Санкт-Петербургский политехнический университет

Высшая школа прикладной математики и вычислительной физики, ИПММ

Направление подготовки

«01.03.02 Прикладная математика и информатика»

Отчет по лабораторной работе №**4**

**тема "Численное интегрирование квадратурными формулами типа Гаусса и смешанного типа"**

**дисциплина "Численные методы"**

Выполнил студент гр. **3630102/80001** Д.В. Хрипунков

Преподаватель: С.Б. Добрецова

Санкт-Петербург

**2020**

1. Формулировка задачи

Для алгебраической функции необходимо найти определённый интеграл на заданном промежутке с использованием формулы Радо с тремя узлами с заданной точность

По результатам решения найти и проанализировать зависимости:

1. Погрешность , где - точное значение определённого интеграла, – значение интеграла, вычисленное по методу, а – точность вычисления
2. , где – число разбиений отрезка для заданной точности
3. Взаимное расположение фактической точности вычислений и порядка теоретической погрешности

По зависимостям построить графики в MATLAB и установить закономерности

2. Постановка задачи

3. Алгоритм метода

Первым делом, проведём замену и тогда интеграл принимает вид

Вычисления по квадратурной формуле так же преобразуются:

Для трёх узлов будем начинать нумерацию с нуля, тогда будем использовать значение

Значения узлов будем получать как корни полинома , полином Лежандра

Для

Квадратурная формула будет иметь 5 параметров, тогда и

*; ; ; ;*

;

Из решения этой системы уравнений получается:

Итоговая формула для подсчёта значения интеграла по формуле Радо с тремя узлами:

Для вычислений с заданной точностью используется правило Рунге, вычисления с уменьшением мелкости разбиения продолжаются пока , а для формулы Радо с 3 узлами берётся . Разбиение уменьшается в 2 раза на каждой итерации, число текущее количество разбиений.

4. Анализ задачи

В качестве тестовой алгебраической функции используем . Функция непрерывна на всём промежутке и дифференцируема на нём.

При таких условиях точное значение интеграла

5. Тестовый пример

Возьмём .

Точное значение интеграла

Посчитаем интеграл, разбив отрезок на 2 части:

1. На отрезке [1, 1.5]:
2. На отрезке [1.5, 2]:

Фактическая точность:

6. Контрольные тесты

Весь анализирующий функционал реализован на языке c++ в Visual Studio 2019, а графическая интерпретация в пакете MATLAB.

Берём алгебраическую функцию

Вычисления проводятся по формуле с 3 узлами, поэтому используем значения:

Точность до 30 знаков после запятой для данных значений нужна для более высокой точности вычислений. На практике установлено, что тесты заметно менее точны при меньшем числе цифр в дробной части

Значение точности вычислений будет последовательно изменяться от с шагом в

Точное значение интеграла используем

7. Модульная структура программы

double CountFunc(double x) – функция вычисления

double CountIntegral() – функция вычисления интеграла на заданном отрезке, принимает в качестве глобальных параметров границы текущего отрезка, а также число N = 2, внутри неё задаются массивы значений и , возвращает значение интеграла на отрезке

8. Численный анализ

Графики зависимостей, построенные в MATLAB:

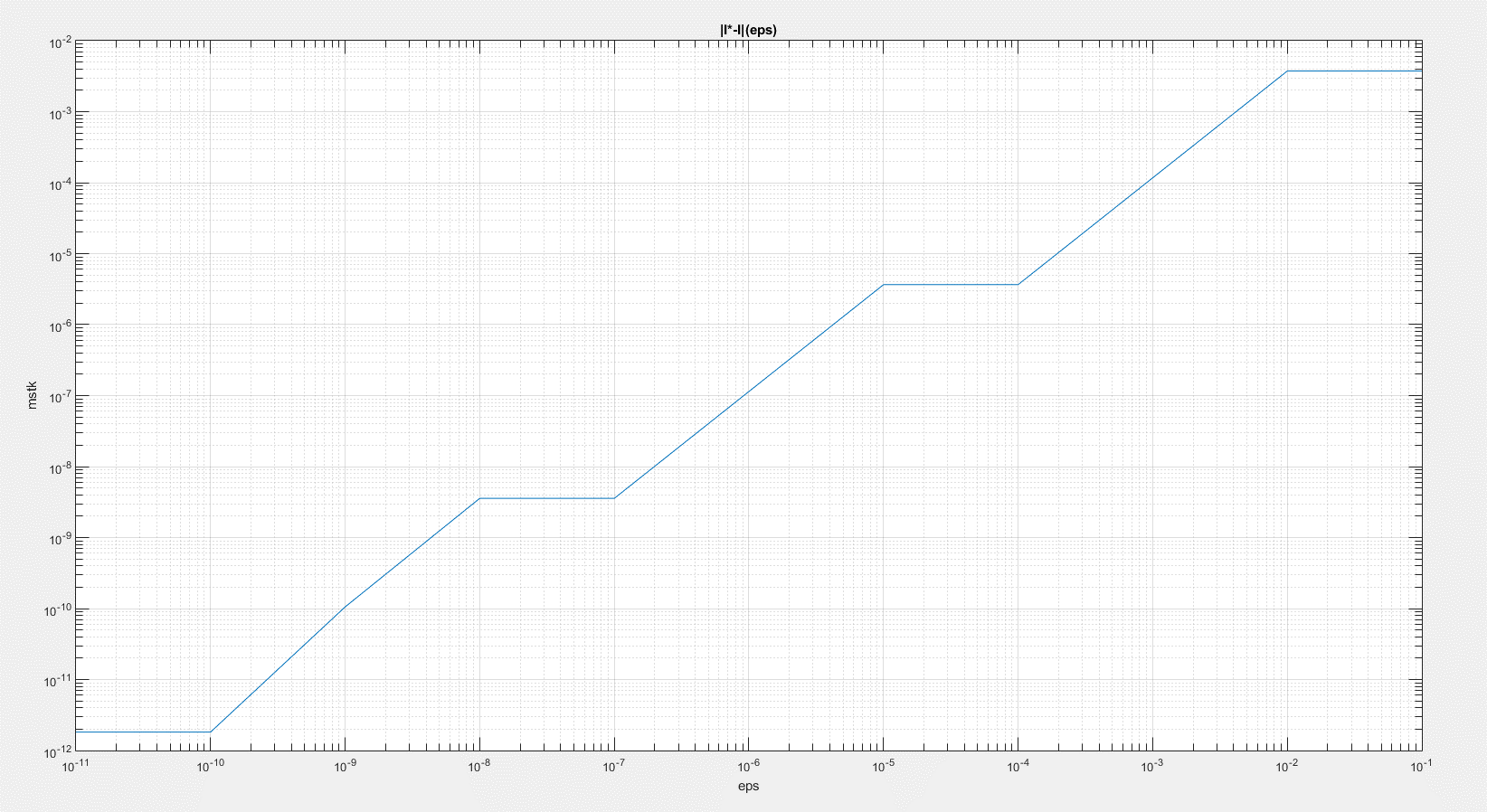
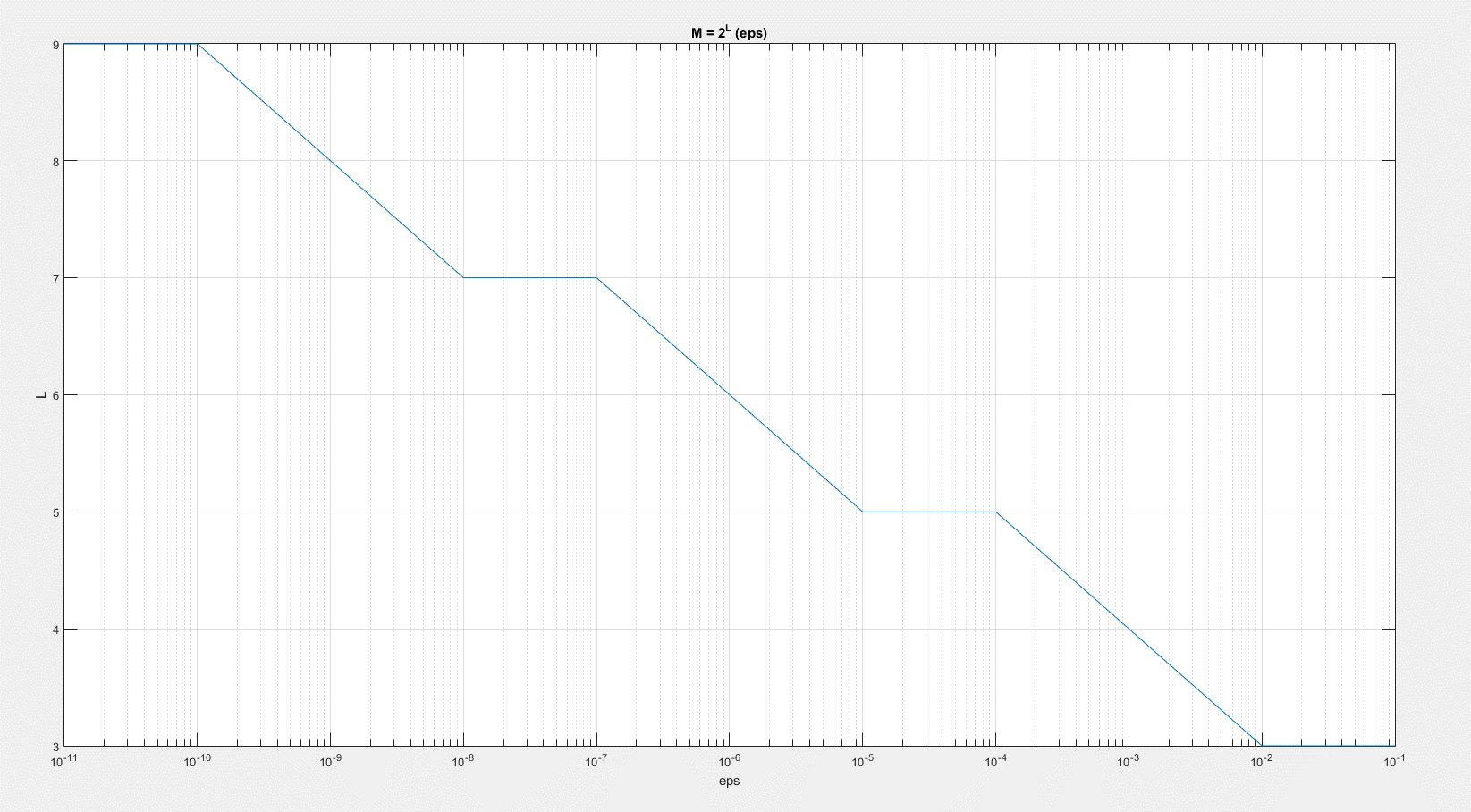
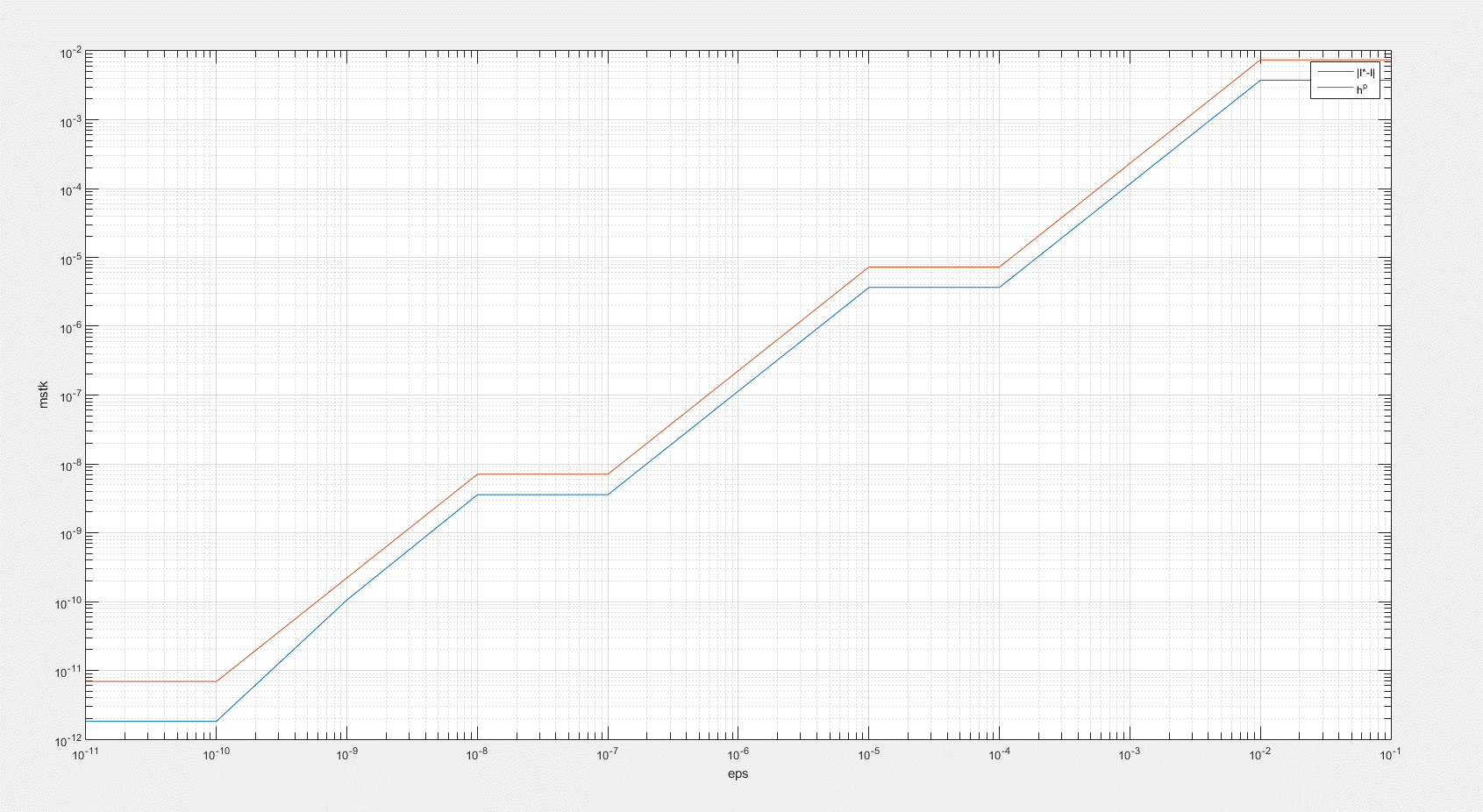


График демонстрирует, что фактическая точность вычислений ступенчато увеличивается при увеличении задаваемой точности. Плоские участки обусловлены вторым графиком, который показывает, что для некоторых вычислений необходимо одно и то же число разбиений отрезка, а значит и одна и та же фактическая точность. Так же стоит отметить, что фактическая точность часто примерно на порядок лучше, чем задаваемая



L(ε)

Данный график подтверждает наблюдения с первого графика. Он демонстрирует, что для большей точности нужно большее число разбиений, а так же, что для достижения некоторых точностей нужно одинаковое число разбиений



Фактическая и теоретическая точность

Данный график сравнивает теоретическую и фактическую точность. Он показывает, что две линии идут полностью параллельно, и фактическая точность даже чуть лучше приблизительной теоретической

9. Вывод

Формула Радо с 3 узлами показывает хорошие результаты по фактической точности, всегда превосходя ожидаемое значение. В это же время число разбиений остаётся не очень большим даже для высокой точности.

Тем не менее, данный метод требует достаточно громоздких предварительных вычислений и преобразований, поэтому сложнее и дольше в реализации, чем, например, методы 3 работы.

Однако, эта сложность окупается более высокой точностью, как и меньшей необходимостью в количестве разбиений. Таким образом, более высокая сложность метода обеспечивает его большую надёжность.