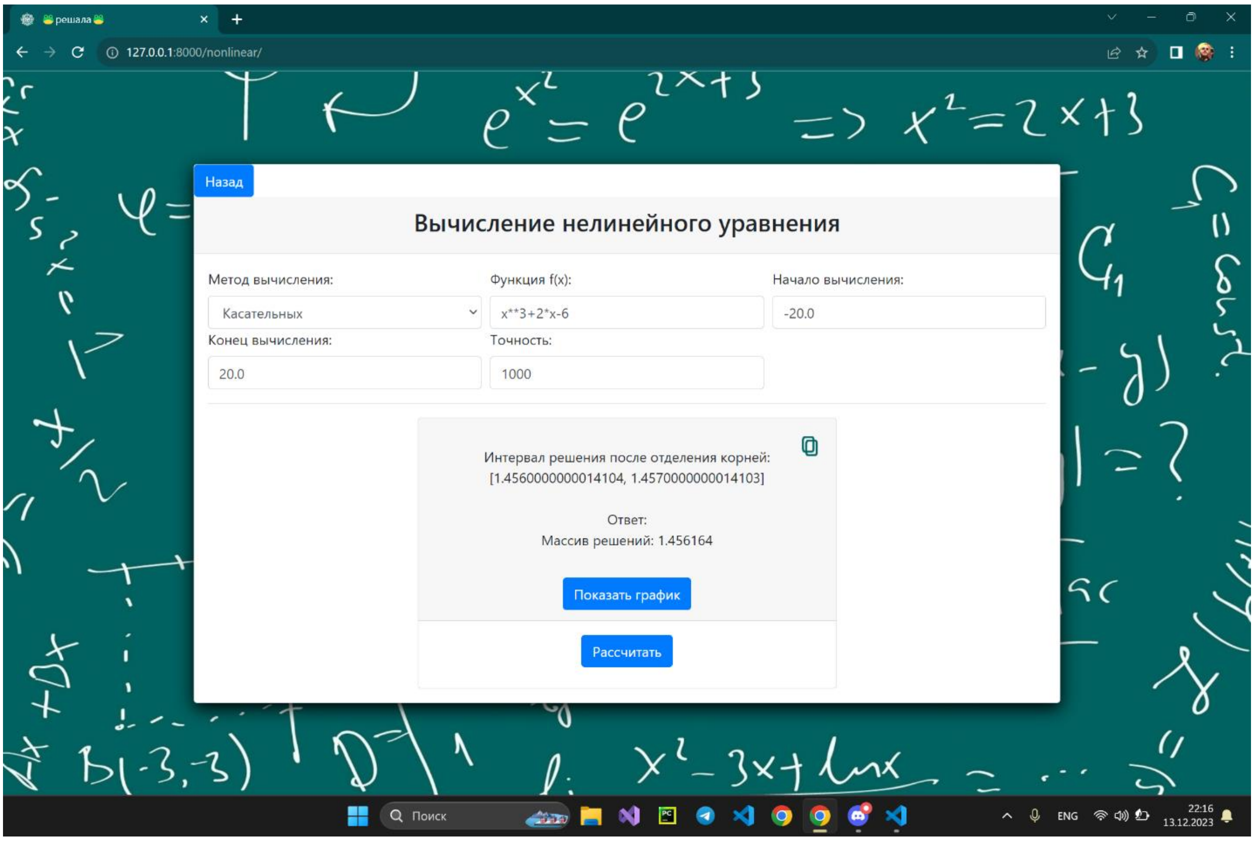
https://github.com/webbsalad/Computational_Mathematics_LW1

Для открытия веб-приложения, необходимо скачать файлы, в терминале зайди в директорию, выполнить команду «python manage.py run server».

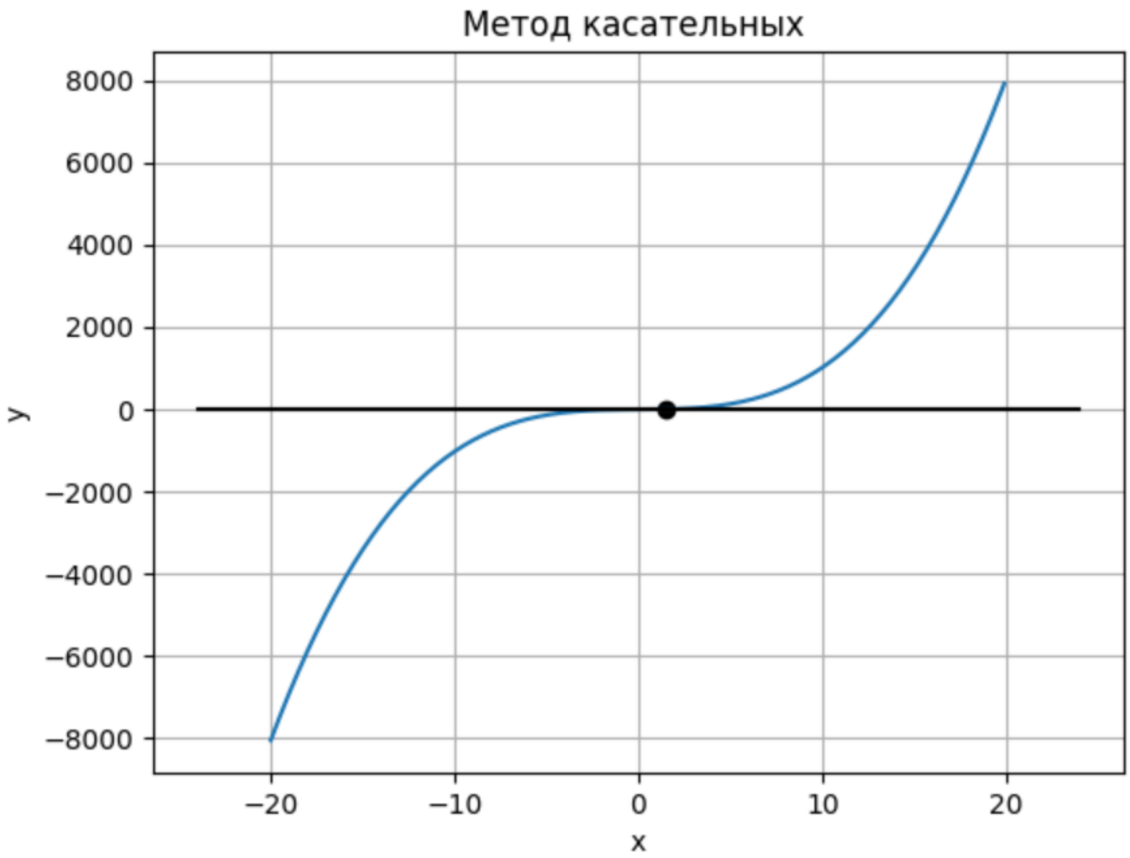
Результат выполнения работы:


127.0.0.1:8000/nonlinear/
Назад
Вычисление нелинейного уравнения
Метод вычисления:
Деления отрезка
Функция f(x):
x**3+2*x-6
Начало вычисления:
-20.0
Конец вычисления:
20.0
Точность:
1000
Интервал решения после отделения корней:
[1.4560000000014104, 1.4570000000014103]
Ответ:
Массив решений: [1.4565]
Показать график
Рассчитать

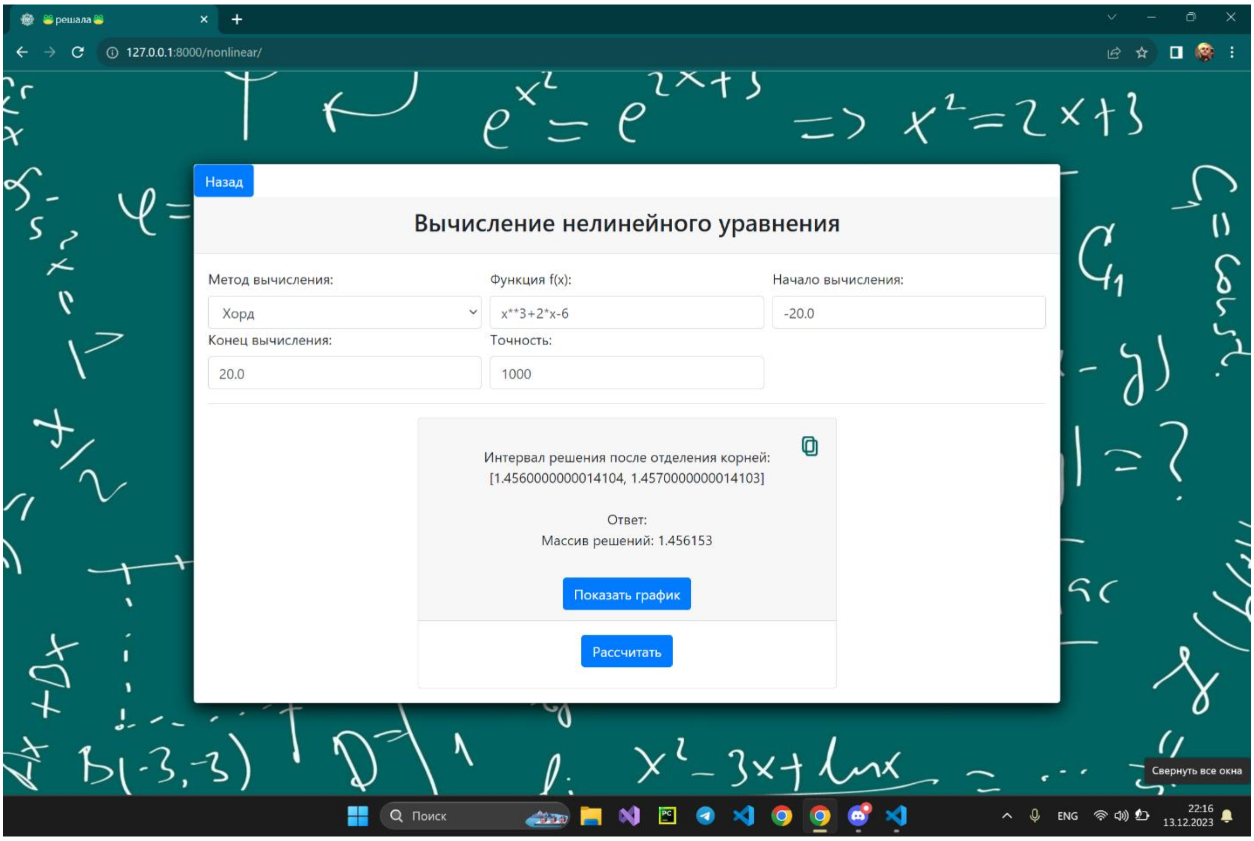

Метод деления
8000
6000
4000
2000
0
−2000
−4000
−6000
−8000
y
−20
−10
0
10
20
x

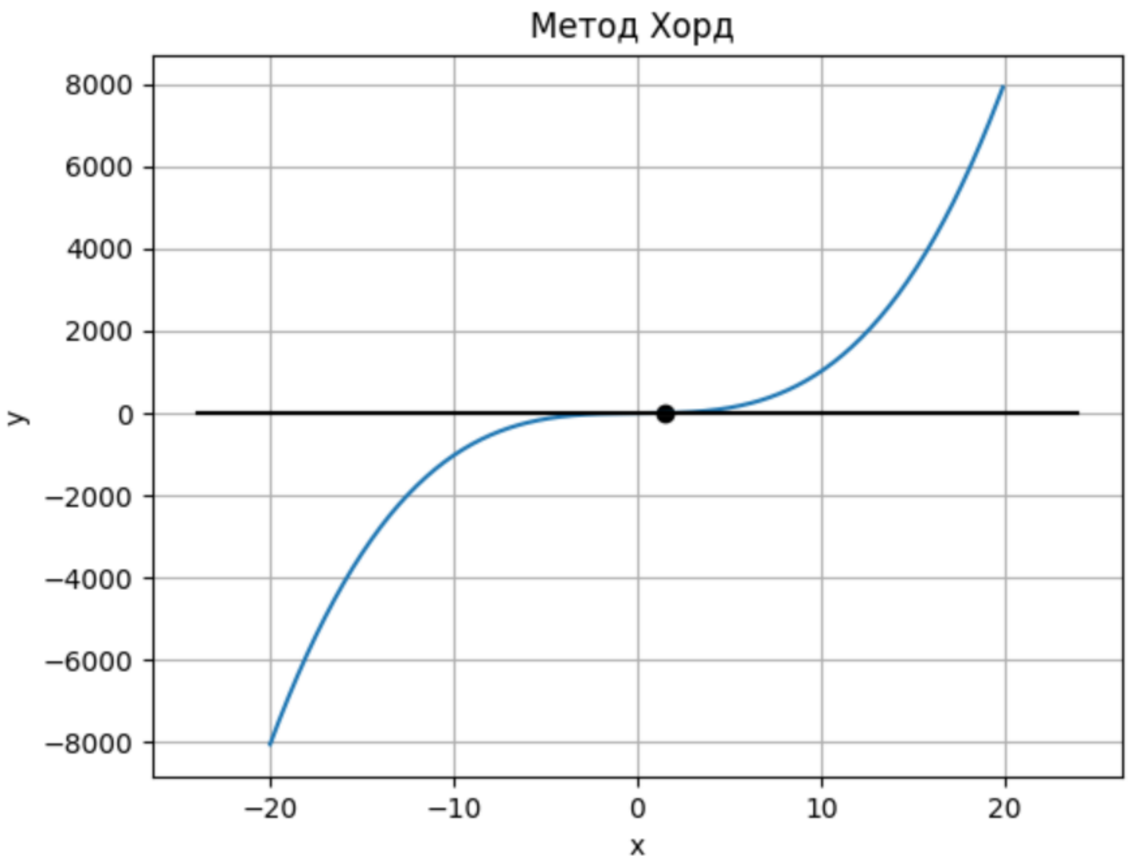


решала
127.0.0.1:8000/nonlinear/
Назад
Вычисление нелинейного уравнения
Метод вычисления:
Касательных
Функция f(x):
x**3+2*x-6
Начало вычисления:
-20.0
Конец вычисления:
20.0
Точность:
1000
Интервал решения после отделения корней:
[1.4560000000014104, 1.4570000000014103]
Ответ:
Массив решений: 1.456164
Показать график
Рассчитать



Метод касательных
8000
6000
4000
2000
0
−2000
−4000
−6000
−8000
y
−20
−10
0
10
20
x





Сравнительный анализ полученных результатов:

Возьмём контрольный пример $x^3 + 2x - 6 = 0$ на отрезке от -20 до +20 с точностью 10^{-6} с ответом x ≈ 1.456162, мы выяснили, что метод касательных является самым точным.

Вывод:

Нам удалось, верно, реализовать численные методы решения нелинейных уравнений разными методами в нашем веб-приложении. В ходе работы получилось выяснить, что самым точным методом для решения нелинейных уравнений является метод касательных.

Отчёт Гневнов А.Е.

Тема: «Численные методы решения нелинейных уравнений»

Используемое оборудование: ПК, языки программирования: Python, HTML, CSS, SQL, JavaScript; используемые сторонние библиотеки: Django, Matpotlib, Numpy, SQLlite3; среда разработки Visual Studio Code.

Постановка задачи: Изучить численные методы деления отрезка пополам, касательных, хорд

Математическая модель:

1. Метод деления отрезка пополам

Метод можно использовать если: 1) $f(x)$ – непрерывна на $[a; b]$; 2) $f(x)$ принимает на концах отрезка разные знаки, то есть $f(a)f(b) < 0$

Алгоритм:

1) За $x0$ выбираем середину отрезка $[a; b]$, то есть $x0 = a+b /2$ Будем искать корень на одном из отрезков: $[a; x0]$ и $[x0; b]$. На концах этих отрезков функция $f(x)$ принимает разные знаки ($f(a)f(x0) < 0$ и $f(x0)f(b) < 0$).

2) Вычислить $f(a)$ и $f(x0)$.

3) Найти $f(a)f(x0)$ и проверить выполнение неравенства $f(a)f(x0) < 0$. Если неравенство верно, то корень принадлежит отрезку $[a; x0]$ и далее пункт 5.

4) Вычислить $f(x0)$ и $f(b)$. Найти $f(x0)f(b)$. Если $f(x0)f(b) < 0$, то корень принадлежит отрезку $[x0; b]$ и далее пункт 7.

5) Продолжаем сужать отрезок. Делим его ещё раз пополам и находим следующий предполагаемый корень $x1 = a + x0\ 2$

6) Проверяем выполнение неравенства $f(x1)f(x0) < 0$. Если верно, то корень $\epsilon[x0; x1]$

7) Опять сужаем отрезок. Делим его и находим ещё предполагаемый корень $x1 = b + x0/2$

8) Деление отрезка продолжаем до тех пор, пока его длина не будет превосходить заданную точность h.

2. Метод касательных

1. **Итерационный процесс:**
   - Формула итерации: $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$
   - Где x_k - текущее приближение, $f(x_k)$ - значение функции в точке x_k, а $f'(x_k)$ - производная функции в точке x_k.

Процесс останавливается, когда значение функции достаточно близко к нулю или достигнута заданная точность

3. Метод хорд

Теория метода хорд:

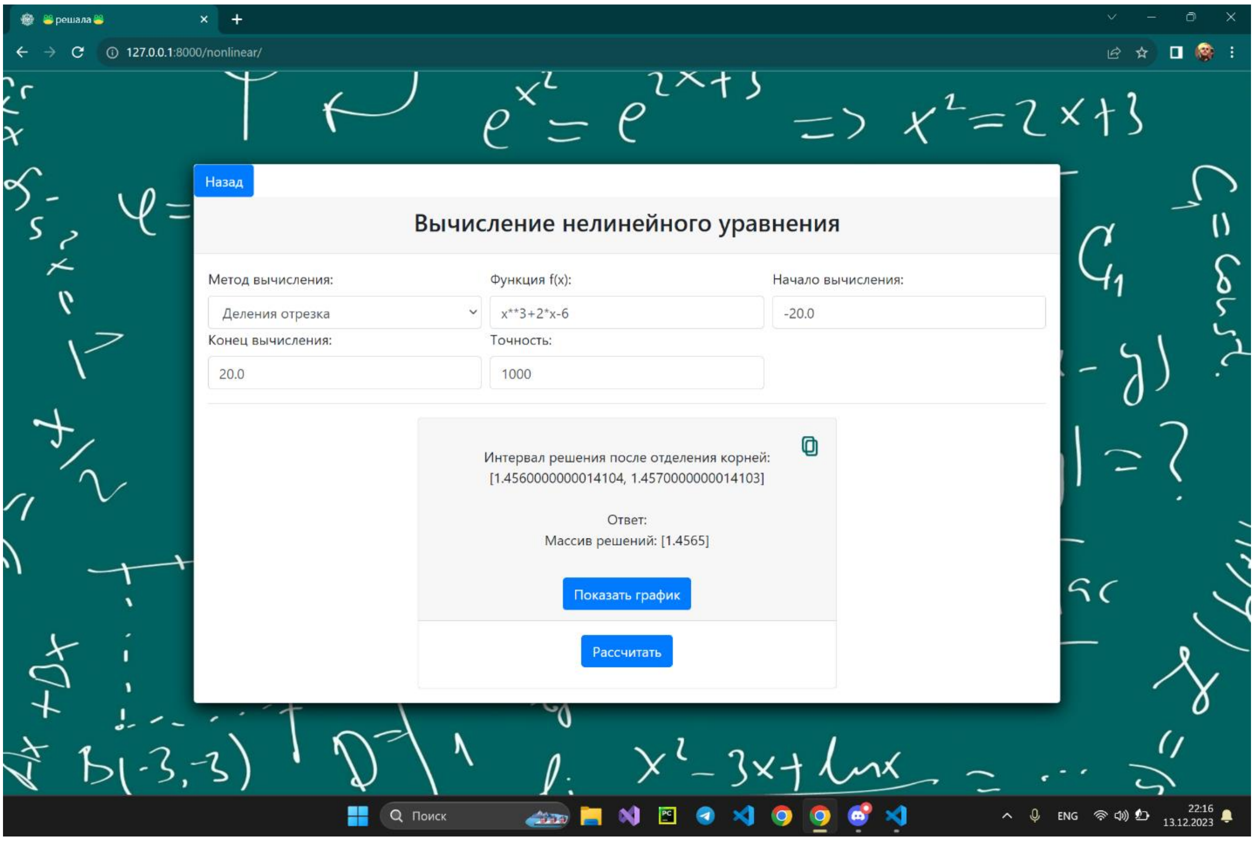
1. **Итерационный процесс:**
   - $x_{k+1} = x_k - \frac{f(x_k) \cdot (x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$
   - Где x_k и x_{k-1} - текущее и предыдущее приближения, соответственно.
2. **Условие остановки:**
   - Остановить процесс, когда значение функции достаточно близко к нулю или достигнута заданная точность.

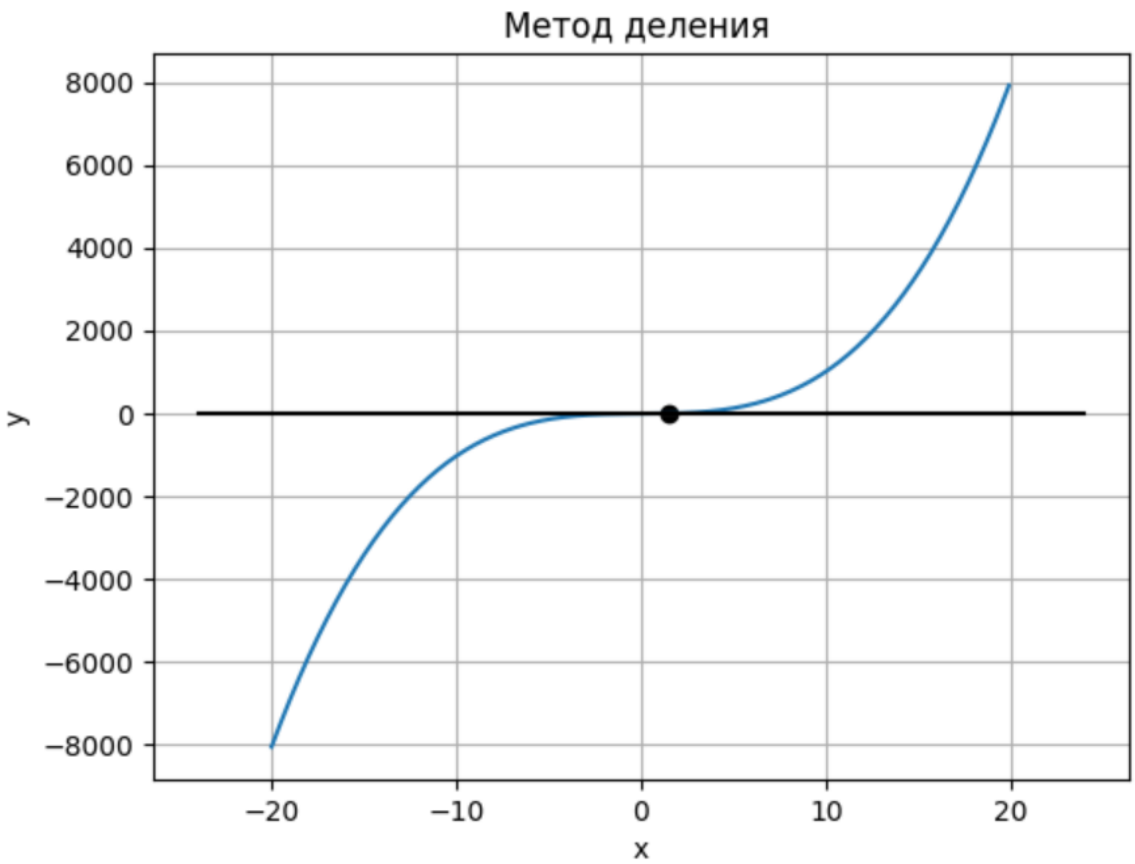
Код программы:

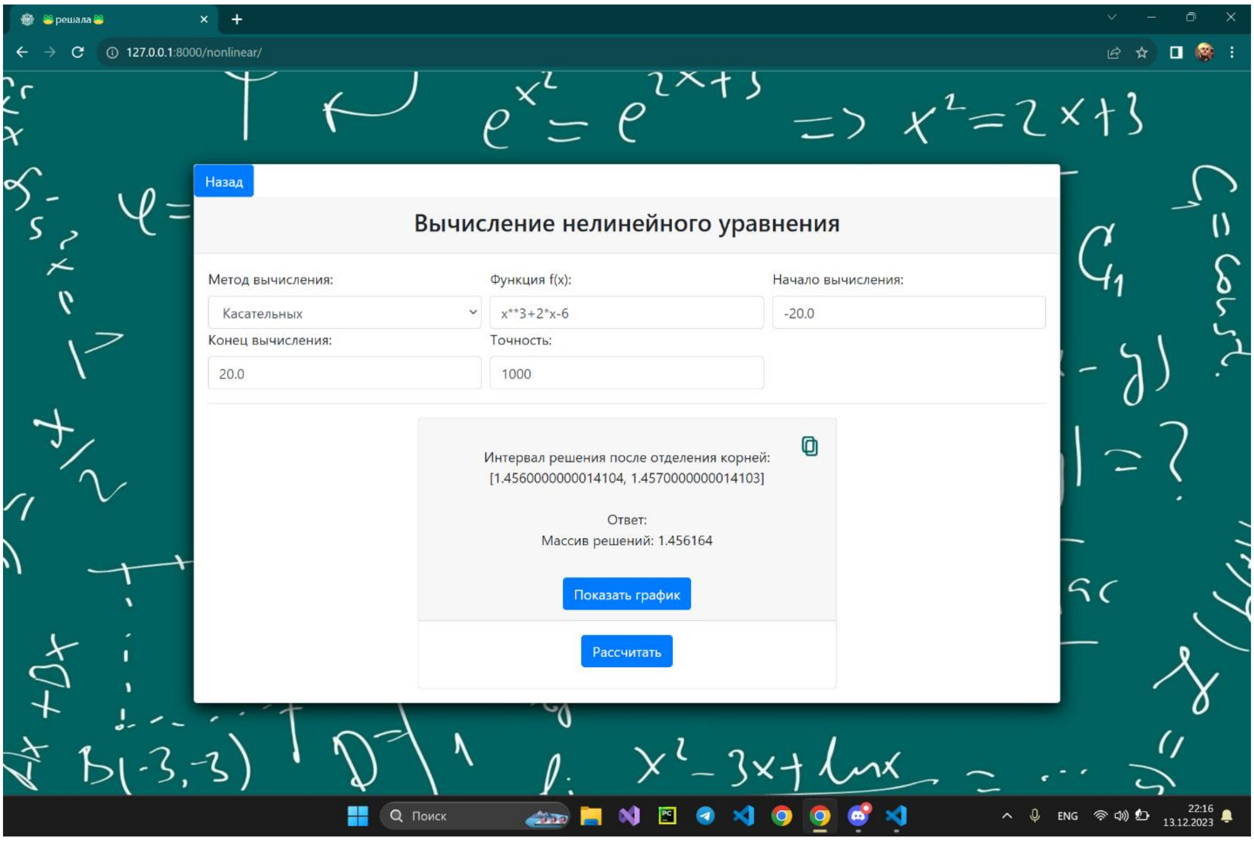
https://github.com/webbsalad/Computational_Mathematics_LW1

Для открытия веб-приложения, необходимо скачать файлы, в терминале зайди в директорию, выполнить команду «python manage.py run server».

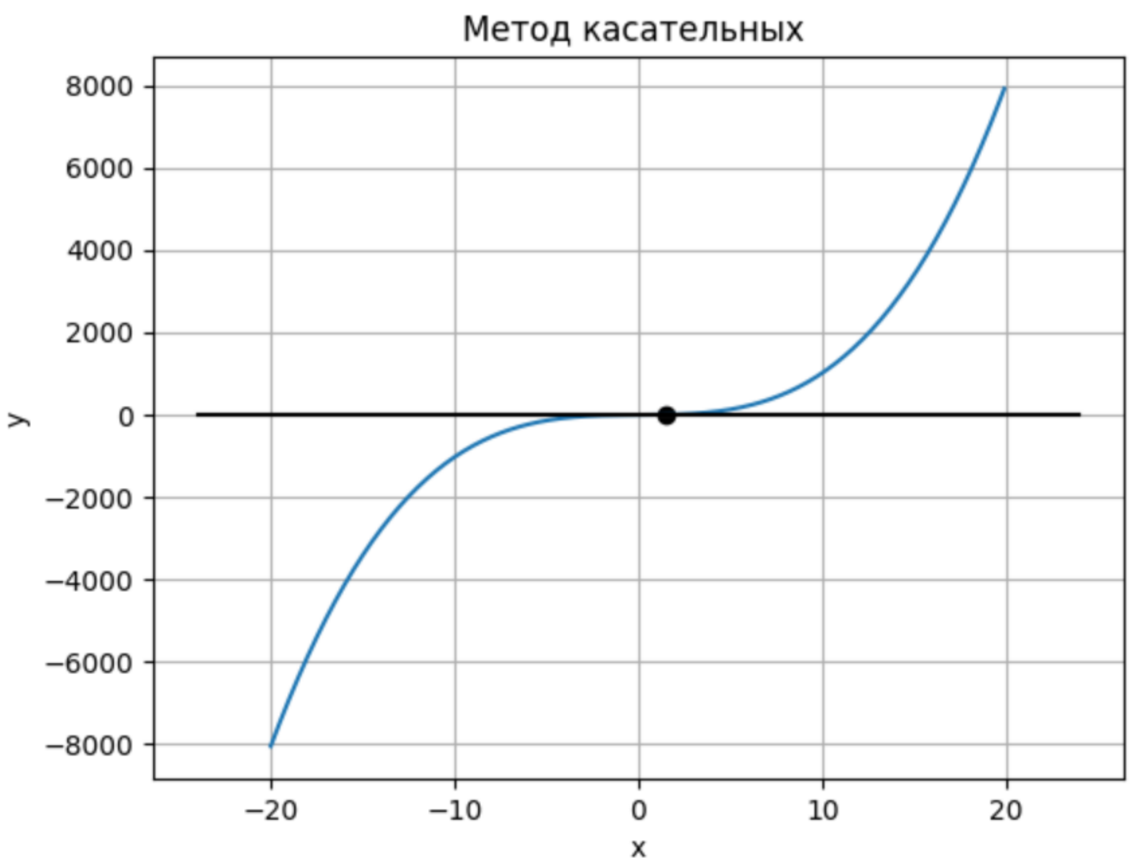
Результат выполнения работы:


решала
127.0.0.1:8000/nonlinear/
Назад
Вычисление нелинейного уравнения
Метод вычисления:
Деления отрезка
Функция f(x):
x**3+2*x-6
Начало вычисления:
-20.0
Конец вычисления:
20.0
Точность:
1000
Интервал решения после отделения корней:
[1.4560000000014104, 1.4570000000014103]
Ответ:
Массив решений: [1.4565]
Показать график
Рассчитать
Поиск
ENG
22:16
13.12.2023

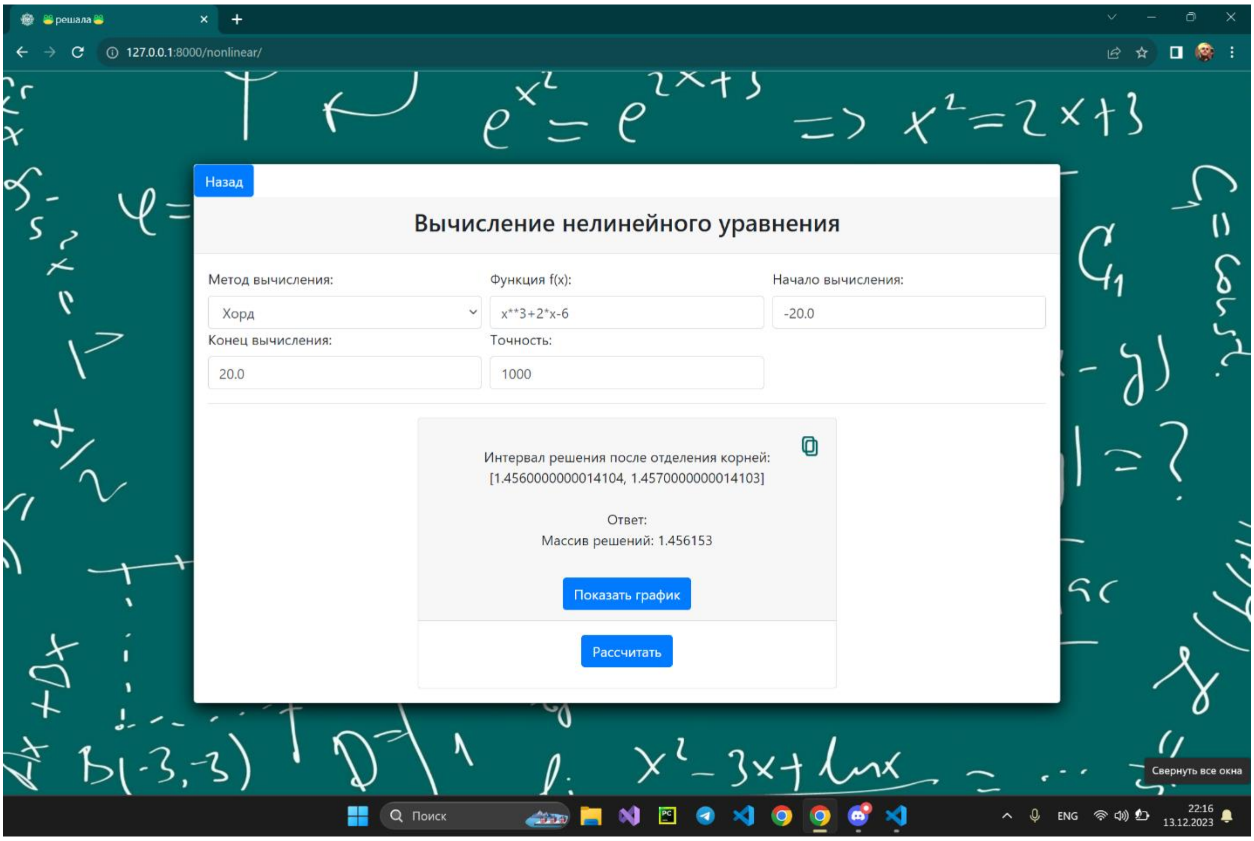

Метод деления
8000
6000
4000
2000
0
−2000
−4000
−6000
−8000
y
−20
−10
0
10
20
x

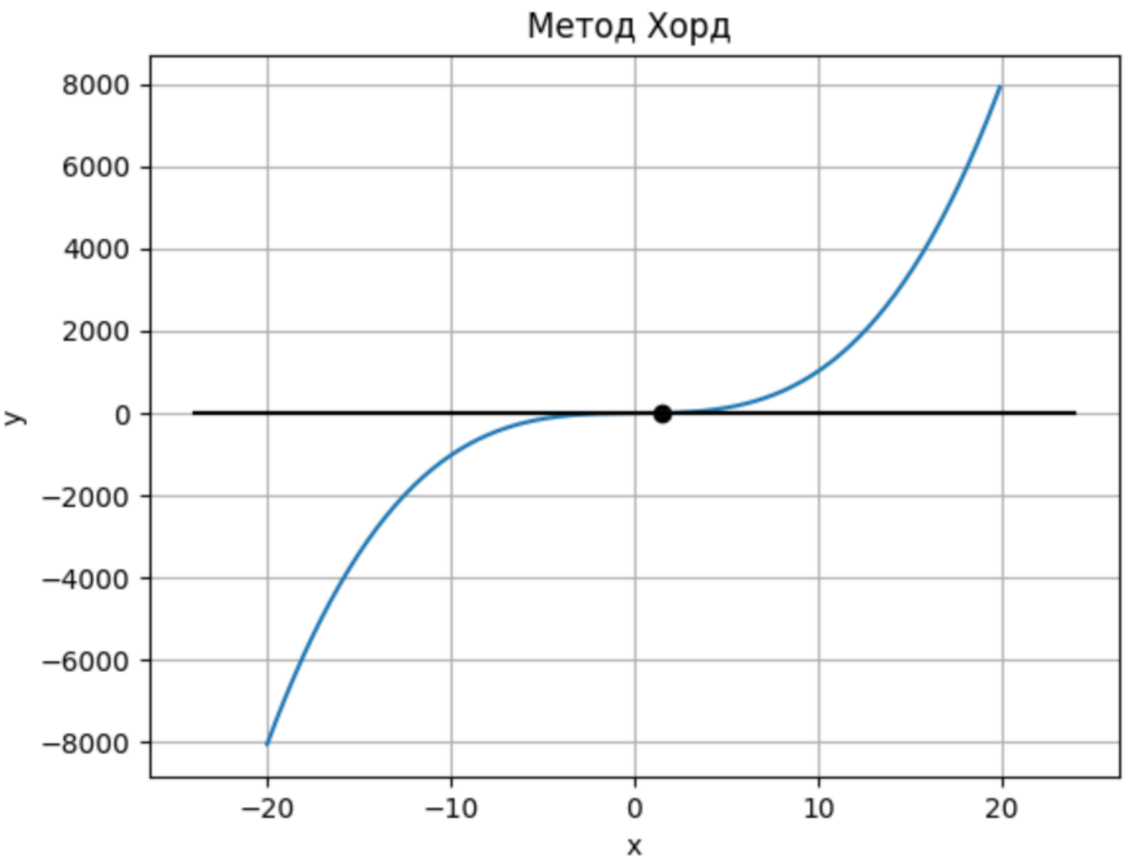


решала
127.0.0.1:8000/nonlinear/
Назад
Вычисление нелинейного уравнения
Метод вычисления:
Касательных
Функция f(x):
x**3+2*x-6
Начало вычисления:
-20.0
Конец вычисления:
20.0
Точность:
1000
Интервал решения после отделения корней:
[1.4560000000014104, 1.4570000000014103]
Ответ:
Массив решений: 1.456164
Показать график
Рассчитать
Поиск
ENG
22:16
13.12.2023



Метод касательных
8000
6000
4000
2000
0
−2000
−4000
−6000
−8000
y
−20
−10
0
10
20
x





Сравнительный анализ полученных результатов:

Возьмём контрольный пример $x^3 + 2x - 6 = 0$ на отрезке от -20 до +20 с точностью 10^{-6} с ответом х ≈ 1.456162, мы выяснили, что метод касательных является самым точным.

Вывод:

Нам удалось, верно, реализовать численные методы решения нелинейных уравнений разными методами в нашем веб-приложении. В ходе работы получилось выяснить, что самым точным методом для решения нелинейных уравнений является метод касательных.

Отчёт Суворов Р.М.

Тема: «Численные методы решения нелинейных уравнений»

Используемое оборудование: ПК, языки программирования: Python, HTML, CSS, SQL, JavaScript; используемые сторонние библиотеки: Django, Matpotlib, Numpy, SQLlite3; среда разработки Visual Studio Code.

Постановка задачи: Изучить численные методы деления отрезка пополам, касательных, хорд

Математическая модель:

1. Метод деления отрезка пополам

Метод можно использовать если: 1) $f(x)$ – непрерывна на $[a; b]$; 2) $f(x)$ принимает на концах отрезка разные знаки, то есть $f(a)f(b) < 0$

Алгоритм:

1) За $x0$ выбираем середину отрезка $[a; b]$, то есть $x0 = a+b$ /2 Будем искать корень на одном из отрезков: $[a; x0\ ]$ и $[x0; b]$. На концах этих отрезков функция $f(x)$ принимает разные знаки ($f(a)f(x0\ ) < 0$ и $f(x0\ )f(b) < 0$).

2) Вычислить $f(a)$ и $f(x0\ )$.

3) Найти $f(a)f(x0\ )$ и проверить выполнение неравенства $f(a)f(x0\ ) < 0$. Если неравенство верно, то корень принадлежит отрезку $[a; x0\ ]$ и далее пункт 5.

4) Вычислить $f(x0\ )$ и $f(b)$. Найти $f(x0\ )f(b)$. Если $f(x0\ )f(b) < 0$, то корень принадлежит отрезку $[x0; b]$ и далее пункт 7.

5) Продолжаем сужать отрезок. Делим его ещё раз пополам и находим следующий предполагаемый корень $x1 = a + x0\ 2$

6) Проверяем выполнение неравенства $f(x1\ )f(x0\ ) < 0$. Если верно, то корень $\epsilon[x0; x1\ ]$

7) Опять сужаем отрезок. Делим его и находим ещё предполагаемый корень $x1 = b + x0/2$

8) Деление отрезка продолжаем до тех пор, пока его длина не будет превосходить заданную точность h.

2. Метод касательных

1. **Итерационный процесс:**
   - Формула итерации: $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$
   - Где x_k - текущее приближение, $f(x_k)$ - значение функции в точке x_k, а $f'(x_k)$ - производная функции в точке x_k.

Процесс останавливается, когда значение функции достаточно близко к нулю или достигнута заданная точность

3. Метод хорд

Теория метода хорд:

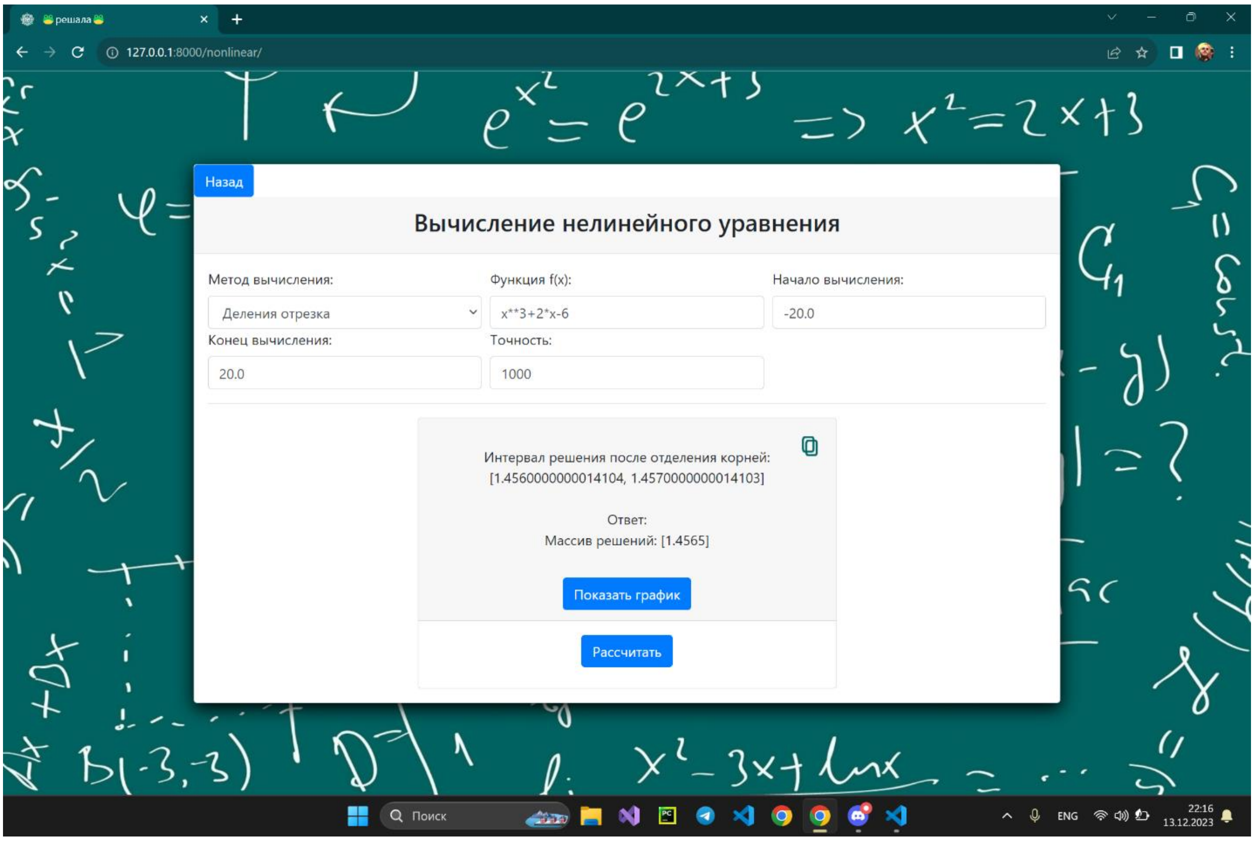
1. **Итерационный процесс:**
   - $x_{k+1} = x_k - \frac{f(x_k)\cdot(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$
   - Где x_k и x_{k-1} - текущее и предыдущее приближения, соответственно.
2. **Условие остановки:**
   - Остановить процесс, когда значение функции достаточно близко к нулю или достигнута заданная точность.

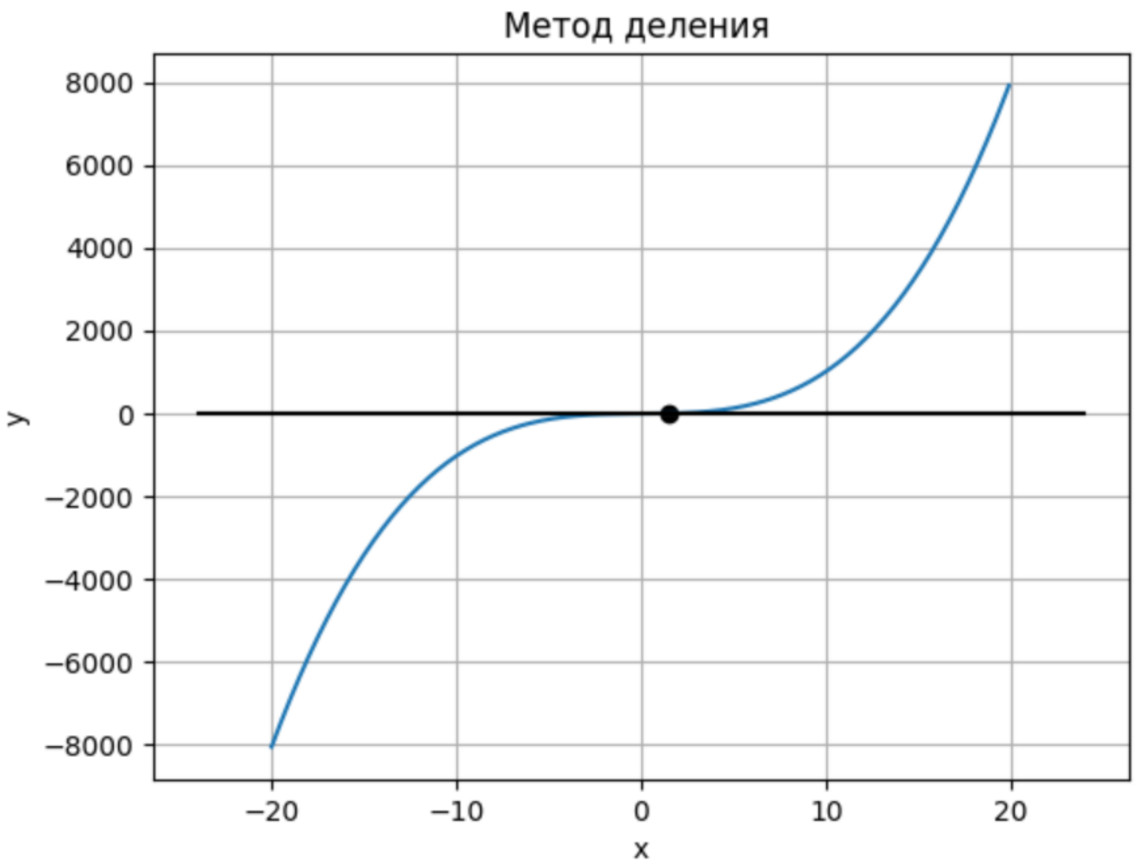
Код программы:

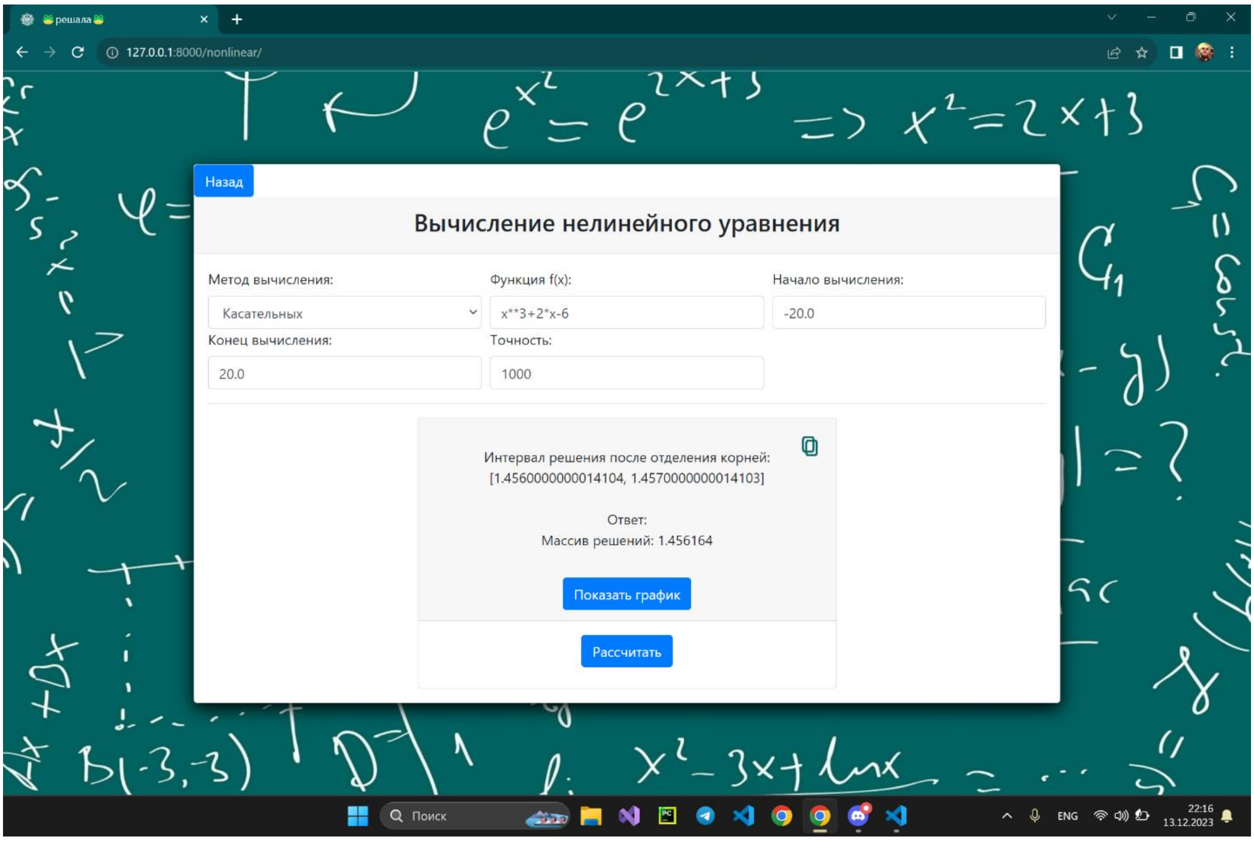
https://github.com/webbsalad/Computational_Mathematics_LW1

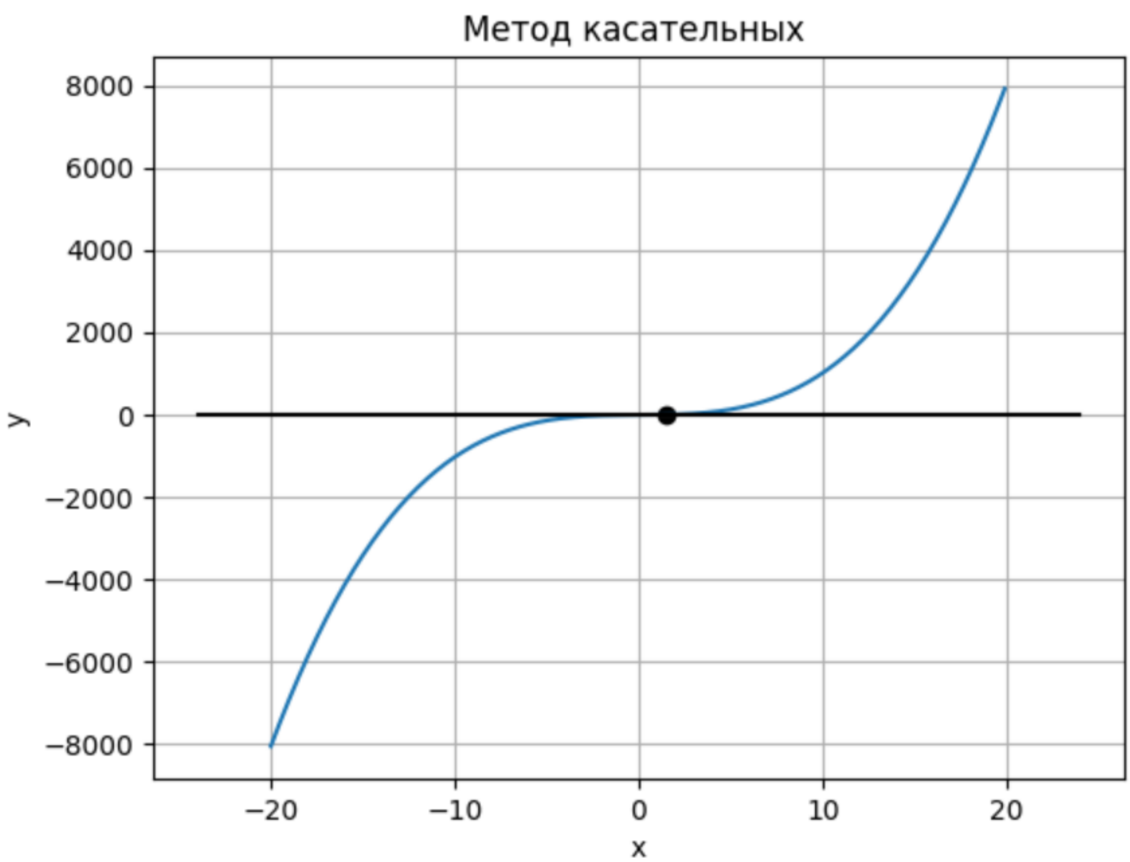
Для открытия веб-приложения, необходимо скачать файлы, в терминале зайди в директорию, выполнить команду «python manage.py run server».

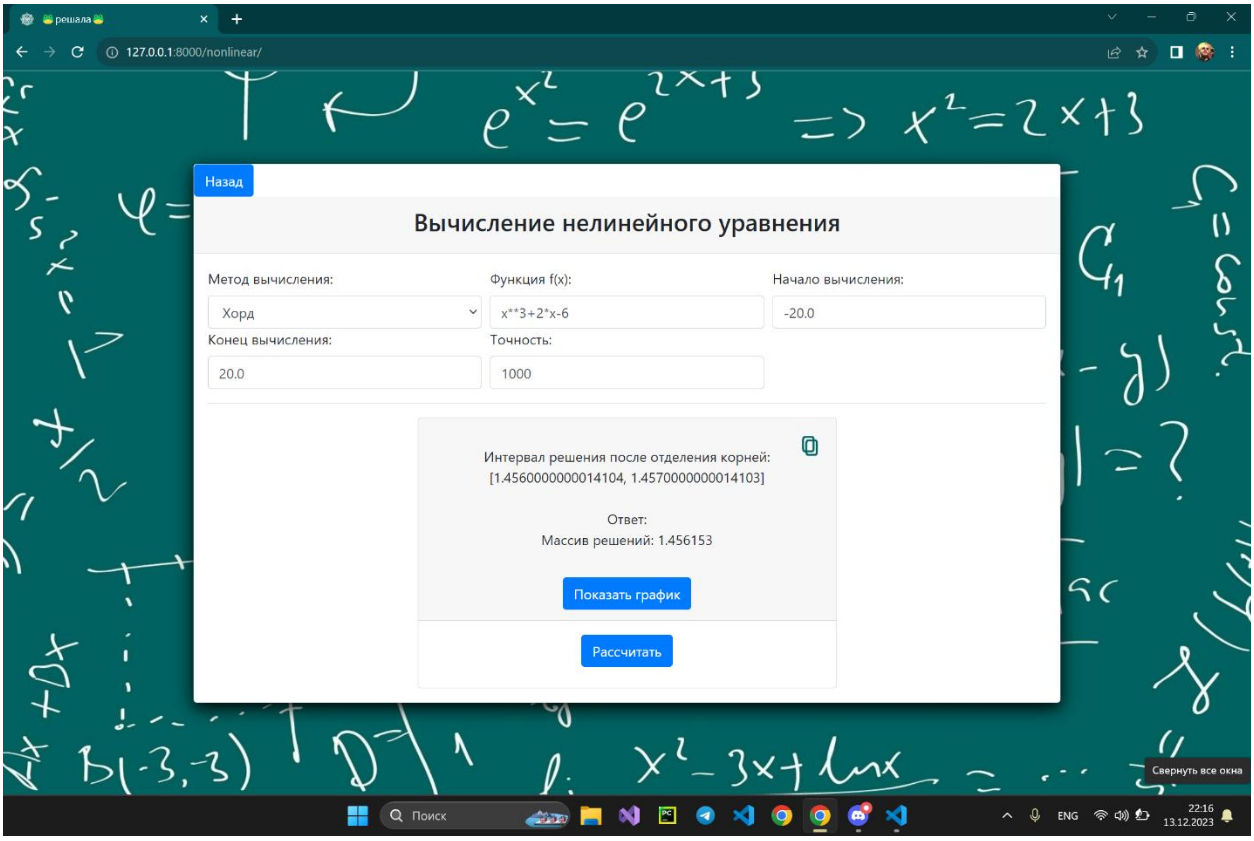
Результат выполнения работы:

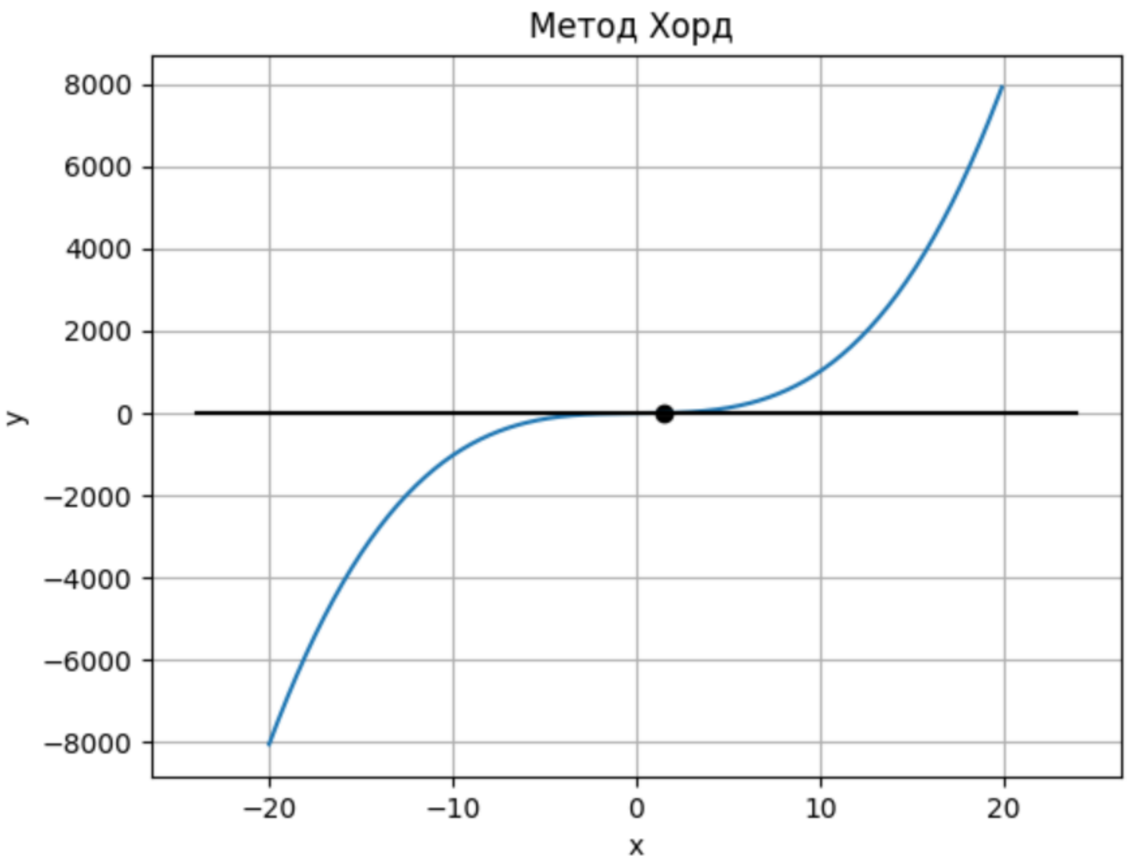

решала
127.0.0.1:8000/nonlinear/
Назад
Вычисление нелинейного уравнения
Метод вычисления:
Деления отрезка
Функция f(x):
x**3+2*x-6
Начало вычисления:
-20.0
Конец вычисления:
20.0
Точность:
1000
Интервал решения после отделения корней:
[1.4560000000014104, 1.4570000000014103]
Ответ:
Массив решений: [1.4565]
Показать график
Рассчитать
Поиск
ENG
22:16
13.12.2023


Метод деления
8000
6000
4000
2000
0
−2000
−4000
−6000
−8000
y
−20
−10
0
10
20
x


решала
127.0.0.1:8000/nonlinear/
Назад
Вычисление нелинейного уравнения
Метод вычисления:
Касательных
Функция f(x):
x**3+2*x-6
Начало вычисления:
-20.0
Конец вычисления:
20.0
Точность:
1000
Интервал решения после отделения корней:
[1.4560000000014104, 1.4570000000014103]
Ответ:
Массив решений: 1.456164
Показать график
Рассчитать


Метод касательных
8000
6000
4000
2000
0
−2000
−4000
−6000
−8000
y
−20
−10
0
10
20
x





Сравнительный анализ полученных результатов:

Возьмём контрольный пример $x^3 + 2x - 6 = 0$ на отрезке от -20 до +20 с точностью 10^{-6} с ответом x ≈ 1.456162, мы выяснили, что метод касательных является самым точным.

Вывод:

Нам удалось, верно, реализовать численные методы решения нелинейных уравнений разными методами в нашем веб-приложении. В ходе работы получилось выяснить, что самым точным методом для решения нелинейных уравнений является метод касательных.