МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»

Лабораторная работа №1
по курсу «Математическое моделирование»

Определение параметров за косым скачком уплотнения на клине с использованием численных методов решения нелинейных уравнений

Выполнил: Э.В. Андриянов
Группа: М8О-412Б-22
Преподаватели: Н.А. Харченко

Москва, 2026

Условие

Вариант 6.

Задание №1: Определение параметров за косым скачком уплотнения

Рассматривается задача о стационарном сверхзвуковом обтекании клина с углом β потоком совершенного невязкого газа. Начальные параметры набегающего потока (индекс 1) заданы и имеют следующие значения:

- Начальное давление, p_1 = 165000 Па
- Начальная температура, T_1 = 300 K
- Число Маха, M_1 = 4.0
- Угол клина, β = 15°

Требуется определить угол наклона присоединенного косого скачка уплотнения θc, который находится как корень нелинейного уравнения (4) из методического пособия на интервале (0, π/2). После нахождения θc необходимо рассчитать параметры потока за скачком уплотнения (p_2, T_2, ρ_2, M_2) и определить установившийся режим течения. Для решения уравнения предписано использовать метод секущих.

Задание №2: Решение нелинейного уравнения

В рамках второго задания требуется найти корень нелинейного трансцендентного уравнения вида:

- e^x - x - 2 = 0

Поиск корня необходимо осуществить на заданном отрезке изоляции [-2, -1]. В качестве численного метода для решения данного уравнения следует применить метод Ньютона (метод касательных).

Задание №3: Сравнительный анализ методов

Целью третьего задания является проведение сравнительного анализа эффективности методов, использованных в предыдущих пунктах. Для этого необходимо повторно решить уравнение (4) из первого задания, но на этот раз применив к нему метод Ньютона.

- Сравнение методов должно проводиться по следующим критериям:
- Точность полученного значения корня.
- Количество итераций, потребовавшихся для достижения заданной точности.
- Процессорное время, затраченное на вычисления.

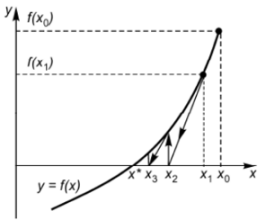
Теория

Метод секущих
Алгоритм

Дано уравнение $f(x) = 0$. Известен интервал $[a, b]$, на котором расположен корень x^*.

1. Задаем две начальные точки x_{n-1} и x_n на отрезке (на первой итерации за x_0 можно принять a или b, x_1 – выбираем произвольно для создания секущей).
2. Заменяем производную (из метода Ньютона) на разделенную разность:

$$f' = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}.$$



3. Строим прямую, проходящую через точки x_n и x_{n-1} (x_1 и x_0).
4. Определяем точку пересечения касательной с осью абсцисс:

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n).$$

5. Находим значение функции в этой точке $f(x_{n+1})$. Если $f(x_{n+1}) \approx 0$, то x_{n+1} – корень уравнения, если $f(x_{n+1}) \neq 0$, то возвращаемся в пункт 2.

Итерационный процесс повторяем до тех пор, пока не выполнится следующее условие:

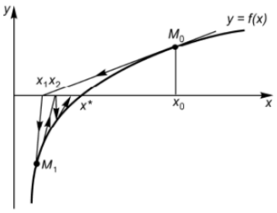
$$|x_{n+1} - x_n| < \delta \quad \text{или} \quad |f(x_n)| < \varepsilon.$$

Метод Ньютона
Алгоритм

Дано уравнение $f(x) = 0$. Известен интервал $[a, b]$, на котором расположен корень x^*.

1. Задаем начальное приближение x_n (на первой итерации x_0) на отрезке.
2. Находим производную функции $f'(x_n)$.
3. В точке x_n строим касательную к графику функции. Определяем точку пересечения x_{n+1} касательной с осью абсцисс:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$



4. Находим значение функции в этой точке $f(x_{n+1})$. Если $f(x_{n+1}) \approx 0$, то x_{n+1} – корень уравнения, если $f(x_{n+1}) \neq 0$, то повторяем пункт 2.

Метод является **условно** сходящимся, то есть необходимо выполнение условия: $|ff''| < (f')^2$ на всем рассматриваемом промежутке $[a, b]$.

Итерационный процесс повторяется, пока не выполнится одно из следующих условий:

$$|x_{n+1} - x_n| < \delta \quad \text{или} \quad |f(x_n)| < \varepsilon.$$

Метод решения

Для выполнения поставленных задач было разработано интерактивное приложение с использованием языка программирования Python и набора специализированных библиотек. Пользовательский интерфейс реализован на базе фреймворка Streamlit, позволяет вводить исходные данные и мгновенно получать результаты расчетов и графики. Основные вычисления производятся с помощью библиотеки NumPy, обеспечивающей высокую производительность при работе с массивами данных. Для построения интерактивных графиков используется библиотека Plotly, а для аналитического дифференцирования в третьем задании — библиотека символьных вычислений SymPy.

Реализация Задания №1 (Метод секущих)

Первое задание заключалось в нахождении угла косого скачка уплотнения θc путем решения нелинейного уравнения из методического пособия.

1. Математическая функция. Уравнение (4) было представлено в виде программной функции f_theta_numeric(theta_c, M1, beta, k). Эта функция принимает на вход искомый угол theta_c в радианах (так как все тригонометрические функции в NumPy работают с радианами) и параметры потока. На выходе она возвращает значение невязки, то есть разность между левой и правой частями исходного уравнения.

2. Алгоритм метода секущих. Для численного решения был реализован алгоритм в функции secant_method(f, x0, x1, epsilon, max_iter, *args).

- В качестве входных данных функция принимает: f — ссылка на решаемую функцию (f_theta_numeric); x0 и x1 — два начальных приближения к корню; epsilon — требуемая точность; max_iter — предельное число итераций для предотвращения зацикливания. Параметр *args используется для передачи дополнительных аргументов (M_1, β, k) внутрь f.
- Внутри функции организован итерационный цикл, на каждом шаге которого вычисляется новое приближение по канонической формуле метода секущих:

x_next = x1 - f(x1, *args) * (x1 - x0) / (f(x1, *args) - f(x0, *args))

- Процесс останавливается, когда разница между двумя последовательными приближениями становится меньше заданной точности: abs(x_next - x1) < epsilon.
- Для повышения устойчивости алгоритма добавлена проверка знаменателя в формуле: если он становится пренебрежимо мал, итерации прекращаются во избежание деления на ноль.

3. Расчет параметров. После успешного нахождения корня θc отдельный блок кода производит вычисление всех физических параметров потока за скачком (p_2, T_2, ρ_2, M_2) по стандартным формулам газовой динамики, приведенным в методическом пособии.

Реализация Задания №2 (Метод Ньютона)

Во втором задании решалось трансцендентное уравнение e^x - x - 2 = 0.

1. Определение функций. Были определены две функции: основная функция f(x) и ее первая производная df(x) = e^x - 1. Для их вычисления использовались высокопроизводительные функции из библиотеки NumPy (numpy.exp).

2. Проверка условий сходимости. Перед запуском основного алгоритма программа выполняет проверку условий, гарантирующих сходимость метода на заданном отрезке [a, b]. Проверяется выполнение следствия из теоремы Больцано-Коши (f(a) * f(b) < 0) и знакопостоянство первой и второй производных на этом интервале. Также реализован автоматический выбор начального приближения x_0 из границ отрезка a или b в соответствии с условием $f(x_0) * f''(x_0) > 0$.

3. Алгоритм метода Ньютона. Метод реализован в функции newton_method(f, df, x0, epsilon, max_iter).

- Функция принимает саму функцию f, ее производную df, начальное приближение x0, точность epsilon и максимальное число итераций.
- Итерационный процесс основан на формуле метода касательных:

    x_next = x - f(x) / df(x)

- Критерий остановки аналогичен методу секущих: abs(x_next - x) < epsilon.
- Также предусмотрена защита от деления на ноль, если значение производной df(x) окажется слишком близким к нулю.

Реализация Задания №3 (Сравнительный анализ)

Третье задание требовало наиболее комплексного подхода к реализации.

1. Аналитическое нахождение производной. Применение метода Ньютона к сложному уравнению из первого задания потребовало найти его производную f'(θc). Выполнять это вручную было бы трудоемко и чревато ошибками. Поэтому была использована библиотека символьных вычислений SymPy. Функция f(θc) была задана в символьном виде, после чего с помощью функции diff() была автоматически вычислена ее аналитическая производная.

2. Оптимизация вычислений. Символьное выражение, полученное от SymPy, вычисляется достаточно медленно. Для ускорения расчетов оно было преобразовано в быструю числовую функцию с помощью lambdify. Этот прием позволил совместить точность аналитического дифференцирования со скоростью численных расчетов NumPy.

3. Измерение производительности. Простое измерение времени через time.process_time() давало нулевой результат, так как вычисления происходят слишком быстро. Для получения корректных данных был применен стандартный для Python модуль timeit.

- Ключевой особенностью реализации стало использование lambda-функций для передачи измеряемого кода в timeit.timeit(). Вместо передачи кода строкой (что вызывало ошибку импорта в среде Streamlit), были созданы вызываемые объекты:

secant_callable = lambda: secant_method(...)

newton_callable = lambda: newton_method(...)

- timeit выполняет каждый из этих объектов заданное число раз (num_runs = 10000) и возвращает общее время. Итоговое время расчета для таблицы — это среднее значение, полученное делением общего времени на num_runs. Такой подход обеспечивает высокую точность и стабильность измерений даже для очень быстрых операций.

Проверка сходимости и аналитический вывод производных

Проверка существования и единственности корня

Для Задания №2 (уравнение f(x) = e^x - x - 2 = 0 на отрезке [-2, -1])

а) Существование корня (Следствие теоремы Больцано-Коши)

Теорема гласит, что если непрерывная на отрезке [a, b] функция принимает на его концах значения разных знаков, то внутри этого отрезка существует как минимум один корень.

Функция: f(x) = e^x - x - 2. Данная функция является непрерывной на всей числовой оси как комбинация элементарных непрерывных функций.

Отрезок: [a, b] = [-2, -1].

Значения на концах отрезка:

- f(a) = f(-2) = e^{-2} - (-2) - 2 = e^{-2} ≈ 0.135. Результат положительный.
- f(b) = f(-1) = e^{-1} - (-1) - 2 = e^{-1} - 1 ≈ 0.368 - 1 = -0.632. Результат отрицательный.

Проверка условия: f(a) * f(b) ≈ 0.135 * (-0.632) < 0.

Вывод: Условие Больцано-Коши выполняется, следовательно, на отрезке [-2, -1] гарантированно существует как минимум один корень уравнения.

б) Единственность корня и сходимость метода Ньютона

Для единственности корня и стабильной сходимости метода Ньютона необходимо, чтобы первая (f'(x)) и вторая (f''(x)) производные сохраняли постоянный знак на всем отрезке [a, b].

Первая производная: f'(x) = d/dx (e^x - x - 2) = e^x - 1.

Вторая производная: f''(x) = d/dx (e^x - 1) = e^x.

Анализ знака f'(x) на отрезке [-2, -1]:

- Функция g(x) = e^x является монотонно возрастающей. Следовательно, на отрезке [-2, -1] ее максимальное значение достигается при x = -1: e^{-1} ≈ 0.368.

- Поскольку максимальное значение e^x на отрезке меньше 1, то разность e^x - 1 будет отрицательной на всем отрезке.

- f'(x) = e^x - 1 < 0 для всех x ∈ [-2, -1].

Вывод: Первая производная сохраняет знак (всегда отрицательна). Это означает, что функция f(x) является строго монотонной на данном отрезке, что, в свою очередь, гарантирует единственность корня.

Анализ знака f''(x) на отрезке [-2, -1]:

- Функция f''(x) = e^x является показательной и принимает строго положительные значения для любого действительного x.

- f''(x) = e^x > 0 для всех x ∈ [-2, -1].

Вывод: Вторая производная также сохраняет постоянный знак на отрезке.

Итоговый вывод для Задания №2: Все условия для успешного применения метода Ньютона выполнены. На отрезке [-2, -1] существует единственный корень, и метод гарантированно сойдется к нему при правильном выборе начального приближения.

Для Задания №3 (уравнение из задачи газовой динамики)

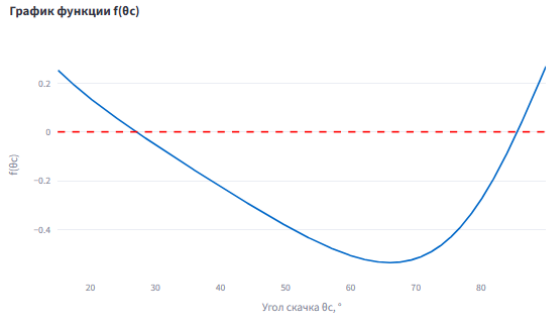
Проверка условий существования и единственности корня была реализована в программе графическим методом.

1. Существование корня: при построении графика функции f(θc) (реализовано в task1.py и task3.py) на физически осмысленном интервале (β, π/2) было визуально установлено, что график пересекает ось абсцисс. Это является наглядным подтверждением выполнения условия Больцано-Коши на некотором отрезке, содержащем корень.

2. Единственность корня: известно, что уравнение θ-β-M может иметь два решения: "слабый" (меньший θc) и "сильный" (больший θc) скачок уплотнения. Физически устойчивым является только слабый скачок. Графический анализ позволил:

- Визуально подтвердить, что в области, соответствующей слабому скачку, функция ведет себя монотонно, что говорит о локальной единственности корня.

- Выбрать начальные приближения для численных методов таким образом, чтобы обеспечить сходимость именно к физически корректному, меньшему корню.



Аналитические формулы производных для Задания №2

При решении трансцендентного уравнения e^x - x - 2 = 0 методом Ньютона использовались следующие аналитически выведенные производные:

Исходная функция:
f(x) = e^x - x - 2

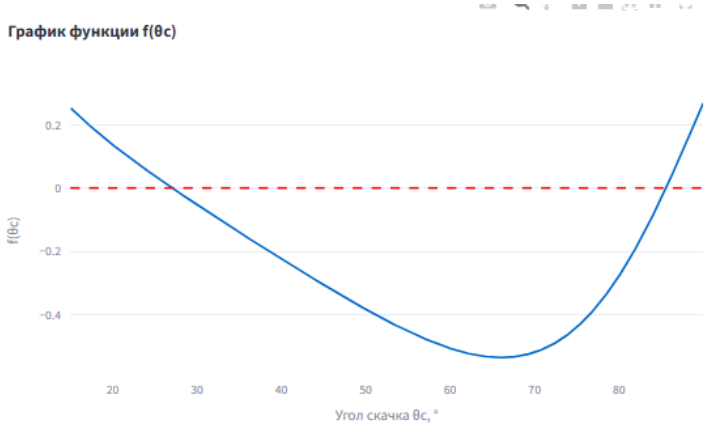
Первая производная (необходима для итерационной формулы Ньютона):
f'(x) = d/dx (e^x - x - 2) = d/dx(e^x) - d/dx(x) - d/dx(2) = e^x - 1 - 0
f'(x) = e^x - 1

Вторая производная (необходима для проверки условия сходимости $f(x_0) * f''(x_0) > 0$):
f''(x) = d/dx (e^x - 1) = d/dx(e^x) - d/dx(1) = e^x - 0
f''(x) = e^x

Результаты

Для задания 1

Графики:



В первом задании методом секущих был определен угол наклона косого скачка уплотнения θc и последующие параметры газового потока. При заданной точности ε = 10^{-9} метод сошелся за **5 итераций**.

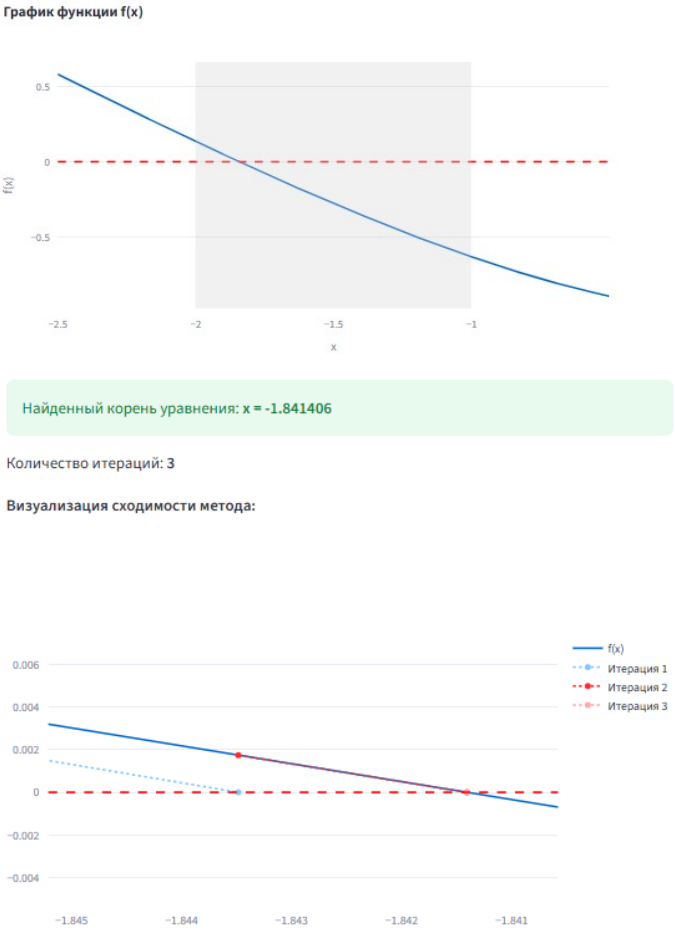
Найденное значение угла составило **θc ≈ 27.06°** (0.4723 рад).

Используя найденный угол `θc`, рассчитаем остальные параметры:

	Параметр	До скачка (индекс 1)	После скачка (индекс 2)
0	Давление, p	165000 Па	610000.0 Па
1	Температура, T	300 K	464.0 K
2	Плотность, ρ	1.92 кг/м³	4.58 кг/м³
3	Число Маха, M	4.0	2.93

Для задания 2

Графики:



Во втором задании с помощью метода Ньютона находился корень уравнения e^x - x - 2 = 0 на отрезке [-2, -1]. Программа автоматически выбрала начальное приближение x_0 = -2.0, так как для этой точки выполнялось условие сходимости $f(x_0) * f''(x_0) > 0$.

Для достижения заданной точности методу потребовалось всего **3 итерации**.

Найденный корень уравнения: **x ≈ -1.841406**.

Высокая скорость сходимости (малое число итераций) полностью соответствует теоретическим ожиданиям от метода Ньютона, обладающего квадратичной скоростью сходимости на отрезке, где выполнены все необходимые условия.

Для задания 3

В третьем задании проводилось прямое сравнение эффективности метода секущих и метода Ньютона на примере задачи о косом скачке уплотнения. Оба метода были запущены с одинаковой высокой точностью ($\varepsilon = 10^{-9}$) для поиска одного и того же корня.

Замеры времени производились как среднее по **10000** запускам каждого метода.

	Параметр	Метод Секущих	Метод Ньютона
0	Метод	Секущих	Ньютона
1	Найденный корень, °	27.06288	27.06288
2	Количество итераций	6	4
3	Среднее время расчета, сек	8.97244e-05	5.96010e-05

Выводы

В ходе выполнения данной лабораторной работы были успешно решены две задачи на нахождение корней нелинейных уравнений с использованием численных методов, реализованных в виде интерактивного программного приложения.

Были изучены и применены на практике метод секущих и метод Ньютона. Оба метода продемонстрировали свою работоспособность, с высокой точностью нашли решения как для задачи из области газовой динамики, так и для трансцендентного математического уравнения.

На примере Задания №2 была проведена аналитическая проверка условий существования и единственности корня, а также сходимости метода Ньютона, подтвердив теоретическую обоснованность его применения.

Сравнительный анализ (Задание №3) наглядно показал преимущество метода Ньютона над методом секущих для рассматриваемой задачи. Метод Ньютона сошелся к решению за меньшее количество итераций (4 против 6) и показал лучшее среднее время выполнения.