

Лабораторная работа №1

Численные методы решения нелинейных уравнений

Выполнил: Меркулов Роман Александрович

Группа: 6

Цель работы

Изучение численных методов решения нелинейных уравнений (метод бисекции, метод хорд, метод Ньютона), их реализация в Jupyter Notebook и сравнительный анализ скорости сходимости на примере функции $f(x) = x^3 - 6x^2 + 5x$.

Ход работы

№ этапа	Описание этапа	Результат этапа
1	Подготовка. Ознакомление с лабораторной работой, ходом решения. Просмотр конспекта в тетради. Ознакомление с Jupyter Notebook и краткое ознакомление с matplotlib.	Составлен план работы, начата сама работа в Jupyter Notebook, попробовал построить минимальный график.
2	Выполнение первого и второго этапа.	Визуализирована функция $f(x) = x^3 - 6x^2 + 5x$ на интервале $[-1, 6]$, определение корней ($x = 0, 1, 5$) и интервалов локализации
3	Выполнение 3 этапа. Использование методических рекомендаций для написания алгоритмов	Реализованы 3 численных метода: бисекции, хорд и Ньютона
4	Визуализация графиков в линейной и логарифмической шкале функции ошибки 3 методов от кол-ва итераций	Построены сравнительные графики 3 численных методов, демонстрирующие скорость сходимости каждого из них
5	Анализ полученных графиков	Сделаны выводы о скорости и надежности каждого метода

Выводы

1. Метод Ньютона показал наилучшую скорость сходимости на всех исследованных интервалах.
2. Метод хорд и метод бисекции показали сопоставимые результаты, при этом метод хорд часто сходился быстрее на начальных итерациях.
3. Метод Ньютона стабильно работал на любом интервале, содержащем один корень, в то время как методы хорд и бисекции требовали более тщательного подбора интервалов.
4. Логарифмическая шкала графиков позволила наглядно увидеть разницу в скорости сходимости методов, особенно на поздних итерациях.

5. Все три метода успешно реализованы и могут использоваться для решения нелинейных уравнений с заданной точностью.

Импортируемые функции сторонних библиотек

Библиотека	Функция	Назначение	Основные параметры
matplotlib	plt.figure()	Создание нового рисунка	figsize - размер рисунка
matplotlib	plt.plot()	Построение графика	массив значений x, y, color - цвет, linewidth - толщина, label - подпись
matplotlib	plt.semilogy()	График с логарифмической шкалой по оси Y	массив значений x, y, color, linewidth, label
matplotlib	plt.axhline()	Горизонтальная линия	color - цвет линии
matplotlib	plt.axvline()	Вертикальная линия	color - цвет линии
matplotlib	plt.title()	Заголовок графика	сам текст, fontsize, fontweight
matplotlib	plt.xlabel()	Подпись оси X	сам текст, fontsize
matplotlib	plt.ylabel()	Подпись оси Y	сам текст, fontsize
matplotlib	plt.grid()	Сетка на графике	Visible – видимость, alpha - прозрачность
matplotlib	plt.legend()	Легенда	fontsize
matplotlib	plt.xlim()	Границы по оси X	минимальное и максимальное значение
matplotlib	plt.ylim()	Границы по оси Y	минимальное и максимальное значение
matplotlib	plt.tight_layout()	Автоматическая подгонка элементов	—
matplotlib	plt.show()	Отображение графика	—
numpy	np.linspace()	Создание равномерно распределенных точек	start - начало, stop - конец, num - количество
numpy	np.abs()	Модуль числа	—