|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 |  | Шашко Олег Владимирович |
| Группа |  | РК6-61Б |
| Тип задания |  | лабораторная работа |
| Тема лабораторной работы |  | Численное дифференцирование и интегрирование |

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шашко О.В.**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Соколов А.П.**

*подпись, дата фамилия, и.о.*

*Москва, 2022 г.*

Оглавление

[Задание на лабораторную работу 3](#_Toc32399212)

[Цель выполнения лабораторной работы 3](#_Toc32399213)

[Выполненные задачи 3](#_Toc32399214)

[1. @Название раздела в соответствии с задачей 1@ 4](#_Toc32399215)

[2. @Название раздела в соответствии с задачей 2@ 4](#_Toc32399216)

[Заключение 4](#_Toc32399217)

[Список использованных источников 4](#_Toc32399218)

# Задание на лабораторную работу

1.1 Требования к знаниям для выполнения

Для выполнения лабораторной работы обучающийся должен обладать знаниями:  
– владеть навыками разработки программного обеспечения на языке Python (ре-

комендуется) или С++ на базовом уровне;  
– владеть навыками использования программных инструментов: numpy;  
– знать понятия: численное дифференцирование, численное интегрирование, ме-

тод наименьших квадратов, тригонометрические полиномы, быстрое преобразование Фурье.

1.2 Численное дифференцирование и интегрирование (вариант 1)

Известно, что порядок точности формулы численного дифференцирования опреде- ляется степенью, в которую возводится шаг дифференцирования h в выражении для остаточного члена формулы (он имеет форму 𝑂(h𝑁 )). Порядок формулы также легко продемонстрировать, если построить log-log график зависимости абсолютной погрешности численного дифференцирования в некоторой заданной точке от шага дифференцирования h. В базовой части лабораторной работы предлагается постро- ить подобный график для формулы численного дифференцирования второго поряд- ка. Формулы численного дифференцирования более высокого порядка часто выводят с помощью разложения в ряд Тейлора и последующего вычисления значений ряда в нескольких узлах. Увеличение количества узлов приводит к увеличению порядка формулы и, с другой стороны, к увеличению количества арифметических операций с плавающей запятой, что увеличивает вычислительную погрешность. В продвинутой части этого задании предлагается вывести формулу численного дифференцирова- ния 4-го порядка и проанализировать ее вычислительную устойчивость. Аналогич- ный анализ можно произвести и для формул численного интегрирования, то есть квадратур. В базовой части лабораторной работы предлагается исследовать состав- ную формулу Симпсона, в то время как в продвинутой части задания необходимо вывести и исследовать квадратуру Гаусса степени точности 5.

Задача 6 (численное дифференцирование и интегрирование)

Даны функции:

(1)

С узлом и функция

Заданная на интервале

Требуется (базовая часть):  
1. Написать функцию diff2(x\_0, h, f), которая возвращает значение первой произ-

водной функции 𝑓 на основе центральной формулы численного дифференцирования 2-го порядка в точке 𝑥0 для шага дифференцирования h27.

2. Рассчитать производную 𝑔1′ (𝑥) в точке 𝑥0 = 2 для множества значений h ∈ [10−16;1] с помощью функции diff2. Построите log-log графики зависимости абсолютной погрешности численного дифференцирования от шага дифференцирования

3. Написать функцию composite\_simpson(a, b, n, f) численного интегрирования

функции f на интервале [a; b] по n узлам с помощью составной формулы Симпсона.

4. Рассчитать интеграл с помощью составной формулы Симпсона для

множества значений 𝑛 ∈ [3; 9999]. Постройте log-log график зависимости абсолютной погрешности численного интегрирования от шага интегрирования.

Требуется (продвинутая часть):

1. Вывести общую центральную формулу численного дифференцирования 4-го порядка вместе с остаточным членом, аппроксимирующую первую производную по 5 узлам:

(3)

Продемонстрируйте, что формула действительно имеет 4-й порядок точности.

2. Написать функцию diff4(x\_0, h, f), которая возвращает значение первой производной функции 𝑓 на основе центральной формулы численного дифференцирования 4-го порядка в точке x\_0 для шага дифференцирования h.

3. Рассчитать производную в точке 𝑥0 = 2 для множества значений с помощью функции diff4. Добавьте log-log график зависимости абсолютной погрешности численного дифференцирования от шага дифференцирования к существующему графику для diff2. Для каждого случая (diff2 и diff4) ответьте на следующие вопросы:

– Каким образом на log-log графике можно увидеть порядок формулы дифференцирования? Докажите это формульно и продемонстрируйте на графике по аналогии с лекциями.

– Совпадает ли порядок выведенной формулы дифференцирования на log-log графике с ее действительным порядком?

– Каков оптимальный шаг дифференцирования, при котором абсолютная погрешность минимальна? С чем связано существование такого минимума? Обоснуйте свой ответ, ссылаясь на данные log-log графика.

– Сравните оптимальный шаг дифференцирования и соответствующую минимально достижимую погрешность для формул 2-го и 4-го порядка. Как вы думаете, чем обоснована разница между ними?

4. Сравните порядок формулы, полученный с помощью графика для составной формулы Симпсона, с аналитическим порядком точности составной формулы Симпсона. Существует ли оптимальный шаг интегрирования для данной формулы, минизимирующий достижимую погрешность? Обоснуйте свой ответ.

5. С помощью теоремы о корнях многочленов Лежандра, доказанной в лекциях, вывести квадратуру Гаусса, имеющую степень точности 5. Сколько узлов необходимо для использования такой квадратуры?

6. Написать функцию gauss\_quad5(f) численного интегрирования функции f с помощью квадратуры Гаусса пятой степени точности.

7. Доказать, что квадратура Гаусса имеет степень точности 5, с помощью следующего вычислительного эксперимента:

– постройте последовательность полиномов 𝑃0(𝑥), 𝑃1(𝑥), 𝑃2(𝑥), 𝑃3(𝑥), 𝑃4(𝑥), 𝑃5(𝑥), 𝑃6(𝑥), имеющих степени соответственно 0, 1, 2, 3, 4, 5, и 6, используя случайно сгенерированные значения коэффициентов полиномов33;

– проинтегрируете их на интервале [0; 2] аналитически и с помощью выведенной квадратуры Гаусса;

– посчитайте абсолютную погрешность и сделайте вывод о степени точности вы- ведённой квадратуры;

– все выкладки и посчитанные значения должны быть в отчете.

# Цель выполнения лабораторной работы

**Цель выполнения лабораторной работы** – научится использовать и реализовывать численные методы интегрирования и дифферинцирования

# Выполненные задачи

* **Задача 1 – написать функцию diff2.**
* **Задача 2 – рассчитать производную в точке**
* **Задача 3 - Написать функцию composite\_simpson**
* **Задача 4 – рассчитать интеграл**
* **Задача 5 – вывести общую центральную формулу численного дифференцирования**
* **Задача 6 – написать функцию diff4**
* **Задача 7 – рассчитать производную в точке**
* **Задача 8 – сравнить порядок формулы**
* **Задача 9 – вывести квадратуру Гаусса**
* **Задача 10 - Написать функцию gauss\_quad**

**1. Численное дифференцирование и интегрирование**

1. Задача 1 – написать функцию diff2(x\_0, h, f)

Значение производной функции вычисляется по формуле 4:

*(4)*

Реализация функции, возвращающей результат вычислений по формуле 4 представлена на листинге 1:

def diff2(x, h, f):  
 return (f(x + h / 2) - f(x - h / 2)) / h

Листинг 1 – реализация функции diff2

1. Задача 2 – рассчитать производную в точке

Для построения log-log графика погрешности вычислений нам понадобится два массива точек – значение шага h и значение производной, вычисленное с помощью функции из задачи 1, вызванной с параметрами .

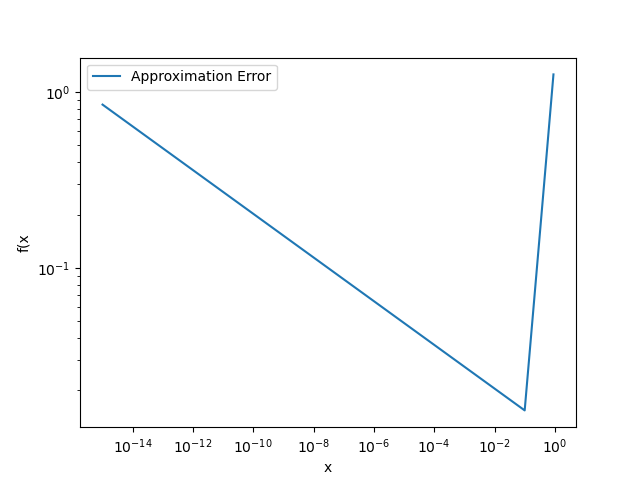
Создадим вспомогательный массив h\_values со значениями шага h с помощью функции np.arange – в качестве аргумента передадим границы отрезка и шаг равный 10e-4. Затем, итерируясь по массиву, вычислим для каждого h значение производной и запишем полученный результат в массив g1\_der\_values.

Теперь вычислим точное значение производной аналитически (получим ) и инициализируем третий массив approximation\_errors – в нем будут хранится абсолютные значения разностей между точным и приближенным значением производной. Программная реализация представлена на листинге 2:

g1\_der\_values = []  
h\_values = np.arange(10e-16, 1, 10e-4)  
for i in h\_values:  
 g1\_der\_values.append(diff2(2, i, g1))  
origin\_g1\_der = 3 \* pow(np.e, 2)  
approximation\_errors = []  
for value in g1\_der\_values:  
 approximation\_errors.append(abs(value - origin\_g1\_der))

*Листинг 2 – расчет погрешности вычислений*

График зависимости абсолютной погрешности численного дифференцирования от шага дифференцирования представлен на рисунке 1



*Рисунок 1 – log-log график*

1. Задача 3 – написать функцию composite\_simpson(a, b, n, f)

Формула Симпсона для нахождения приближенного значения интеграла имеет вид (5):

Реализация функции, вычисляющей значение по формуле (5) представлена на листинге 3:

def composite\_simpson(a: float, b: float, number\_of\_points: int, f):  
 h = (b - a) / number\_of\_points  
 k1 = 0  
 k2 = 0  
 for i in range(1, number\_of\_points, 2):  
 k1 += f(a + i \* h)  
 k2 += f(a + (i + 1) \* h)  
 return h / 3 \* (f(a) + 4 \* k1 + 2 \* k2)

*Листинг 3 – реализация функции composite\_simpson*

Задача 4 – рассчитать интеграл.

Для расчета интегралах по формуле Симпсона вызовем функцию из предыдущего пункта с параметрами , где i – элемент массива, состоящий из всех натуральных чисел в отрезка с шагом 10. Так же для вычисления погрешности метода необходимо вычислить точное значение интеграла аналитически:

В результате проведенных расчетов получаем список, состоящий из абсолютных величин разностей между точным и приближенным значением.

График зависимости величины абсолютной погрешности от числа точек интегрирования представлен на рисунке 2:

*Рисунок 2*

Задача 5 - вывести общую центральную формулу численного дифференцирования и продемонстрировать, что формула действительно имеет 4-й порядок точности

(5)

Выполним разложение формулы (5) в ряд Тейлора. Для этого, разложим по отдельности каждую скобку и затем перепищем выражение целиком в разложенном виде:

*(6.1)*

*(6.2)*

*(6.3)*

*(6.4)*

*(6.5)*

Подставим правые части выражений (6.1 – 6.4) в формулу и выполним группировку слагаемых:

(6.6)

Запишем систему уравнений (6.7):

(*6.7*)

(*6.8*)

Используя найденные коэффициенты (6.8), перепишем изначальную формулу:

(*6.9)*

В выражении (6.9) остаточный член имеет степень, равную 4. Таким образом, порядок точности для данной формулы равен 4.

Задача 6 – написать функцию diff4(x\_0, h, f)

Для реализации функции используем полученную в предыдущей задаче формулу (6.9). Программный код представлен на листинге 4:

def diff4(x, h, f):  
 return (-f(x - 2 \* h) - 8 \* f(x - h) + 8 \* f(x + h) + f(x + 2 \* h)) / (12 \* h)

*Листинг 4 – реализация функции diff4*

Задача 7 – рассчитать производную в точке

Для расчета воспользуемся массивом значений h, а также вычисленным аналитически точным значение производной из задачи 2. Полученный log-log график зависимости абсолютной погрешности от шага h представлен на рисунке

Теперь ответим на вопросы:

1) Каким образом на log-log графике можно увидеть порядок формулы дифференцирования? Докажите это формульно и продемонстрируйте на графике по аналогии с лекциями

Чтобы увидеть порядок, нужно построить на том же графике функцию вида , где a – коэффициент смещения, n – порядок точности

2) Совпадает ли порядок выведенной формулы дифференцирования на log-log графике с ее действительным порядком?

Да, совпадает

3) Каков оптимальный шаг дифференцирования, при котором абсолютная погрешность минимальна? С чем связано существование такого минимума? Обоснуйте свой ответ, ссылаясь на данные log-log графика

Оптимальный шаг дифференцирования для данных формул – это такой шаг, при котором суммарная погрешность дифференцирования и расчета функции минимальна, так как при так как при увлечении шага уменьшается погрешность дифференцирования, но возрастает погрешность вычисления функции.

Задача 8 – сравните порядок формулы

Порядок формулы Симпсона для графика равен 5, в то время как аналитический порядок точности равен 4. При этом не существует оптимального шага для достижения минимальной погрешности – при увеличении шага будет возрастать и точность вычислений. Происходит это потому, что в отличие от операции деления при дифференцировании, при интегрировании используется операция умножения, которая в свою очередь является вычислительно устойчивой.

Задача 9 – вывести квадратуру Гаусса

(*7.1*)

(7.2)

Возьмем многочлен Лежандра 5-й степени:

(*7.*2)

Корни многочлена (7.2) равны:

Наилучшее приближение интеграла будет достигаться в корнях многочлена

Задача 10 – написать функцию gauss\_quad5

Используя формулы (7.1) и (7.2) и предыдущей задачи, реализуем функцию

def gauss\_quad(f):  
 c = []  
 x = x\_5  
 x\_arg = Symbol('x')  
 for i in range(0, len(x)):  
 mul = ""  
 for j in range(0, len(x)):  
 if i != j:  
 mul += "(x-%s)/(%s-%s)\*" % (x[j], x[i], x[j])  
 mul = mul[:-1]  
 ans = sympy.integrate(mul, (x\_arg, -1, 1))  
 c.append(ans)  
 sum = 0  
 for i in range(0, len(x)):  
 sum += c[i] \* f(x[i] + 1)  
 return sum

*Листинг 5 – реализация функции gauss\_quad5*

Чтобы протестировать работу программного кода из листинга 5, возьмем полином Лежандра пятой степени (7.2) из предыдущей задачи

Вычислим интеграл аналитически:

В результате работы программного кода мы получили ответ, равный -1.694818351060768e-32. Из этого можно сделать вывод, что абсолютная погрешность вычислений, по крайней мере, для полинома пятой степени, очень мала.

# Заключение

Мы изучили методы приближенные методы дифференцирования и интегрирования, научились выводить их и вычислять порядок точности, а также реализовали их на языке программирования python

# Список использованных источников

1. **Першин А.Ю.** *Лекции по курсу “Вычислительная математика”.* //

Москва, 2018-2021. С. 140.

2. *Официальная документация numpy* [Электронный ресурс] // Режим доступа: https://numpy.org/, свободный. Яз. англ.

*Шашко О.В. Отчет о выполнении лабораторной работы по курсу “Вычислительная математика”. [Электронный ресурс] – Москва: 2022 – 13 с. URL: https://sa2systems.ru:88 (система контроля версий кафедры РК6)*