

Содержание

Постановка задачи	2
Математическое обоснование	3
Результаты экспериментов	4
Структура программы и спецификация функций	5
Сборка программы (Make-файл)	7
Отладка программы, тестирование функций	8
Ключ <i>-test</i>	8
Программа на Си и на Ассемблере	9
Анализ допущенных ошибок	10
Список цитируемой литературы	11

Постановка задачи

С заданной точностью $\varepsilon = 0.001$ вычислить площадь плоской фигуры, ограниченной тремя кривыми, заданными уравнениями $f_1 = \exp(x) + 2$, $f_2 = -2x + 8$, $f_3 = -\frac{5}{x}$. Для этого используется метод трапеций. Для поиска вершин фигуры используется метод хорд. Отрезки для применения этого метода вычислены аналитически. Также нужно подобрать значения точности ε_1 и ε_2 в данных численных методах, чтобы площадь вычислялась с уже заданной точностью ε

Математическое обоснование

Условие сходимости метода хорд: $f(x)$ имеет на сегменте $[a, b]$ монотонную и непрерывную производную, сохраняющую определенный знак[1]. $(f_1 - f_3)' = \exp(x) - 5/x^2$ - удовлетворяет критерию на $[-3; -2]$, $(f_2 - f_3)'' = -5/x^2$ - удовлетворяет критерию на $[-1; -0.5]$, $(f_1 - f_2)' = \exp(x)$ удовлетворяет критерию на $[1; 2]$.

Условие сходимости метода трапеций: $f(x)$ имеет на рассматриваемом сегменте непрерывную вторую производную[1]. В области определения все три функции имеют непрерывную вторую производную, поэтому метод применим. Точность вычисления определенного интеграла обеспечивается использованием правила Рунге[2].

При подсчете интеграла и нахождения корня использовались $\varepsilon_1 = \varepsilon_2 = 10^{-5}$

При нахождении точек пересечения функций мы теряем в точности интегрирования методом прямоугольников не более $\varepsilon_1 \cdot Fabsmax$, где $Fabsmax = \max(|F(a)|, |F(b)|)$, где a, b - границы отрезка, на котором ищется корень, $F(x)$ - функция, у которой ищется корень (если функции монотонны). Такое выражение для потери точности интегрирования характеризуется тем, что на одном из этапов мы оцениваем площадь разбиения сверху площадью прямоугольника со сторонами ε_1 и $Fabsmax$. При подсчёте каждого интеграла, потеря точности по оценке сверху составляет не более чем $\varepsilon_1 \cdot Fabsmax_{ij} + \varepsilon_1 \cdot Fabsmax_{kl} + \varepsilon_2$ (здесь считается потеря с двух концов отрезка интегрирования и сама погрешность интегрирования). Значит, итоговая оценка сверху: $2\varepsilon_1(Fabsmax_{12} + Fabsmax_{13} + Fabsmax_{23}) + 3\varepsilon_2$. $Fabsmax_{ij} = \max(|f_i(a) - f_j(a)|, |f_i(b) - f_j(b)|)$.

Обозначим $F_{13} = f_1 - f_3$, $F_{12} = f_1 - f_2$, $F_{23} = f_2 - f_3$. Эти функции являются монотонными на заданных отрезках. Отсюда находим

$$Fabsmax_{13} = F_{13}(-3)$$

$$Fabsmax_{12} = F_{12}(2)$$

$$Fabsmax_{23} = F_{23}(-1)$$

Вычисление площади с заданной точностью ε удастся, если оценка погрешности сверху меньше ε , значит

$$2\varepsilon_1(Fabsmax_{12} + Fabsmax_{13} + Fabsmax_{23}) + 3\varepsilon_2 < \varepsilon$$

Отсюда при $\varepsilon_1 = \varepsilon_2 = 10^{-5}$

$$2\varepsilon_1(Fabsmax_{12} + Fabsmax_{13} + Fabsmax_{23}) + 3\varepsilon_2 < 0.0003 < 0.001$$

Значит, данные значения точности подходят для вычисления площади

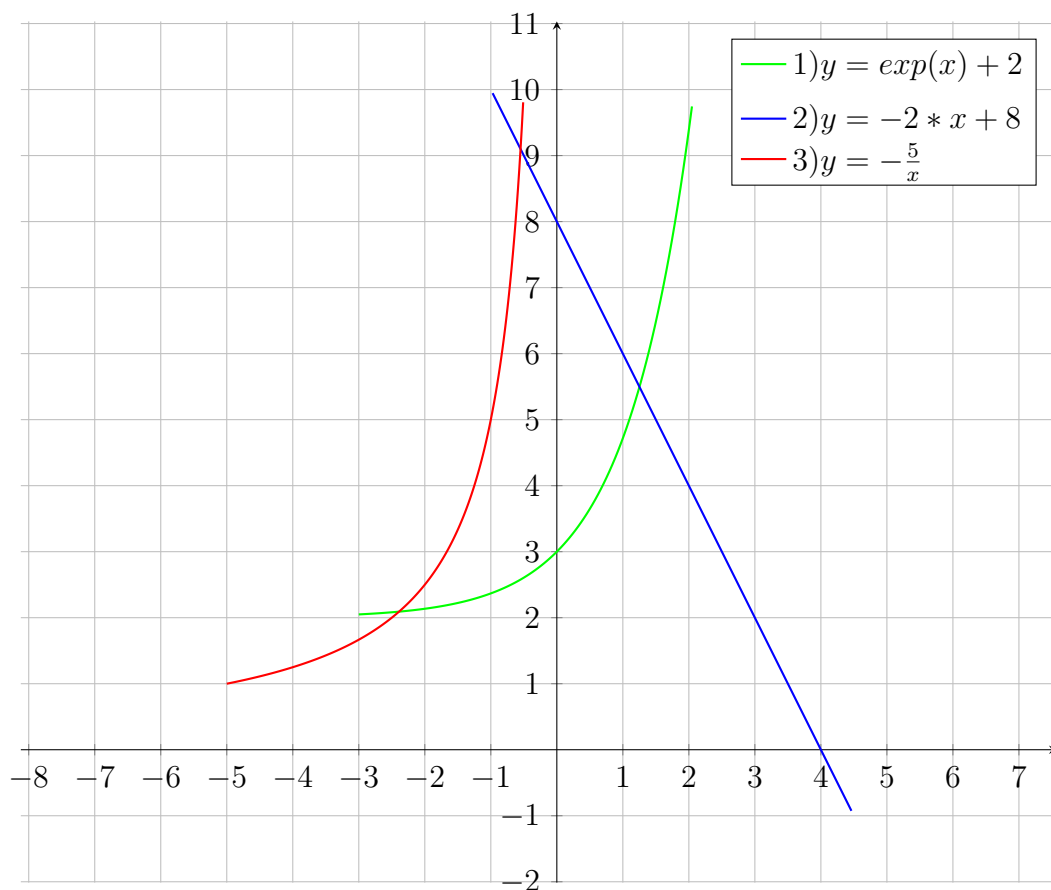


Рис. 1: Плоская фигура, ограниченная графиками заданных уравнений

Результаты экспериментов

Кривые	x	y
1 и 2	-0.54951	9.099
2 и 3	1.251757	5.496
1 и 3	-2.390541	2.092

Таблица 1: Координаты точек пересечения

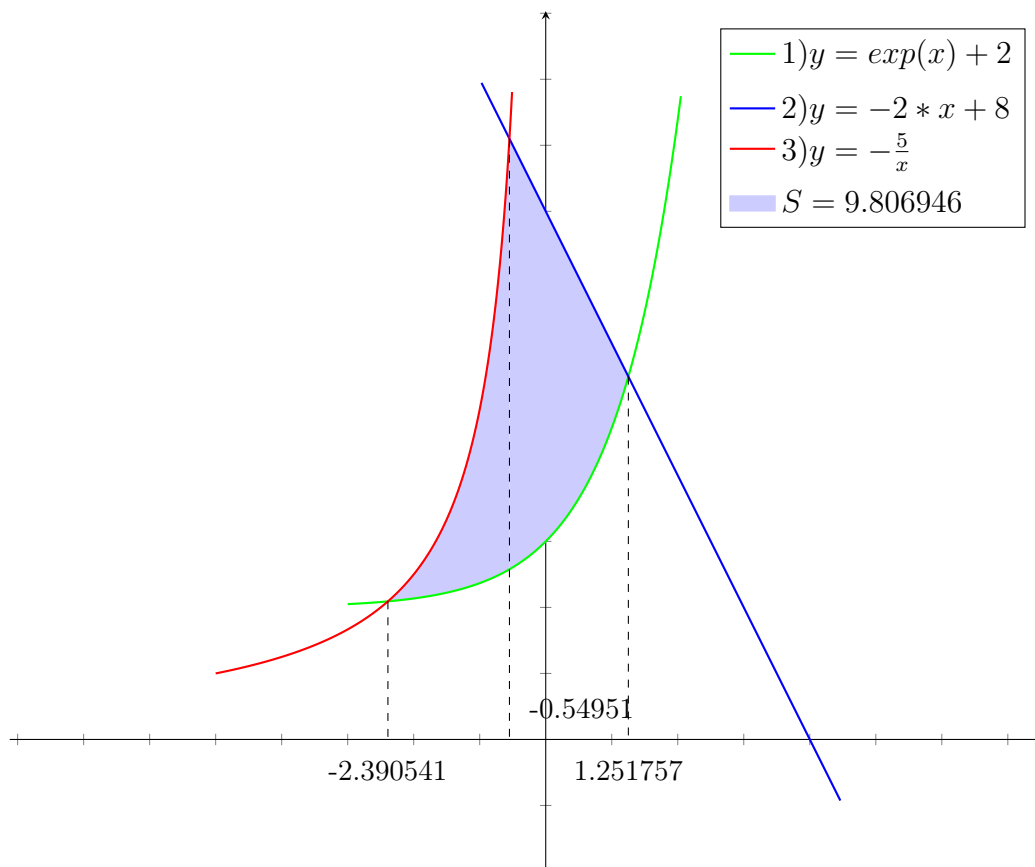


Рис. 2: Плоская фигура, ограниченная графиками заданных уравнений

Структура программы и спецификация функций

Программа состоит из двух модулей: `main.c` и `s.asm`.

В ассемблерном файле объявлены функции в соглашении `cdecl`, возвращающие значения математических функций, ограничивающих фигуру. Они используются в `main.c`

В `main.c` происходит поиск абсцисс точек пересечения функций с заданной точностью, а также подсчет определенного интеграла. Программа поддерживает следующие ключи командной строки:

- `-help` - Выводит список доступных флагов.
- `-intersections` - Выводит абсциссы точек пересечения тестируемых функций.
- `-iterations` - Выводит количество итераций, потребовавшихся для нахождения абсцисс точек пересечения тестируемых функций.
- `-root_test` - Позволяет провести тестирование функции поиска абсцисс точек пересечения функций (функции, отрезок для поиска и точность вычисления выбираются пользователем).

- `-integral_test` - Позволяет провести тестирование функции расчета определенного интеграла (функция, отрезок для поиска и точность вычисления выбираются пользователем).
- `-test` - Тестирование функций `root` и `integral` на удобных для аналитического вычисления функциях

Содержит в себе следующие функции:

- `double root(double(*f)(double), double(*g)(double), double a, double b, double eps, int * iterations)`
Поиск абсциссы пересечения функций `f` и `g` на заданном промежутке с заданной точностью с помощью метода хорд
- `double integral(double(*f)(double), double a, double b, double eps)`
Данная функция считает определенный интеграл методом трапеций на заданном промежутке с заданной точностью.
- `f1, f2, f3, fln, fsin, f0, fxx`
Принимают на вход значение `double` и выводят соответствующее значение математической функции.
- `int icheck(char *), int fcheck(char *)`
Проверка пользовательского ввода на корректность.
- `double check(double res, double ans, double eps)`
Функции для отладки программы: сравнивает вывод программы и аналитический ответ

Сборка программы (Make-файл)

```
all: main.o s.o
gcc -m32 -o program main.o s.o -lm -Wall -Wextra

main.o: main.c
gcc -m32 -c -o main.o main.c

s.o: s.asm
nasm -f elf32 -o s.o s.asm

clean:
rm *.o
```

Makefile собирает все модули ключом -all. Конечный файл program зависит от объектных файлов main.o и s.o, они соответственно от main.c и s.asm. Ключ clean удаляет объектные файлы

Отладка программы, тестирование функций

Тестирование численных методов проводилось с помощью ключей командной строки, которые запускали функции численных методов с заданными параметрами.

Ключ $-test$

По этому ключу программа выводит результат работы функции на модельных примерах, где определенный интеграл и корни вычислимы аналитически

№	Функция
1	$\ln(x) - 1$
2	$x^4 + x^2 - 2$
3	$\sin(x)$

Результаты работы программы с ключом $-test$:

№	Функция	Отрезок	Корень	Точность	Аналитическое решение
1	$\ln(x) - 1$	$[1, 3]$	2.718282	0.00001	e (2.7182818...)
2	$x^4 + x^2 - 2$	$[0, 2]$	0.999997	0.00001	1
3	$\sin(x)$	$[2, 4]$	3.141590	0.001	π (3.1415926...)

№	Функция	Первообразная	Отрезок	Вывод	Точность	Ответ
1	$x^4 + x^2 - 2$	$\frac{x^5}{5} + \frac{x^3}{3}$	$[0; 2]$	-2.933330	0.00001	-2.9(3)
2	$\sin(x)$	$\cos(x)$	$[0; \pi]$	1.999994	0.00001	2

Также для отладки программы существуют ключи $-root_{test}$ и $-integral_{test}$, с помощью которых пользователь может сам выбирать функции и отрезки для тестирования, вводя их в командную строку.

Программа на Си и на Ассемблере

Исходные файлы программы прилагаются в архиве. Программа собирается с помощью цели `all`, запуск программы осуществляется с помощью команды `./program`. Для ознакомления со списком ключей существует ключ `'-help'`

Анализ допущенных ошибок

В ходе разработки программы был допущен ряд ошибок и неточностей:

- Функция *integral* работала крайне неэффективно из-за перевычисления уже посчитанных значений, а также медленного роста числа разбиений отрезка.
- Была добавлена встроенная проверка корректности полученного результата для отладки программы.
- Компиляция программы с флагами *-Wall -Wextra* выдавала предупреждения. Для исправления были исправлены синтаксические неточности.
- Пользовательский ввод не проверялся на корректность.
- В коде программы были лишние пробелы в концах строк.
- Аргументы в ключах *-integral_test*, *-root_test* передавались в самой программе, а не в командной строке.

Список литературы

- [1] Ильин В. А., Садовничий В. А., Сендов Бл. Х. Математический анализ. Ч. 1 — Москва: Издательство Проспект, 2004.
- [2] А.А. Федотов, П.В. Храпов. Численные методы интегрирования, решения дифференциальных уравнений и задач оптимизации. Москва: Издательство МГТУ им. Н.Э.Баумана, 2015