Министерство образования и науки РФ
ФГБПОУ ВПО Тульский государственный университитет
КАФЕДРА АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ

ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

Лабораторная работа № 4
по курсу «Вычислительный практикум»

Вариант № 4

Выполнил:	студент группы 220601	________ (подпись)	Белым А.А.
Проверил:	к. т. н., доцент	________ (подпись)	Карцева А.С.

Тула 2011

Цель работы

Цель работы заключается в том, чтобы изучить различные методы численного интегрирования и написать программу, реализующий один из таких методов.

Задание на работу

Вычислить значение интеграла методом средних прямоугольников:

$$\int_a^b \frac{x}{1+x^2}dx$$

Теоретическая справка

Пусть $f(x)$ - функция, непрерывная и достаточно гладкая на отрезке $[a, b]$.

Требуется найти интеграл $I = \int_a^b f(x)dx$.

Численные методы обычно применяются при вычислении интегралов, которые не берутся, или имеют сложные подынтегральные функции.

Все численные методы строятся на том, что подынтегральная функция приближенно заменяется более простой (горизонтальной или наклонной прямой, параболой 2-го, 3-го или более высокого порядка), от которой интеграл легко вычислить. В результате получаются формулы интегрирования, называемые квадратурными, в виде взвешенной суммы ординат подынтегральной функции в отдельных точках:

$\int_a^b f(x)dx \approx \sum_i \omega_i f(x_i)$

Чем меньше интервалы, на которых производят замену, тем точнее вычисляется интеграл. Поэтому исходный отрезок $[, b]$ для повышения точности делят на несколько равных или неравных интервалов, на каждом из которых применяют формулу интегрирования, а затем складывают результаты.

Все методы различаются значениями ординат x_i и весов ω_i . В большинстве случаев погрешность численного интегрирования определяется путем двойного интегрирования: с исходным шагом (шаг определяется путем равномерного деления отрезка $b - a$ на число отрезков n:$h = (b - a)/n$ и с шагом, увеличенным в 2 раза. Разница вычисленных значений интегралов определяет погрешность.

Сравнение эффективности различных методов проводится по степени полинома, который данным методом интегрируется точно, без ошибки. Чем выше степень такого полинома, тем выше точность метода, тем он эффективнее.

К простейшим методам можно отнести методы прямоугольников (левых, правых и средних) и трапеций. В первом случае подынтегральная функция заменяется горизонтальной прямой ($= {}_0$) со значением ординаты (т.е. значением функции) соответственно слева, справа или посередине участка, во втором случае — наклонной прямой ($= {}_1 + {}_0$). Формулы интегрирования при разбиении отрезка $[a, b]$ на n частей с равномерным шагом h соответственно приобретают вид:

● для одного участка интегрирования (простые формулы):

- метод прямоугольников

$I = \int_a^b f(x)dx \approx f(a)h,$

$I = \int_a^b f(x)dx \approx f(a + h/2)h,$

$I = \int_a^b f(x)dx \approx f(b)h$,

$\omega_0 = h = b - a$;

- метод трапеций $I = \int_a^b f(x)dx \approx \frac{f(a)+f(b)}{h}$,

$\omega_0 = \frac{b-a}{2},$

● для n участков интегрирования (составные формулы):

- метод прямоугольников $I = \int_a^b f(x)dx \approx \sum_{i=0} n - 1\omega_i f(x_i),$

$I = \int_a^b f(x)dx \approx \sum_{i=0} n - 1\omega_i f(x_i + h/2),$

$I = \int_a^b f(x)dx \approx \sum_{i=1} n\omega_i f(x_i),$

$\omega_i = h = const.$

- метод трапеций

$I = \int_a^b f(x)dx \approx \sum_{i=1} n\omega_i f(x_i),$

$\omega_i = h/2 = const.$

Нетрудно заметить, что в методе прямоугольников интеграл вычислится абсолютно точно только при $f(x) = c$, а в методе трапеций — при $f(x)$ линейной или кусочно-линейной.

Метод прямоугольников не находит практического применения в силу значительных погрешностей

Схема алгоритма

На рисунке 1 представлена схема алгоритма расчета определенного интеграла методом средних прямоугольников до достижения указанной точности.

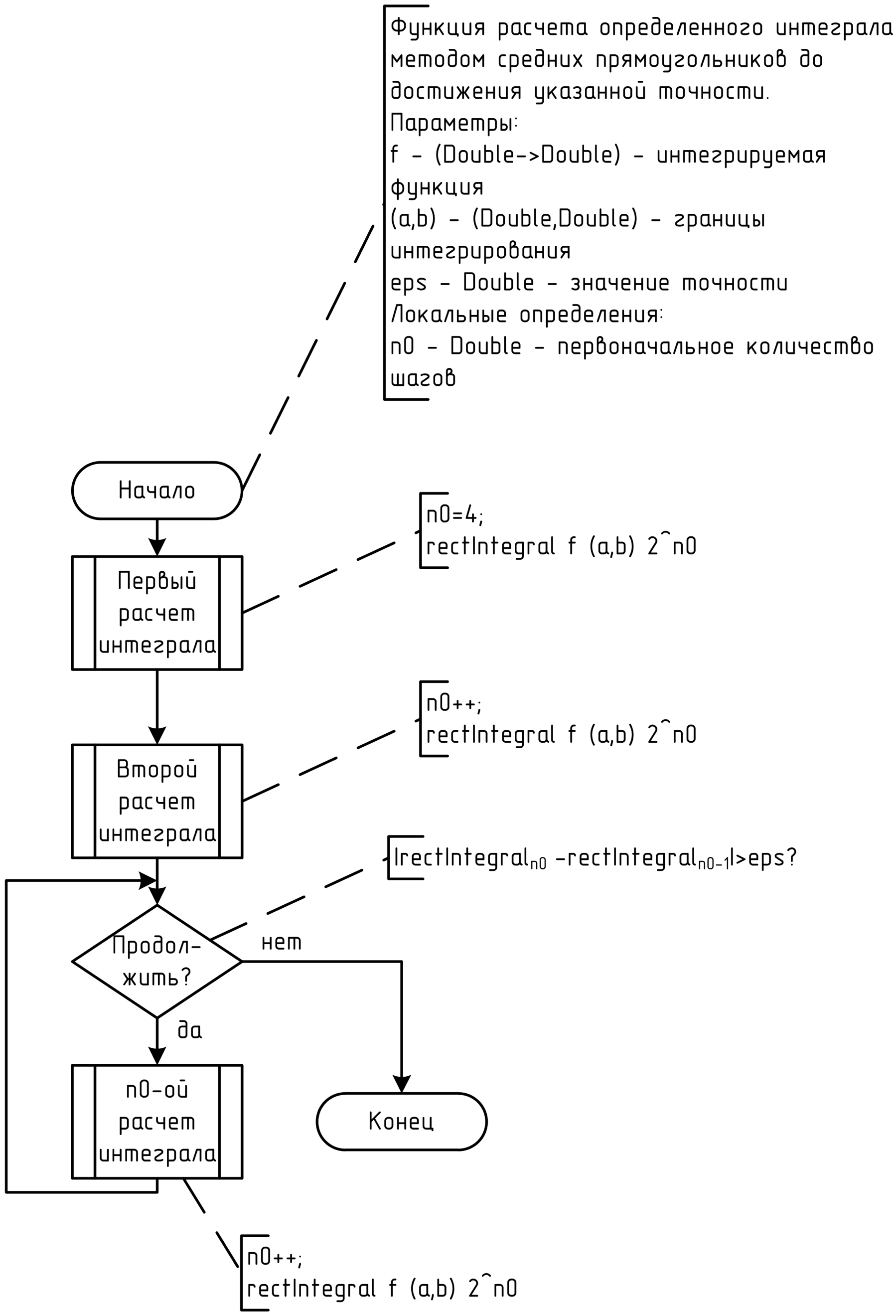

Рисунок 1 - Схема алгоритма расчета интеграла до достижения точности

На рисунке 2 представлена схема алгоритма расчета определенного интеграла методом средних прямоугольников при указанном количестве шагов.

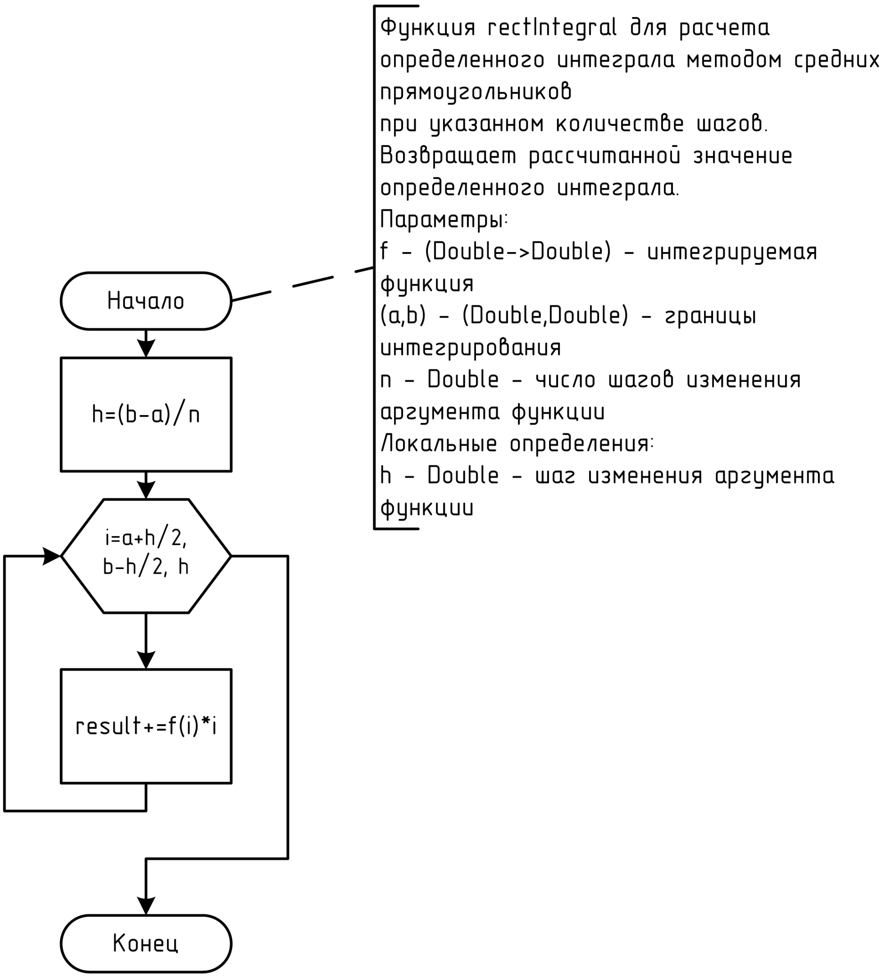

Рисунок 2 - Схема алгоритма расчета интеграла при указанном количестве шагов

Инструкция пользователя

Программа позволяет провести расчет определенного интеграла методом средних прямоугольников до достижения указанной точности.

Для работы введите параметры расчета (границы интервала интегрирования и требуюмую точность) в соотвествующие поля ввода на форме и нажмите кнопку "Посчитать". Программа выведет значение определенного интеграла на форму.

Инструкция программиста

При разработке программы численного интегрирования методом средних прямоугольников были написаны следующие функции:

1. rectIntegral - расчет определенного интеграла методом средних прямоугольников при указанном количестве шагов.

Возвращает рассчитанное значение определенного интеграла.

rectIntegral::(Double->Double)->(Double,Double)->Double->Double

Параметры функции представлены в таблице 1 :

Таблица 1 - Параметры функции расчета определенного интеграла при указанном количестве шагов

имя	тип	предназначение
f	(Double->Double)	интегрируемая функция
(a,b)	(Double,Double)	границы интегрирования
n	Double	число шагов изменения аргумента функции

Локальные определения функции представлены в таблице 2 :

Таблица 2 - Локальные определения функции расчета определенного интеграла при указанном количестве шагов

имя	тип	предназначение
h	Double	шаг изменения аргумента функции

2. firstWhenEps - возвращает первый элемент списка, для которого выполнено условие точности.

firstWhenEps::[Double]->Double->Double

Параметры функции представлены в таблице 3 :

Таблица 3 - Параметры функции выбора элемента списка

имя	**тип**	**предназначение**
xs	[Double]	список приближений
eps	Double	значение точности

3. rectIntegrate - расчет определенного интеграла методом средних прямоугольников до достижения указанной точности.

rectIntegrate::(Double->Double)->(Double,Double)->Double->Double

Параметры функции представлены в таблице 4 :

Таблица 4 - Параметры функции расчета определенного интеграла до достижения точности

имя	**тип**	**предназначение**
f	(Double->Double)	интегрируемая функция
(a,b)	(Double,Double)	границы интегрирования
eps	Double	значение точности

Локальные определения функции представлены в таблице 5 :

Таблица 5 - Локальные определения функции расчета определенного интеграла до достижения точности

имя	**тип**	**предназначение**
n0	Double	первоначальное количество шагов

Текст программы

Реализиция задачи численного интегрирования методом средних прямоугольников написана на языке Haskell 98 и состоит из двух частей.

Первая часть является вычислительным ядром программы, текст этой части приводится ниже:

```
module MyIntegral where

-- rectIntegral - расчет определенного интеграла методом средних прямоугольников
-- при указанном количестве шагов.
-- Возвращает рассчитанной значение определенного интеграла.
-- Параметры:
-- f - (Double->Double) - интегрируемая функция
-- (a,b) - (Double,Double) - границы интегрирования
-- n - Double - число шагов изменения аргумента функции
-- Локальные определения:
-- h - Double - шаг изменения аргумента функции
rectIntegral::(Double->Double)->(Double,Double)->Double->Double
rectIntegral f (a,b) n=sum (map f [a+h/2,a+3*h/2..b-h/2])*h
                        where
                            h=(b-a)/n

-- firstWhenEps - возвращает первый элемент списка, для которого
-- выполнено условие точности.
-- Параметры:
-- xs - [Double] - список приближений
-- eps - Double - значение точности
firstWhenEps::[Double]->Double->Double
firstWhenEps (x:xs) eps |abs(x-head xs)>eps = lastWhenEps (xs) eps
                       |otherwise = head xs

-- rectIntegrate - расчет определенного интеграла методом средних прямоугольников
-- до достижения указанной точности.
-- Параметры:
-- f - (Double->Double) - интегрируемая функция
-- (a,b) - (Double,Double) - границы интегрирования
-- eps - Double - значение точности
-- Локальные определения:
-- n0 - Double - первоначальное количество шагов
rectIntegrate::(Double->Double)->(Double,Double)->Double->Double
rectIntegrate f (a,b) eps = lastWhenEps (map (rectIntegral f (a,b))
                                (map (\x->2^x) [n0,n0+1..])) eps
                             where
                                n0=4
f x=x/(1+x^2)
```

Во втрой части содержится реализация графического интерфейса:

```
module Main where
import Prelude hiding (catch)
import Control.Exception
import Graphics.UI.Gtk
import Graphics.UI.Gtk.Builder
import MyIntegral


getValue:: Entry->IO (Double)
getValue entry=do
    y<-get entry entryText
    evaluate (read y)
```

```
onClickExit=mainQuit

getValues::[Entry]->IO [Double]
getValues xs=getValues' xs []

getValues'::[Entry]->String->IO [Double]
getValues' [] ers |ers==[]= return []
                  |otherwise= error ers

getValues' (x:xs) ers=do
    y<-try(getValue x)::IO(Either SomeException Double)

    case y of
        Left e->do
            v<-entryGetText x
            vs<-getValues' xs (ers++"Неправильное значение: \"" ++ v++"\"!\n")
            return vs
        Right v->do
            vs<-getValues' xs ers
            return (v:vs)
main::IO ()
main = do
   let

   initGUI
   gtkbuilder<-builderNew
   builderAddFromFile gtkbuilder "gui.glade"
   let
        showFail::SomeException-> IO()
        showFail e=do
            dlg<-messageDialogNew Nothing [DialogModal] MessageError
                ButtonsClose ("Ошибки:\n"++ show e)
            _<-dialogRun dlg
            widgetDestroy dlg
        onClickRun=do
            entryA <- builderGetObject gtkbuilder castToEntry "entry1"
            entryB <- builderGetObject gtkbuilder castToEntry "entry2"
            entryEps <- builderGetObject gtkbuilder castToEntry "entry3"
            labelResult <- builderGetObject gtkbuilder castToLabel "labelResult"
            [a,b,eps]<-getValues [entryA,entryB,entryEps]
            if a==b then error "Границы равны!" else do
                if eps<=0 then error "Точность меньше/равна 0!" else do
                    let
                        res=rectIntegrate f (a, b) (eps)
                    set labelResult [labelText:="Результат: "++ show res]
   window    <- builderGetObject gtkbuilder castToWindow "window1"
   buttonRun <- builderGetObject gtkbuilder castToButton "buttonRun"
   onClicked buttonRun (catch onClickRun showFail)
   buttonExit <- builderGetObject gtkbuilder castToButton "button2"
   onClicked buttonExit onClickExit
   onDestroy window mainQuit
   widgetShowAll window
   mainGUI
```

Тестовый пример

Ниже на рисунке 3 представлен пример работы численного интегрирования методом средних прямоугольников $\int_0^1 \frac{x}{1+x^2} dx$.

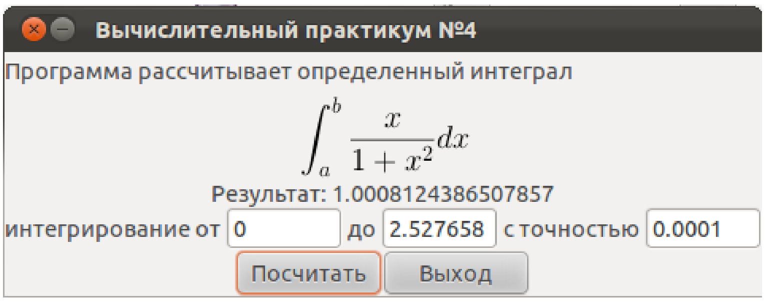

Рисунок 3 - Пример работы программы численного интегрирования

Вывод

В этой лабораторной работе я изучил различные методы численного интегрирования. Эти методы основаны на геометрическом смысле определенного интеграла. Самым простым методом интегрирования является метод прямоугольников. Численное интегрирование требуется при решении различных прикладных задач.