Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

по дисциплине: «Вычислительная математика»

Студент	Фуров Павел Павлович
Группа	РК6-61Б
Тип задания	Лабораторная работа №1
Тема лабораторной работы	Численное и автоматическое дифференцирование

Студент ____________________ **Фуров П.П.**
подпись, дата *фамилия, и.о.*

Преподаватель ____________________ **Першин А.Ю.**
подпись, дата *фамилия, и.о.*

Оценка ______________________________________

Москва, 2021 г.

Оглавление

Задание на лабораторную работу

Даны функции

$$f(x) = \frac{x^5 + 2x^4 - 3x^3 + 4x^2 - 5}{x + 2}, \qquad (1)$$

$$\mathrm{g}(x) = x^2 \sin(x), \qquad (2)$$

и узел $x_0 = 2$.

Требуется (базовая часть):

1. Вывести улучшенную формулу численного дифференцирования для нахождения значения первой производной функции, используя в качестве основной формулы формулу численного дифференцирования 1-го порядка и применив к ней экстраполяцию Ричардсона. Экстраполяция Ричардсона основывается на том, что аппроксимацию значения производной можно записать в следующем виде:
$$M = N_1(h) + K_1 h + K_2 h^2 + K_3 h^3 + \cdots, \qquad (3)$$
где M – точное значение производной в заданной точке, $N_1(h)$ – формула численного дифференцирования первого порядка и $K_1, K_2, K_3, \cdots$ – константы, появляющиеся в остаточном члене при разложении его в ряд Тейлора. Используя $h/2$ вместо h в выражении выше, выведите формулу, которая будет иметь второй порядок точности. Можно ли получить эту формулу другими способами? Если да, то какими?
2. Написать функцию diff1(x_0, h, f), которая возвращает значение первой производной функции на основе центральной формулы численного дифференцирования 1-го порядка в точке x_0 для шага дифференцирования h
3. Написать функцию diff1(x_0, h, f), которая возвращает значение первой производной функции на основе центральной формулы численного дифференцирования 2-го порядка в точке x_0 для шага дифференцирования h
4. Рассчитать производную $g'(x)$ в точке x_0 для множества значений $h \in [10^{-16}; 1]$ сначала с помощью функции diff1, а затем с помощью функции diff2. Для обоих случаев постройте log-log графики зависимости абсолютной погрешности численного дифференцирования от шага дифференцирования.

Требуется (продвинутая часть):

1. Для случая функций diff1 и diff2 из базовой части ответить на следующие вопросы:
   - Каким образом на log-log графике можно увидеть порядок формулы дифференцирования? Доказать это формульно и продемонстрировать на графике по аналогии с лекциями.
   - Каков оптимальный шаг дифференцирования, при котором абсолютная погрешность минимальна? С чем связано существование такого минимума? Обосновать свой ответ, ссылаясь на данные log-log графика.
2. Реализовать прямой режим автоматического дифференцирования с использованием дуальных чисел:
   - Написать класс AutoDiffNumber, использование которого позволяет построить вычислительный граф.
   - Написать функцию forward_autodiff(fun_args), вычисляющую значение производной функции, вычислительной граф которой передаётся в fun_args.
   - Продемонстрировать корректность автоматического дифференцирования на примере функции $f(x)$ и 100 случайных точек $x \in [-1;\ 1]$ и сравнить полученные значения производных с аналитически полученными значениями.
   - Вычислить значения производных в тех же точках, используя функции diff1 и diff2, и сравнить полученные значения с аналитическими и полученными с помощью автоматического дифференцирования.

Цель выполнения лабораторной работы

Численное и автоматическое дифференцирование – два принципиально различных подхода к нахождению производной функции. Особенностью численных методов нахождения производных является остаточный член – верхняя граница абсолютной погрешности метода. Очевидная задача – максимально уменьшить эту верхнюю границу. Одним из способов улучшения формул численного дифференцирования является экстраполяция Ричардсона, которая позволяет получить аппроксимацию большей точности, используя формулу меньшей точности. Другой особенностью является вычислительная неустойчивость формул численного дифференцирования, поэтому другая очевидная задача сводится к минимизации полной погрешности (погрешности с учётом ошибок округления) за счёт подбора оптимального шага дифференцирования. Автоматическое дифференцирование позволяет получить точное значение производной (вплоть до машинного эпсилон). Одной из версий автоматического дифференцирования является прямой режим с дуальными числами.

Цели базовой части:

- Продемонстрировать работоспособность экстраполяции Ричардсона на примере улучшения формулы численного дифференцирования первого порядка.
- Выведенную формулу получить другими способами, либо объяснить, почему это невозможно.
- Реализовать исходную и новую формулы дифференцирования на выбранном языке программирования, написав соответствующие функции.
- Продемонстрировать вычислительную неустойчивость обеих формул, построив график зависимости полной погрешности численного дифференцирования от шага дифференцирования.

Цели продвинутой части:

- Найти связь между порядком формулы дифференцирования и log-log графиком зависимости полной погрешности от шага h и доказать её формульно.
- Найти оптимальный шаг дифференцирования и обосновать его существование.
- Реализовать прямой режим автоматического дифференцирования с использованием дуальных чисел
- Продемонстрировать корректность автоматического дифференцирования и на основе результатов сравнить численный метод и метод автоматического дифференцирования.

Ход выполнения лабораторной работы

1. Базовая часть

1.1. Вывод улучшенной формулы

Исходя из определения формулы (3), «улучшаемая» формула численного дифференцирования первого порядка основывается на разложении функции в ряд Тейлора. В лекции 4 было выведено значение ряда в точке $x_0 + h$:

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2!}f''(x_0) + \frac{h^3}{3!}f'''(x_0) + \cdots + \frac{h^n}{n!}f^{(n)}(x_0); \quad (4)$$

Отсюда получено выражение первой производной в точке x_0:

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \frac{h}{2!}f''(x_0) - \frac{h^2}{3!}f'''(x_0) - \cdots - \frac{h^{n-1}}{n!}f^{(n)}(x_0); \quad (5)$$

Легко сопоставить формулу (5) с формулой (3):

$$M = f'(x_0),\ N_1(h) = \frac{f(x_0 + h) - f(x_0)}{h},\ K_1 = -\frac{f''(x_0)}{2!}, \cdots,\ K_{n-1} = -\frac{f^{(n)}(x_0)}{(n)!}; \ (6)$$

Чтобы улучшить аппроксимацию, нужно исключить $K_1 h$ из ошибки. Согласно заданию, необходимо использовать $h/2$ вместо h в выражении (3):

$$M = N_1\left(\frac{h}{2}\right) + K_1\frac{h}{2} + K_2\frac{h^2}{4} + K_3\frac{h^3}{8} + \cdots; \quad (7)$$

Легко заметить, что для исключения второго слагаемого необходимо вычесть выражение (3) из удвоенного выражения (7):

$$2 \cdot (7) - (3) = M = 2N_1\left(\frac{h}{2}\right) - N_1(h) - K_2\frac{h^2}{2} - K_3\frac{3h^3}{4} - \cdots; \quad (8)$$

Для получения искомой формулы необходимо подставить сопоставления (6) в выражение (8):

$$f'(x_0) = 4\frac{f\left(x_0 + \frac{h}{2}\right) - f(x_0)}{h} - \frac{f(x_0 + h) - f(x_0)}{h} + \frac{h^2}{12}f'''(x_0) + \cdots;$$

или

$$f'(x_0) = \frac{-3f(x_0) + 4f\left(x_0 + \frac{h}{2}\right) - f(x_0 + h)}{h} + \frac{h^2}{12}f'''(x_0) + \cdots; \qquad (9)$$

Для получения искомой формулы, необходимо в формуле (9) сделать обратную замену $h/_2$ на h, а бесконечную последовательность, основываясь на теореме Коши о среднем значении, заменить остаточным членом:

$$f'(x_0) = \frac{-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)}{2h} + \frac{h^2}{3}f'''\big(\xi(x_0)\big),$$

где $\xi(x_0) \in (x_0; x_0 + 2h)$.

Таким образом, экстраполяция Ричардсона позволила получить формулу, имеющую второй порядок точности из формулы численного дифференцирования первого порядка.

1.1.1. Второй способ получения формулы

Теперь, зная итоговый вид формулы, нетрудно догадаться, что её можно получить из разложений функции $f(x)$ в ряд Тейлора в точках $x_0 + h$ и $x_0 + 2h$. Значение ряда в первой точке уже записано в формуле (4), а для второй точки ряд имеет следующий вид:

$$f(x_0 + 2h) = f(x_0) + 2hf'(x_0) + 2h^2f''(x_0) + \frac{4h^3}{3}f'''(x_0) + \cdots; \qquad (10)$$

Прийти к искомой формуле возможно с помощью следующей линейной комбинации выражений (10) и (3):

$$4 \cdot (3) - (10) = 4f(x_0 + h) - f(x_0 + 2h) =$$
$$= 3f(x_0) + 2hf'(x_0) - \frac{2h^3}{3}f'''\big(\xi(x_0)\big);$$

Откуда можно получить выражение для $f'(x_0)$:

$$f'(x_0) = \frac{-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)}{2h} + \frac{h^2}{3} f'''\big(\xi(x_0)\big),$$

которое совпадает с искомой формулой, полученной ранее.

1.1.2. Третий способ получения формулы

Вспомнив лекцию 4, а именно выведенную там формулу для метода дифференцирования многочлена Лагранжа

$$f'(x_j) = \sum_{i=1}^{n} f(x_i) l_i'(x_j) + \frac{\prod_{i=1, i\neq j}^{n} (x_j - x_i)}{n!} f^{(n)}\left(\xi(x_j)\right), \tag{11}$$

нетрудно заметить, что при использовании трёх узлов с промежутком h, второе слагаемое в выражении будет сильно напоминать остаточный член формулы численного дифференцирования второго порядка. Итак, логично попробовать использовать качестве узлов для формулы (11) точки x_0, $x_0 + h$ и $x_0 + 2h$. Для начала следует найти производную базисного полинома Лагранжа для трёх узлов:

$$l_i'(x) = \left(\frac{(x - x_{k_0})(x - x_{k_1})}{(x_i - x_{k_0})(x_i - x_{k_1})}\right)' = \frac{2x - x_{k_0} - x_{k_1}}{(x_i - x_{k_0})(x_i - x_{k_1})},$$

где x_{k_0}и x_{k_1}– два других узла.

Тогда выражение (11) принимает вид:

$$f'(x_0) = f(x_0)\frac{2x_0 - 2x_0 - 3h}{-h(-2h)} + f(x_0 + h)\frac{2x_0 - 2x_0 - 2h}{h(-h)} +$$
$$+ f(x_0 + 2h)\frac{2x_0 - 2x_0 - h}{2h \cdot h} + \frac{(x_0 - x_0 - h)((x_0 - x_0 - 2h)}{6} f'''\big(\xi(x_0)\big);$$
$$f'(x_0) = -\frac{3}{2h} f(x_0) + \frac{2}{h} f(x_0 + h) - \frac{1}{2h} f(x_0 + h) + \frac{h^2}{3} f'''\big(\xi(x_0)\big),$$

или

$$f'(x_0) = \frac{-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)}{2h} + \frac{h^2}{3} f'''\big(\xi(x_0)\big),$$

что соответствует формуле, полученной с помощью экстраполяции Ричардсона.

Таким образом, была продемонстрирована работоспособность экстраполяции Ричардсона, и доказана справедливость полученного выражения с помощью получения его двумя другими способами.

1.2. Разработка функции diff1

Функция diff1 должна возвращать значение производной функции f на основе центральной формулы численного дифференцирования 1-го порядка, то есть на основе следующей формулы:

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h} \tag{12}$$

Функция принимает 3 аргумента: точка x_0, шаг дифференцирования h и дифференцируемая функция f.

Листинг 1 – функция diff1, возвращающая значение производной функции f на основе формулы (12)

```
def diff1(x_0, h, f):
    return (f(x_0+h)-f(x_0))/h
```

1.3. Разработка функции diff2

Функция diff2 должна возвращать значение производной функции f на основе новой формулы численного дифференцирования 2-го порядка, то есть на основе следующей формулы:

$$f'(x_0) \approx \frac{-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)}{2h} \tag{13}$$

Функция принимает 3 аргумента: точка x_0, шаг дифференцирования h и дифференцируемая функция f.

Листинг 2 – функция diff2, возвращающая значение производной функции f на основе формулы (13)

```
def diff2(x_0, h, f):
    return (-3*f(x_0)+4*f(x_0+h)-f(x_0+2*h))/(2*h)
```

1.4. Анализ абсолютной погрешности численного дифференцирования

Для начала требуется аналитически найти значение производной функции $g(x)$:

$$g'(x) = \left(x^2 \sin(x)\right)' = 2x\sin(x) + x^2\cos(x) \tag{14}$$

Далее необходимо реализовать саму функцию и её производную с погрешностью, сопоставимой с машинным эпсилон:

Листинг 3 – функция g, реализующая одноимённую функцию и функция g_deriv, реализующая формулу (14)

```
def g(x):
    return x*x*np.sin(x)

def g_deriv(x)
    return 2*x*np.sin(x)  + np.cos(x)*x**2
```

В данном случае значение абсолютной погрешности есть модуль разности между производной вычисленной «напрямую», и вычисленной с помощью функций diff1 и diff2.

Листинг 4 – функция abs_err, возвращающая полную погрешность для метода численного дифференцирования diff в точке x с шагом h

```
def abs_err(x, h, diff):
    return np.abs(g_deriv(x)-diff(x,h,g))
```

Для 150 значений шага $h \in [10^{-16}; 1]$, заданных на равномерной логарифмической сетке, выведено значение abs_err для функций diff1 и diff2:

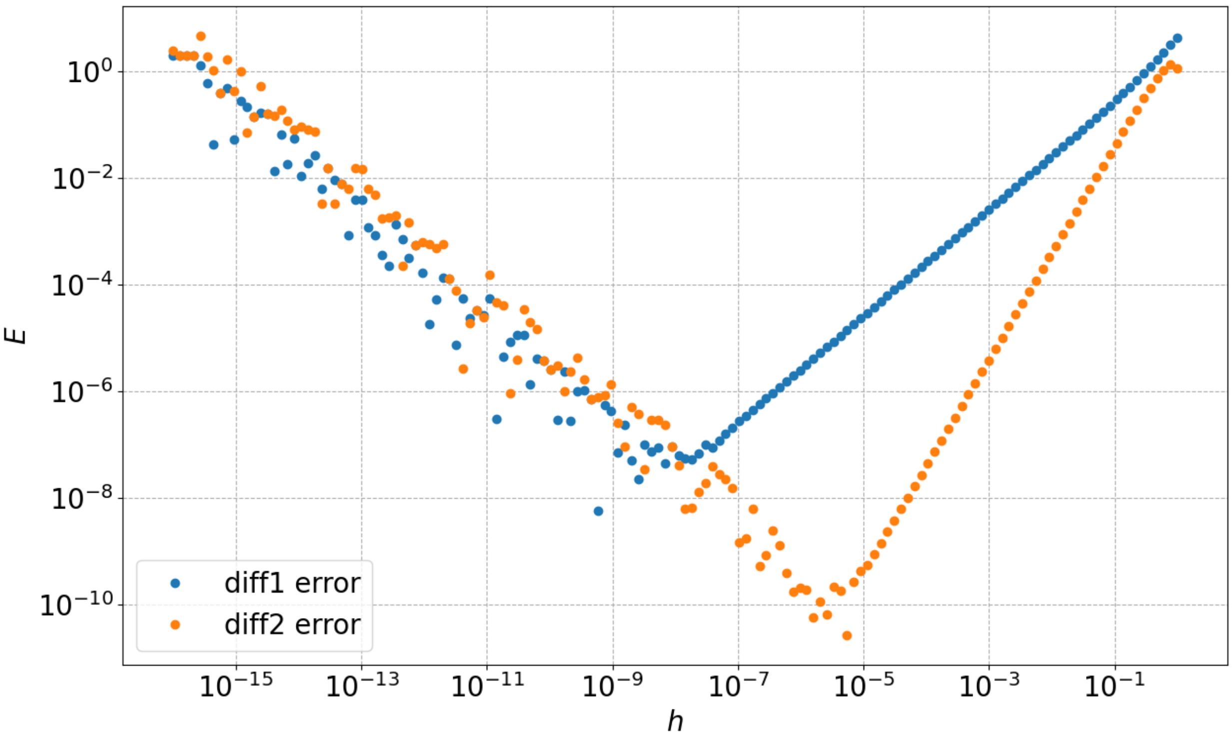

Рисунок 1 – график зависимости полной погрешности E от шага h при вычислении производной с помощью формул численного дифференцирования 1-го (синий) и 2-го (оранжевый) порядка.

Из графика, изображённого на рисунке 1, можно сделать следующие выводы:

- Погрешность вычисления убывает лишь до определённого значения шага, далее погрешность начинает расти за счёт того, что ошибка округления начинает превосходить остаточный член формулы метода численного дифференцирования. Это видно и на графике – для формулы второго порядка точности ошибка округления начала влиять раньше.
- При использовании формулы численного дифференцирования 1-го порядка можно добиться точности порядка 10^{-7} (на точку, которая находится ещё ниже, ориентироваться не стоит, она лишь говорит о случае везения, при котором ошибка округления оказалась мала за счёт того, что истинное значение само было довольно «круглым»). При этом соответствующий шаг дифференцирования $\approx 10^{-8}$

- Рассуждая аналогично, для формулы численного дифференцирования 2-го порядка можно добиться точности порядка 10^{-10}, то есть точность данной формулы действительно оказалась выше. При этом соответствующий шаг дифференцирования $\approx 10^{-5}$, как и говорилось выше, ошибка округления начала доминировать при большем h.

В базовой части наглядно показано явление вычислительной неустойчивости численного дифференцирования и, следовательно, существование оптимального шага, способ нахождения которого будет рассмотрен в продвинутой части лабораторной работы.

2. Продвинутая часть

2.1. Подробный анализ свойств методов численного дифференцирования

2.1.1. Определение порядка формулы по log-log графику

Пусть $\tilde{x} = \log(x)$ – линейное расстояние от 0 до точки x на оси абсцисс, а $\tilde{f}(x) = \log(f(x))$ - линейное расстояние от 0 до $f(x)$ на оси ординат log-log графика. Тогда для $f(x) = x^n$:

$$\tilde{f}(x) = \log(x^n) = n \cdot \log(x) = n \cdot \tilde{x};$$

Таким образом, график функции $f(x) = x^n$ в логарифмическом масштабе представляет из себя прямую с тангенсом угла наклона, равным n. Значит, определить порядок функции численного дифференцирования можно по тангенсу угла наклона прямой, которая является графиком абсолютной погрешности, находящейся в пределах остаточного члена, который и является степенной функцией, зависимой от h. Наглядно это продемонстрировано на рисунке 2, где к исходному графику дополнительно выведены графики степенных функций 1-го и 2-го порядка. По клеткам также видны тангенсы наклона сформировавшихся ранее «прямых», которые равны 1 и 2 соответственно.

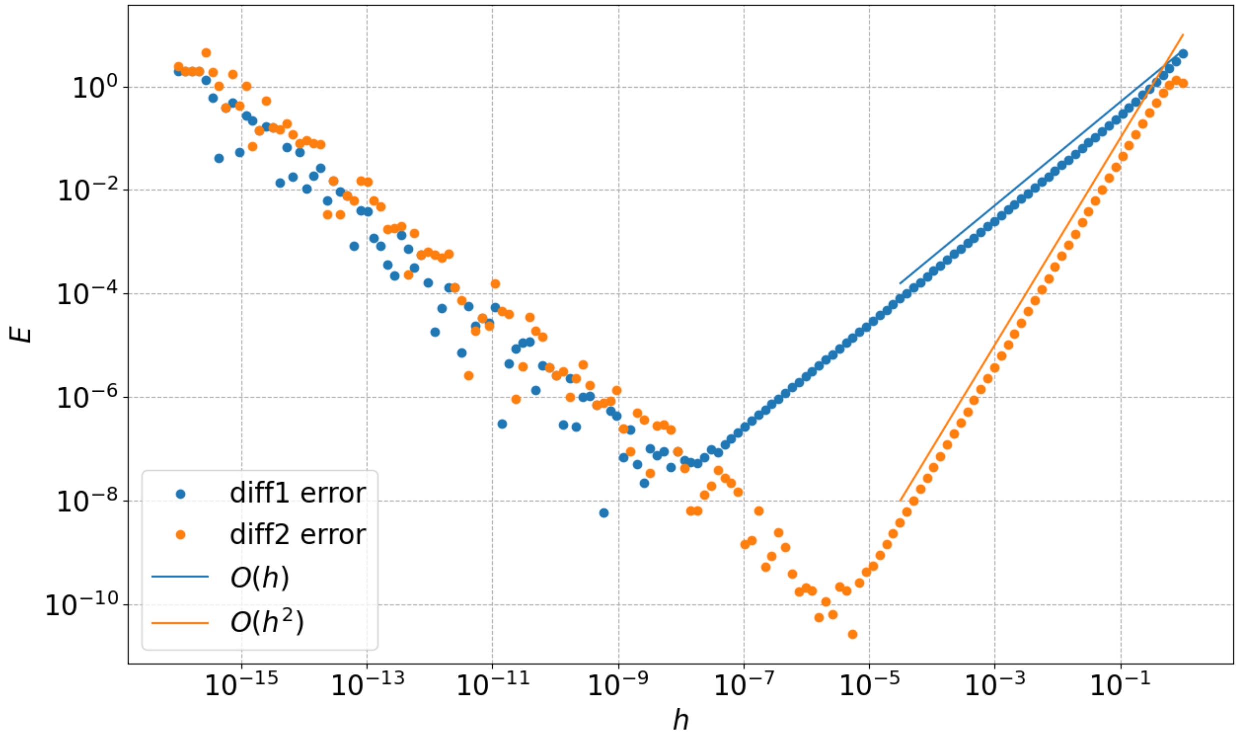

Рисунок 2 – график зависимости полной погрешности E от шага h, полученный ранее в сравнении с графиками соответствующих степенных функций.

2.1.2. Поиск оптимального шага дифференцирования для diff1

Функция численного дифференцирования 1-го порядка имеет вид:

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \frac{h}{2} f''(\xi(x_0));$$

Предположим, что при округлении значений $f(x_0 + h)$ и $f(x_0)$ возникает вычислительная погрешность, равная $e(x_0 + h)$ и $e(x_0)$:

$$f(x_0) = \tilde{f}(x_0) + e(x_0)$$

$$f(x_0 + h) = \tilde{f}(x_0 + h) + e(x_0 + h)$$

Тогда полная погрешность:

$$E = \left| \frac{f(x_0 + h) - f(x_0)}{h} - \frac{h}{2} f''\big(\xi(x_0)\big) - \frac{\tilde{f}(x_0 + h) - \tilde{f}(x_0)}{h} \right| =$$

$$= \left| \frac{e(x_0 + h) - e(x_0)}{h} - \frac{h}{2} f''\big(\xi(x_0)\big) \right|$$

Очевидным предположением является то, что ошибка округления ограничена ϵ а модуль второй производной ограничен числом M:

$$|e(x_0+h)|, |e(x_0)| \leq \epsilon,$$

$$\left|f''\big(\xi(x_0)\big)\right| \leq M,$$

В таком случае:

$$E \leq \left|\frac{e(x_0+h)-e(x_0)}{h} - \frac{h}{2}f''\big(\xi(x_0)\big)\right| \leq \frac{2\epsilon}{h} + \frac{hM}{2};$$

По формуле видно, что при $h \to 0$ оптимального h погрешность возрастает пропорционально $\mathrm{O}(h^{-1})$, а при $h \to \infty$ пропорционально $\mathrm{O}(h)$. Оптимальным шагом является экстремум полученного выражения:

$$\frac{d}{dh}\left(\frac{2\epsilon}{h} + \frac{hM}{2}\right) = -\frac{2\epsilon}{h^2} + \frac{M}{2} = 0 \Longrightarrow h^{(opt)} = \sqrt{\frac{4\epsilon}{M}};$$

По умолчанию numpy использует вещественный тип данных numpy.float64, для которого машинный эпсилон равен $2.22044604925 \cdot 10^{-16}$. Значение M можно найти из максимального значения второй производной на интервале $(x_0;\ x_0 + h_{max})$, то есть на интервале $(2;\ 3)$.

$$g'(x) = \big(x^2\sin(x)\big)' = 2x\sin(x) + x^2\cos(x)$$

$$g''(x) = 2\sin(x) + 4x\cos(x) - x^2\sin(x)$$

$$g'''(x) = 6\cos(x) - 6x\sin(x) - x^2\cos(x) = 0 \Longrightarrow x^{ext} \approx 2.98147$$

Максимальное значение на интервале есть максимальное значение среди экстремумов и значений на границах интервала:

$$g''(2) \approx -5.14776;\ g''(3) \approx -12.86775;\ g''(2.98147) \approx -12.87172$$

Таким образом, $M = 12.87172$, тогда оптимальный шаг:

$$h^{(opt)} = \sqrt{\frac{4\epsilon}{M}} = 8.30676 \cdot 10^{-9}$$

Листинг 5 – Вспомогательный код, вычисляющий значения второй производной для разных точек и оптимальный h

```
def g_derivv(x):
    return 2*np.sin(x) + np.cos(x)*x*4-np.sin(x)*x**2

print(g_derivv(2), g_derivv(3), g_derivv(2.98147))
h_opt=np.sqrt(4*np.finfo(float).eps/abs(g_derivv(2.98147)))
print(h_opt)
```

На графике (рисунок 3) красным цветом было выведено значение полной погрешности при оптимальном h и добавлен график функции со степенью -1.

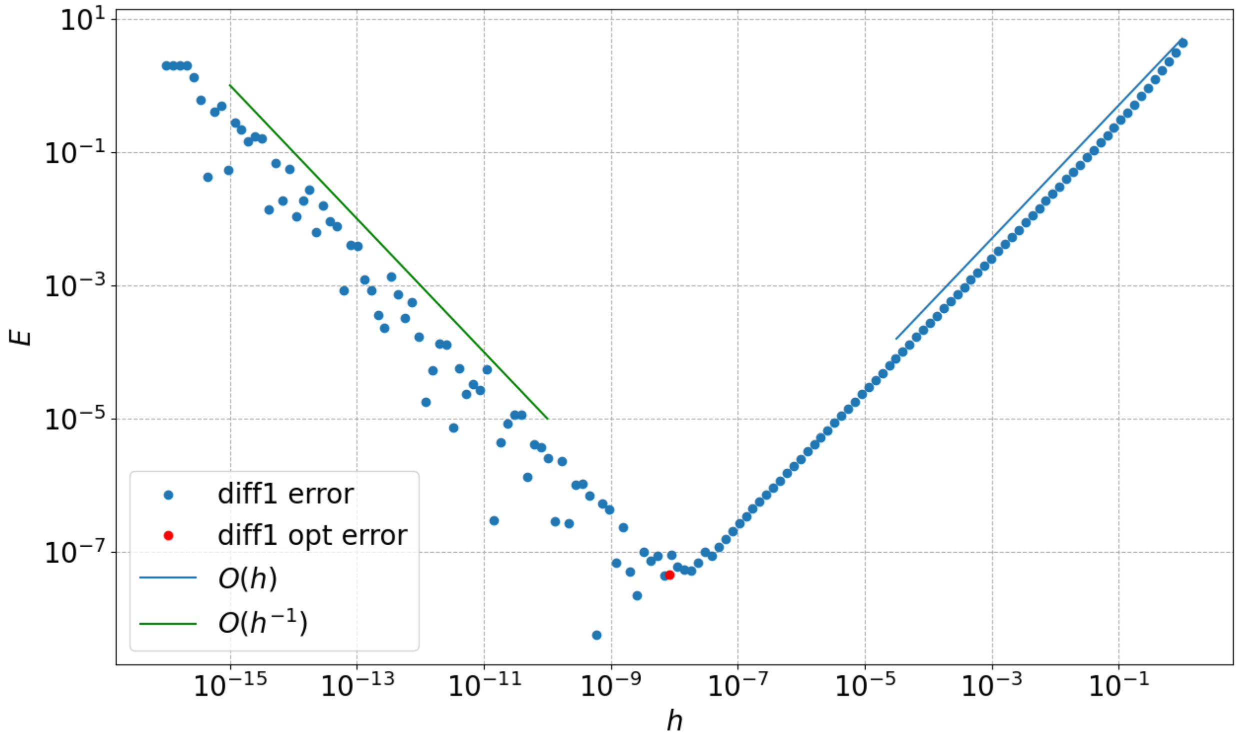

Рисунок 3 – график зависимости полной погрешности E от шага h, полученный ранее в сравнении с графиками соответствующих степенных функций.

Примечательно, что судя по графику, получено не самое минимально возможное значение погрешности, однако, как говорилось в выводе базовой части, самая низко расположенная точка лишь результат того, что ошибка округления оказалась мала, что происходит абсолютно случайно. Зато по графику видно, что точка, соответствующая $h^{(opt)}$ расположена ниже всех среди «закономерных» точек, а такую точку и требовалось найти.

2.1.3. Поиск оптимального шага дифференцирования для diff2

Абсолютно аналогично можно найти оптимальный шаг для формулы численного дифференцирования 2-го порядка:

$$f'(x_0) = \frac{-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)}{2h} + \frac{h^2}{3} f'''\big(\xi(x_0)\big),$$

Полная погрешность:

$$E = \left| \frac{-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)}{2h} + \frac{h^2}{3} f'''\big(\xi(x_0)\big) - \right.$$

$$\left. - \frac{-3\tilde{f}(x_0) + 4\tilde{f}(x_0 + h) - \tilde{f}(x_0 + 2h)}{2h} \right| =$$

$$= \left| \frac{-3e(x_0) + 4e(x_0 + h) - e(x_0 + 2h)}{2h} + \frac{h^2}{3} f'''\big(\xi(x_0)\big) \right| \le \frac{4\epsilon}{h} + \frac{Mh^2}{3}$$

По формуле видно, что при $h \to 0$ оптимального h погрешность возрастает пропорционально $O(h^{-1})$, а при $h \to \infty$ пропорционально $O(h^2)$. Оптимальным шагом является экстремум полученного выражения:

$$\frac{d}{dh}\left(\frac{4\epsilon}{h} + \frac{Mh^2}{2}\right) = -\frac{4\epsilon}{h^2} + \frac{2hM}{3} = 0 \Longrightarrow h^{(opt)} = \sqrt[3]{\frac{6\epsilon}{M}};$$

Значение M можно найти из максимального значения второй производной на интервале $(x_0;\ x_0 + 2h_{max})$, то есть на интервале $(2;\ 4)$.

$$g'''(x) = 6\cos(x) - 6x\sin(x) - x^2\cos(x)$$

$$g^{(4)}(x) = -12\sin(x) - 8x\cos(x) - x^2\sin(x) = 0 \Longrightarrow x^{ext} \approx 2.02463$$

Максимальное значение на интервале есть максимальное значение среди экстремумов и значений на границах интервала:

$$g'''(2) \approx -11.74386;\ g'''(4) \approx 24.69969;\ g''(2.02463) \approx -11.75146$$

Таким образом, $M = 24.69969$, тогда оптимальный шаг:

$$h^{(opt)} = \sqrt[3]{\frac{6\epsilon}{M}} = 3.77833 \cdot 10^{-6}$$

Листинг 6 – Вспомогательный код, вычисляющий значения третьей производной для разных точек и оптимальный h

```
def g_deriv3(x):
    return 6*np.cos(x) - np.sin(x)*x*6-np.cos(x)*x**2

print(g_deriv3(2), g_deriv3(4), g_deriv3(2.02463))
h_opt=(6*np.finfo(float).eps/abs(g_deriv3(4)))**(1/3)
print(h_opt)
```

На графике (рисунок 4) красным цветом было выведено значение полной погрешности при оптимальном h и добавлен график функции со степенью -1.

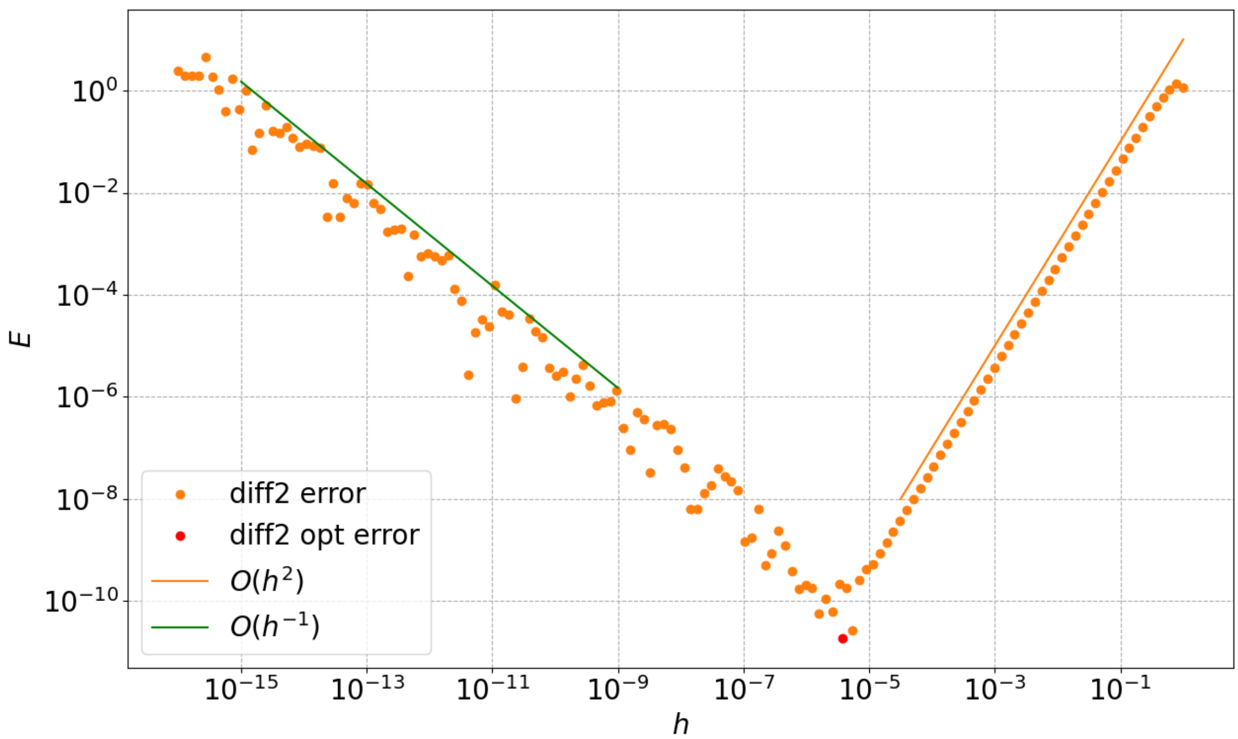

Рисунок 4 – график зависимости полной погрешности E от шага h, полученный ранее в сравнении с графиками соответствующих степенных функций.

Точка расположена в условном экстремуме графика, что говорит о правильности её нахождения. Также она, в отличии от предыдущего случая, расположена ниже других точек.

2.2. Реализация прямого режима автоматического дифференцирования

2.2.1. Создание класса AutoDiffNumber

Для выполнения требуется реализовать класс для дуальных чисел, в котором нужно перегрузить все нужные арифметические операции для функции $f(x)$. Для каждого перегруженного оператора будет использоваться проверка типа, чтобы можно было работать как с дуальными числами, так и с обычными константами. Это требуется для операторов _add_, _sub_, _mul_, _truediv_, и их r-аналогов, так как операндом может являться как обычное число, так и дуальное. Для _pow_ данная проверка не нужна.

Листинг 7 – Пример перегрузки оператора «+» и инициализации данного класса.

```
class AutoDiffNumber:
    def __init__(self, a, b):
        self.a=a
        self.b=b
    def __add__(self, other):
        if isinstance(other, AutoDiffNumber):
            return AutoDiffNumber(self.a+other.a,self.b+other.b)
        else:
            return AutoDiffNumber(self.a+other,self.b)
```

2.2.2. Создание функции forward_autodiff

В лекции 5 было сказано о важном свойстве дуальных чисел, основываясь на котором и будет реализована искомая функция:

$$f(a + b\epsilon) = f(a) + b\epsilon f'(a)$$

Класс дуальных чисел (по крайней мере включающий нужные нам операторы) уже реализован, легко заметить, что функция от дуального числа возвращает дуальное число, и если b принять равным единице, то дуальная часть и будет производной в точке a.

Тогда от функции forward_autodiff требуется совсем немного – создать дуальное число вида $(x_0 + \epsilon)$ с помощью уже готового конструктора и передать это число функции f, а из полученного числа извлечь дуальную часть

Листинг 8 – реализация функций f(x) и forward_autodiff.

```
def f(x):
    return (x**5 + 2*x**4 - 3*x**3 + 4*x**2 - 5)/(x + 2)

def forward_autodiff(f,x):
    dual = AutoDiffNumber(x, 1)
    f_dual = f(dual)
    return(f_dual.b)
```

2.2.3. Проверка корректности автоматического дифференцирования

Для начала требуется найти аналитическое значение производной функции $f(x)$:

$$f'(x) = \left(\frac{x^5 + 2x^4 - 3x^3 + 4x^2 - 5}{x + 2}\right)' = \frac{4x^5 + 16x^4 + 10x^3 - 14x^2 + 16x + 5}{(x + 2)^2}$$

Листинг 9 – функция, вычисляющая значение производной аналитически.

```
def f_deriv(x):
    return(4*x**5 + 16*x**4 + 10*x**3 - 14*x**2 + 16*x + 5) / ((x + 2)**2)
```

На график были выведены значения производных в 100 случайных точках от -1 до 1, посчитанных с помощью функции f_deriv и с помощью функции forward_autodiff:

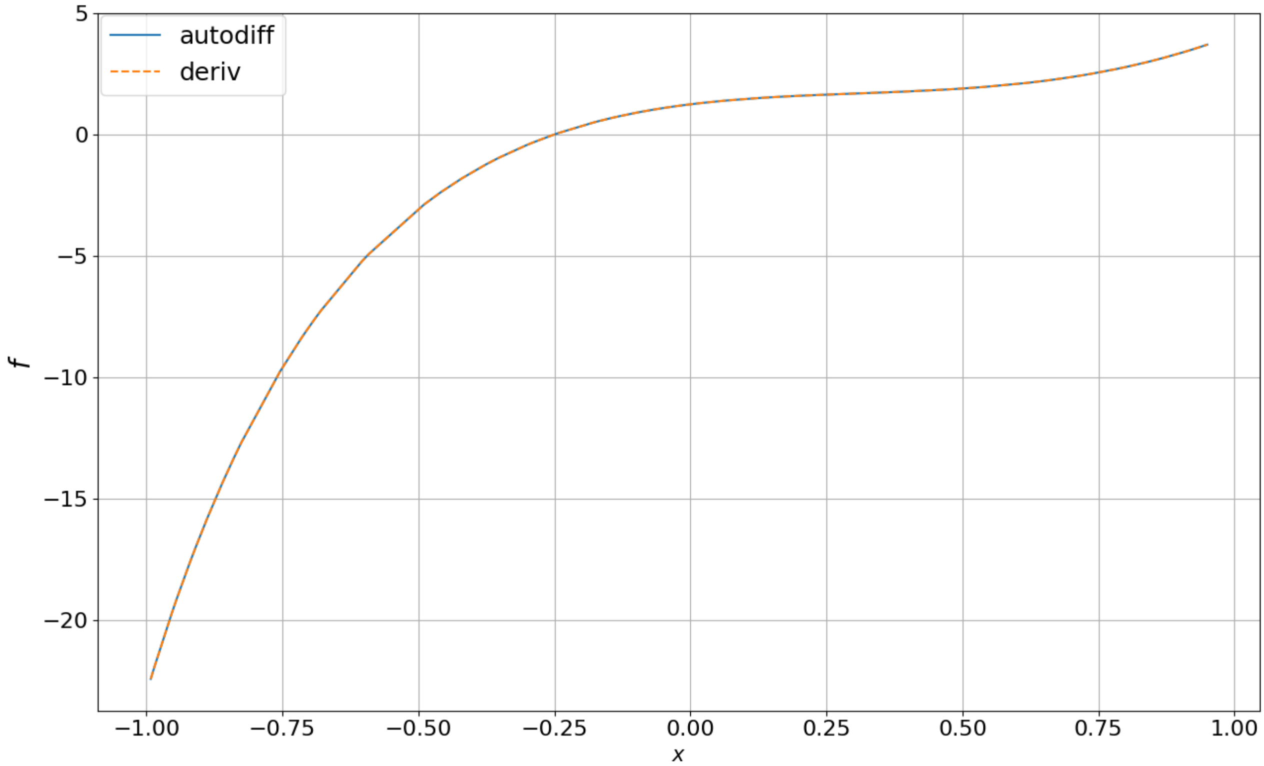

Рисунок 5 – графики производной функции $f(x)$, полученные аналитически, и с помощью прямого режима автоматического дифференцирования

Как видно из графика, автоматическое дифференцирование работает корректно.

2.2.4. Сравнение методов автоматического и численного и численного дифференцирования:

В качестве шага h для каждой из формул численного дифференцирования были взяты найденные ранее оптимальные шаги $h^{(opt)}$. Чтобы не засорять график, линия, соответствующая методу автоматического дифференцирования, была убрана, т.к. свою корректность своей работы данный метод ранее уже показал:

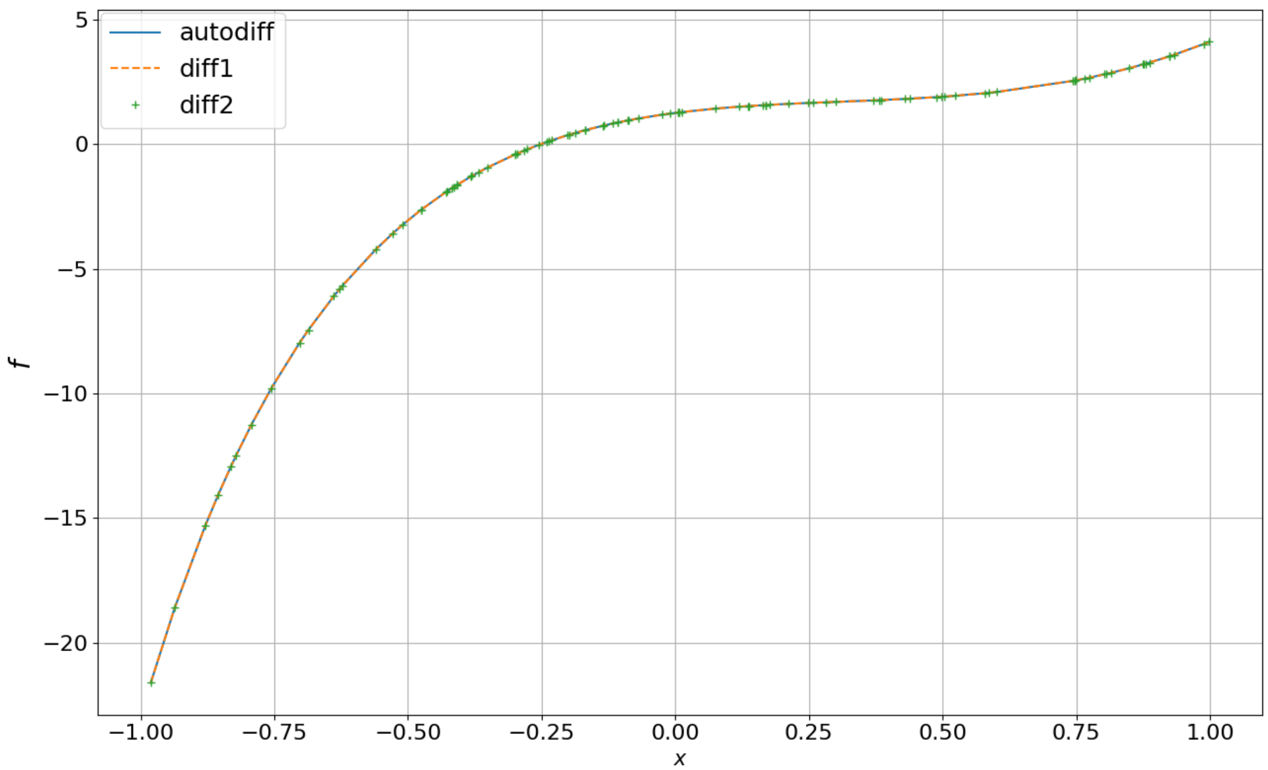

Рисунок 6 – графики производной функции $f(x)$, полученные аналитически, и с помощью формул численного дифференцирования 1-го и 2-го порядка с оптимальным для них шагом.

Как видно из рисунка 5, численные методы также хорошо справляются со своей задачей, так как полная погрешность несоизмеримо мала относительно сетки графика. Поэтому для демонстрации вычислительной неустойчивости и неточности, были построены графики с слишком маленьким и слишком большим шагом соответственно (в легенду выведены значения текущего шага):

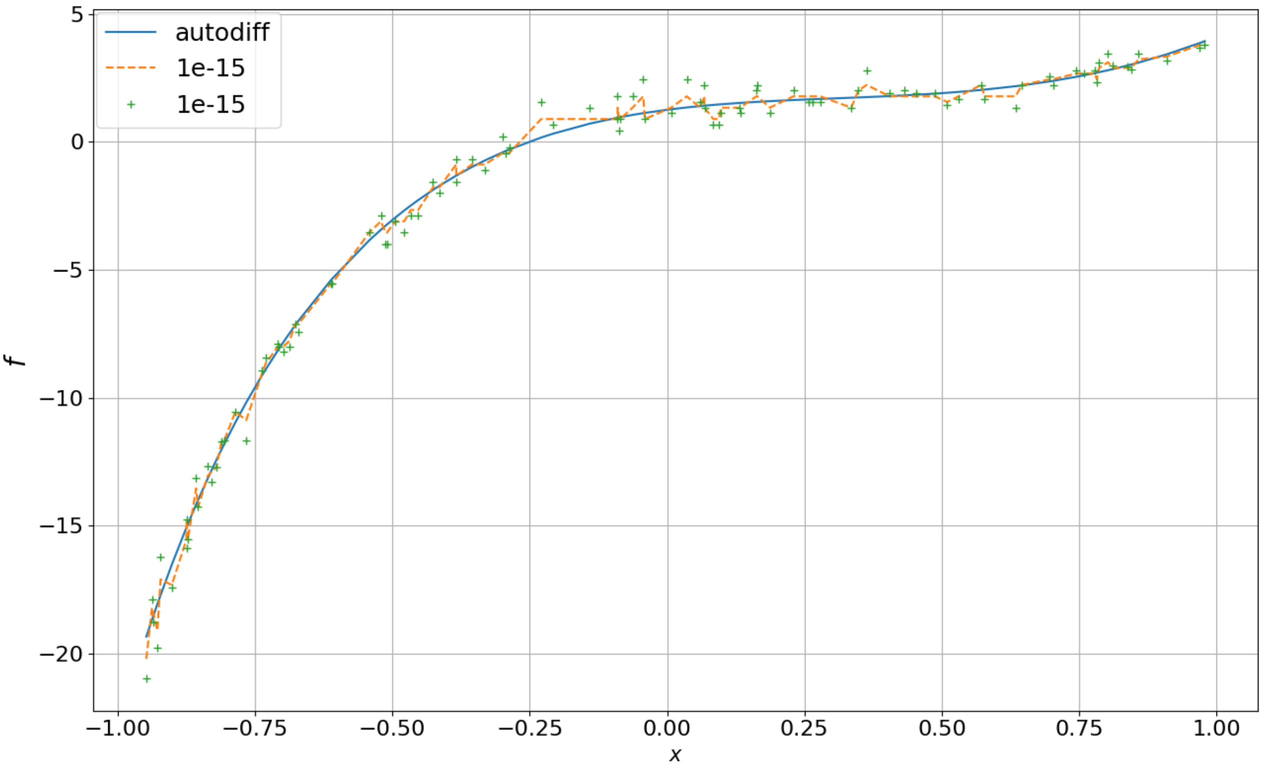

Рисунок 7 – график, демонстрирующий вычислительную неустойчивость при слишком малом шаге.

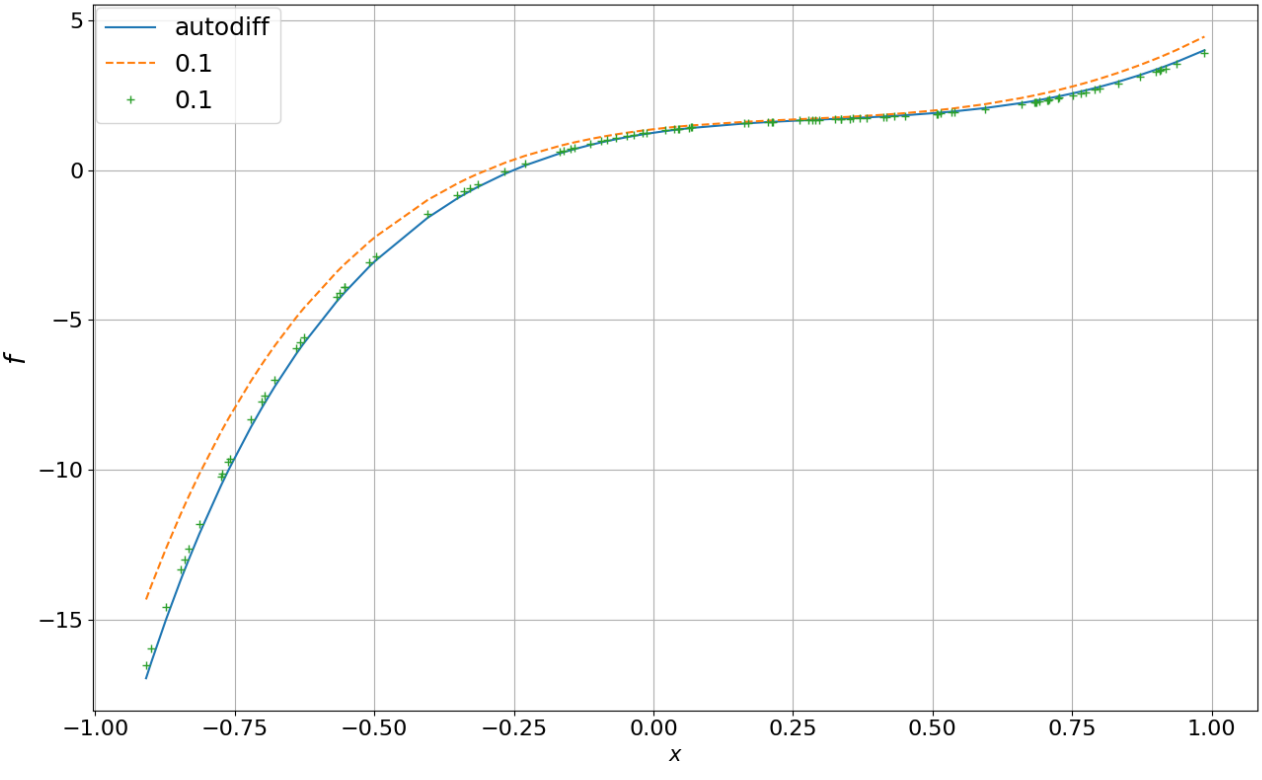

Рисунок 8 – график неточность формул численного дифференцирования при слишком большом шаге.

На рисунке также 8 видно, что при равном шаге, график производной, вычисленной с помощью формулы численного дифференцирования более высокого порядка, отклоняется меньше.

Заключение

В ходе выполнения данной лабораторной работы на практике были реализованы два принципиально разных метода дифференцирования функции, а полученные результаты были глубоко проанализированы.

Были рассмотрены различные способы увеличения порядка точности формул численного дифференцирования, а сами формулы были сравнены на практике.

Также был выявлен существенный недостаток методов численного дифференцирования – вычислительная неустойчивость, однако был найден метод нахождения оптимального шага дифференцирования, благодаря которому, этот недостаток становится контролируемым и безопасным.

Изученный ранее прямой метод автоматического дифференцирования был также реализован на практике. Благодаря свойству дуальных чисел, можно находить значение производной в точке элегантно и без вычислительной погрешности поэтому и на практике данный метод используется гораздо шире в сравнении с численными методами дифференцирования.

Список использованных источников

1. **Першин А.Ю.** *Лекции по вычислительной математике (черновик)*. [archrk6.bmstu.ru] // Кафедра РК6 МГТУ им. Н.Э. Баумана, Москва, 2020, 145.
2. **Першин А.Ю.** *Вычислительная математика, лекция №4*. Видеохостинг «YouTube» [https://www.youtube.com/watch?v=d5xHDd_K_jE] // (дата обращения (04.04.2021).
3. **Першин А.Ю.** *Вычислительная математика, лекция №5*. Видеохостинг «YouTube» [https://www.youtube.com/watch?v=6TcTplqaMxg] // (дата обращения (04.04.2021).