МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное

учреждение высшего образования

**«Национальный исследовательский Нижегородский государственный  
университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**Кафедра: Математической физики и оптимального управления**

Направление подготовки: «Математическое и компьютерное моделирование

в естественных науках»

Магистерская программа: «Математика и компьютерные науки»

**ОТЧЕТ**

по производственной практике

на тему:

«Приближенное вычисление интегралов»

Выполнил:

студент группы 381605м

Кабаров Антон Дмитриевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Научный руководитель:

Доцент кафедры МФиОУ

Галкин Олег Евгеньевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород

2016

Оглавление

[Введение 3](#_Toc482346616)

[1. Квадратурные формулы 4](#_Toc482346617)

[2. Оптимальные узлы 10](#_Toc482346618)

[3. Приближенное вычисление интеграла 15](#_Toc482346619)

[4. Интерполяция 37](#_Toc482346620)

[5. Интерполяционная задача Эрмита 40](#_Toc482346621)

[Список литературы 41](#_Toc482346622)

[Примечание 41](#_Toc482346623)

## Введение

В работе приведены квадратурные формулы простейшего вида и представлены теоремы для определения коэффициента в квадратурной формуле. Приведены оптимальные узлы.

Были представлены вычисления приближенными методами представленных в 1 главе методов и получены графические отображения погрешности с различным разбиением. Посчитана оценка погрешности.

## Квадратурные формулы

Данная глава посвящена квадратурным формулам, а так же формулировке и доказательству теоремы Пеано. Кроме того, в ней изложена теорема Чакалова [2]. Изложение ведется в соответствии с книгой [1].

**Определение**. Квадратурная формула — это формула для получения приближенного численного значения определенного интеграла.

Рассмотрим квадратурную формулу вида:

для которой алгебраический полином степени

, где коэффициент не зависит от функций, точки различны и лежат на интервале и каждый узел соответствует положительному числу . Квадратурная формула типа (1) всегда существует и единственна [1].

**Теорема**. Коэффициент определяется единственным способом.

*Доказательство*. Действительно, пусть единственный полином Эрмита степени , которая удовлетворяет условиям интерполяции вида:

(2)

Этот полином можно записать в виде:

где полином степени для которого выполняется следующее

,

, (4)

Из (2) видно, что квадратура имеет тип (1) с коэффициентом Такая квадратура называется интерполяция 1 и от остаточного члена Эрмитового многочлена

следует, что она точна для многочленов степени

Наоборот, если формула (1) точна для многочленов степени меньше

Из (4) и (6) следует, что

**Теорема 1 (Пеано)**

Пусть — линейный функционал вида:

где функции кусочно-непрерывные и лежат в интервале [a,b]. Если для каждого полинома степени меньше , то для каждой интегрируемой функции выполняется следующее:

,

где

*Доказательство*. Из формулы Тейлора

следует, что

Теорема доказана. □

Из остаточного члена интерполяционного многочлена Эрмита, следует, что

*.* Другой подход к определению и определение коэффициента Чакаловым в [2] с помощью следующей теоремы.

**Теорема 2**. Пусть у последовательности будут различные точки на отрезке и целые числа будут неотрицательными.

Пусть

Необходимое и достаточное условие для квадратурной формулы:

Если быть точным для полинома степени являются отношения , где коэффициент в разложении

остаточный член имеет вид:

где

*Доказательство*. Обозначим

Пусть квадратура (9) будет точным для многочленов степени меньше N. Для полинома степени следующие соотношения выполняются:

Последнее равенство дает и дает

Так как уравнение (11) рациональной функции является уникальной следует, что .

Докажем теперь, что если , тогда квадратура (9) точна для многочленов степени меньше Действительно, пусть удовлетворять

(10) . Из

следует что

Это означает, что для каждого квадратура (9) точна для полиномов

где

Следовательно,

Обе части последнего равенства являются многочленом степени и, следовательно,

Если является алгебраическим полиномом степени меньше, чем равенства

следует из линейной независимости полиномов .

Вид остаточного члена в теореме следует применять к теореме 1 для линейного функционала:

Теорема доказана. □

В случае равноудаленных узлов, и , интерполяция квадратурной формулы (1) называется формулой Ньютона Котеса. Примерами таких простых формул являются:

формула прямоугольника,

формула трапеций,

формула Симпсона,

Ньютоново правило ,

,

## Оптимальные узлы

В предыдущем разделе было показано, что для каждой системы узлов и произвольных неотрицательных целых чисел можно найти номера , такие что квадратура

точна для полиномов степени меньше

**Определение.** Говорят, что квадратурная формула имеет алгебраическую степень точности , если она точна для всех полиномов степени меньше .

*Возникает вопрос*: существуют ли точки и коэффициенты , для которых квадратурная формула (9) при тех же имеет алгебраическую степень точности выше чем

Ответ был дан Чакаловым [2] в теореме 4. Cледующая теорема 3 носит вспомогательный характер.

**Теорема 3 (**Чакалов, 1954**).** Для произвольных неотрицательных четных чисел существуют n различных точек таких, что многочлен

ортогонален любому многочлену степени меньше n на .

*Доказательство*.

Функция:

Положительна для каждого и имеет минимум в области мерном пространстве . Пусть , : , , где является положительным числом. Набор компактна и функция достигает минимальное значение в какой-то точке , Покажем, что функция принимает значение больше, чем в каждой точке при достаточно больших . В самом деле, если то для каждого , .

Следовательно,

Делим интервал на равных частей. Есть более чем точки в . Следовательно, существует подынтервал , которая не содержит точку из Делим интервал на три равные части и пусть лежат в середине. Очевидно и

Таким образом, если по крайней мере одна координата точки в лежит вне интервала , то функция принимает значение больше, чем . Выберем достаточное большое, что

Так как точка следует, что Следовательно, такой выбор гарантирует значение больше, чем для каждой точки вне . Вывод, что минимум в это также и она достигает некоторого конца точки в

Покажем, что координаты различны. Предполагая противное если Пусть , и

функция

принимает значение для и

Следовательно, функция принимает значения меньше, чем достаточным малым и положительным Это невозможно потому что равна заменяющий от для и от и , соответственно. Эта замена не нарушает порядок

при достаточно малых Точки для которой достигает своего минимума в различны. Следовательно,

Отсюда где являются основными многочленами Лагранжа в интерполяционном полиноме Лагранжа для узлов

. Так как каждый многочлен степени может быть представлена в виде линейной комбинации следует, что ортогонален любому полиному степени Докажем, что числа . Предполагая от противного пусть, по крайней мере один нуль из лежат вне . Тогда нули изстепени не больше Следовательно, является многочленом степени не больше и

Последнее равенство невозможно, так как функции не меняет знак в интервале Полученное противоречие доказывает, что все нули из лежат в интервале Наоборот, если многочлен ортогонален в для каждого многочлена степени то

1. Все нули действительны, различные и лежат в
2. Нули кратности нечетные числа;
3. Функция из (17) имеет локальный минимум в точке

Утверждение (1) доказано из ортогональности представленной раньше. Если предположить, что является четным числом многочлена степени и

что противоречит в И, наконец, из ,

отсюда следует, что функция имеет локальный минимум в точке

Теорема доказана. □

**Теорема 4.** Для каждого числа , неотрицательные числа существуют действительных чисел и коэффициентов, ,,, такие, что квадратурная формула (1) точна для любого многочлена степени меньше, чем

С другой стороны для произвольных чисел и квадратурная формула (1) не может быть точна для всех полиномов степени

*Доказательство.*

Докажем сначала обратное утверждение. Многочлен

степени , , , удовлетворяет и в то же время правая часть уравнения (1) обращается в нуль, так как . Таким образом, квадратурная формула (1) не является точным для многочлена при произвольных значениях x и Предполагая, что числа равны, пусть многочлен степени с нулями

и с кратностью ортогонален всем многочленам степени По теореме 1 такой многочлен существует. Пусть произвольный многочлен степени . Представляем в виде , где и являются многочленами степени и , набор узлов в (1) будут равны. Коэффициенты могут быть выбраны в теореме 1; квадратурная формула (1) точна для многочленов степени .

Обозначим через

следует, что

Многочлен ортогонален многочлену . Равенства

, , , означают, что и квадратура (1) точна для каждого многочлена степени. Рассмотрим случай, когда есть нечетные числа между числами . Для каждого обозначим через большим равным числом, которое удовлетворяет следующему условию ; т.е. . Так как числа равны можно построить квадратурную формулу:

которая является точной для многочленов степени . Квадратура (18) типа (1) с коэффициентами , для нечетное число и точна для любого многочлена степени ,

Теорема доказана. □

## Приближенное вычисление интеграла

В данной главе приведено несколько примеров вычисления интегралов с помощью методов, изложенных в предыдущих главах.

1. **Вычисление интеграла**

а). Сначала вычислим интеграл аналитически:

б) Приближенное вычисление

Формула Симпсона (число элементарных отрезков четно :

Оценка погрешности:

Формула правых прямоугольников:

Формула левых прямоугольников:

Оценка погрешности:

Ньютоново правило трех восьмых (разобьем отрезок на отрезков точками):

Оценка погрешности:

Формула трапеций:

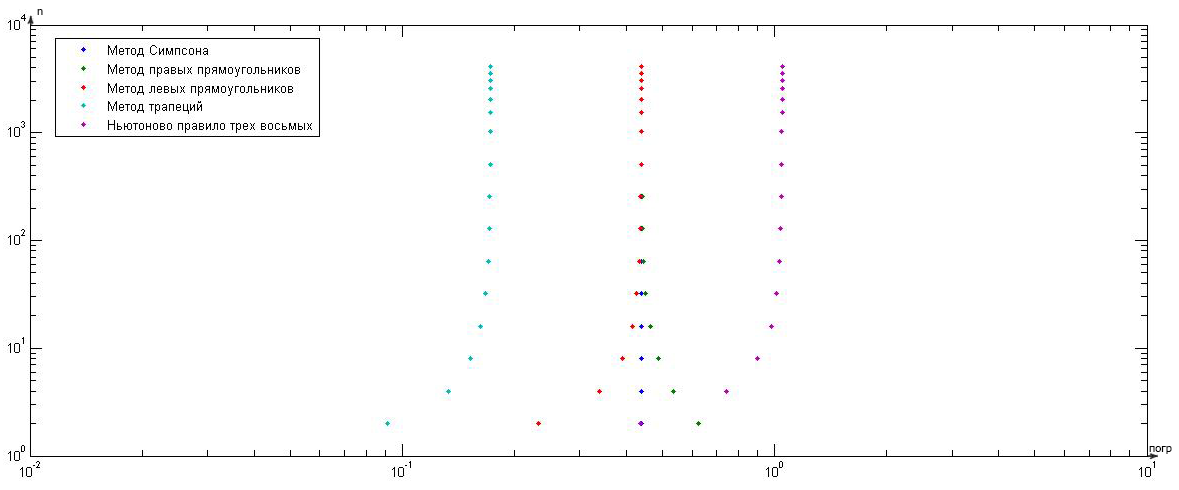
Оценка погрешности:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Название методов | n-число  узлов | 2 | 4 | 8 | 16 | 32 | 64 |
| **Метод**  **Симпсона** | Погрешность | 0.438884 | 0.438828 | 0.438825 | 0.438825 | 0.438825 | 0.438825 |
| Оценка погрешности | 0.1085∙10-4 | 6.8∙10-7 | 4∙10-8 | 2.6∙10-9 | 1.6∙10-10 | 1.03∙10-11 |
| **Метод правых**  **прямоугольников** | | Погрешность | 0.624523 | 0.534381 | 0.48726 | 0.463205 | 0.451056 | 0.44495 |
| Оценка  погрешности | 0.0052084 | 0.0013020 | 0.000325 | 0.0000813 | 0.0000203 | 0.00000508 |
| **Метод левых**  **прямоугольников** | | Погрешность | 0.231824 | 0.338032 | 0.389085 | 0.414118 | 0.426512 | 0.432678 |
| Оценка погрешности | 0.0052084 | 0.0013020 | 0.000325 | 0.0000813 | 0.0000203 | 0.00000508 |
| **Формула трапеций** | | Погрешность | 0.091037 | 0.132745 | 0.152793 | 0.162624 | 0.167491 | 0.169912 |
| Оценка погрешности | 0.0104167 | 0.0026041 | 0.000651 | 0.00016276 | 4.069∙10-5 | 1.01725∙10-5 |
| **Ньютоново правило трех восьмых** | | Погрешность | 0.436408 | 0.740571 | 0.901706 | 0.978313 | 1.01418 | 1.03116 |
| Оценка погрешности | 0.0000100 | 6.255∙10-4 | 3.90∙10-8 | 2.4435∙10--9 | 1.5272∙10-10 | 9.5452∙10-12 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Название  методов | | n-число  узлов | 128 | 256 | 512 | 1024 | 1536 |
| **Метод**  **Симпсона** | | Погрешность | 0.438825 | 0.438825 | 0.438825 | 0.438825 | 0.438825 |
| Оценка погрешности | 6.4∙10-13 | 4.0∙10-14 | 2.5∙10-15 | 1.5∙10-16 | 3.1∙10-17 |
| **Метод правых**  **прямоугольников** | Погрешность | 0.44189 | 0.440358 | 0.439591 | 0.439208 | 0.43908 |
| Оценка  погрешности | 0.0000127 | 3.2∙10-7 | 8∙10-8 | 2∙10-8 | 10∙10-9 |
| **Метод левых**  **прямоугольников** | Погрешность | 0.435754 | 0.43729 | 0.438057 | 0.438441 | 0.438569 |
| Оценка погрешности | 0.0000127 | 3.2∙10-7 | 8∙10-8 | 2∙10-8 | 10∙10-9 |
| **Метод трапеций** | Погрешность | 0.17112 | 0.171723 | 0.172025 | 0.172175 | 0.172226 |
| Оценка погрешности | 2.543∙10-9 | 6.357∙10-7 | 1.589∙10-7 | 3.9736∙10-8 | 1.7660∙10-8 |
| **Ньютоново правило трех**  **восьмых** | | | Погрешность | 1.03933 | 1.0433 | 1.04526 | 1.04623 | 1.04656 |
| Оценка погрешности | 5.965∙10-13 | 3.728∙10-14 | 2.330∙10-15 | 1.4564∙10-16 | 2.8770∙10-17 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Название  методов | n-число  узлов | 2048 | 2560 | | 3072 | 3584 | 4096 |
| **Метод**  **Симпсона** | Погрешность | 0.438825 | 0.438825 | | 0.438825 | 0.438825 | 0.438825 |
| Оценка погрешности | 9.8∙10-18 | 4.0∙10-18 | | 1.9∙10-18 | 1.05∙10-18 | 6.1∙10-19 |
| **Метод правых**  **прямоугольников** | Погрешность | 0.439016 | 0.438978 | | 0.438952 | 0.438934 | 0.43892 |
| Оценка  погрешности | 4.9∙10-9 | 3.1∙10-10 | | 2.2∙10-9 | 1.6∙10-9 | 1.2∙10-9 |
| **Метод левых**  **прямоугольников** | Погрешность | 0.438633 | | 0.438671 | 0.438697 | 0.438715 | 0.438729 |
| Оценка погрешности | 4.9∙10-9 | | 3.1∙10-10 | 2.2∙10-9 | 1.6∙10-9 | 1.2∙10-9 |
| **Метод**  **трапеций** | Погрешность | 0.172251 | 0.172266 | | 0.172276 | 0.172283 | 0.172288 |
| Оценка погрешности | 9.934∙10-9 | 6.3578∙10-9 | | 4.4151∙10-9 | 3.24∙10-9 | 2.48∙10-9 |
| **Ньютоново правило трех восьмых** | Погрешность | 1.04672 | 1.04681 | | 1.04713 | 1.04688 | 1.04696 |
| Оценка погрешности | 9.103042∙10-18 | 3.7286∙10-18 | | 1.7981∙10-18 | 9.70∙10-19 | 5.68∙10-19 |

График абсолютной погрешности вычисления интеграла:



**Вывод**. На рисунке показано что погрешность метода Симпсона лучше, чем использованные методы. Заметим так же что погрешность метода Ньютоново правило трех восьмых отличается от других погрешностей.

1. **Вычисление интеграла**

а). Сначала вычислим интеграл аналитически:

б) Приближенное вычисление

Формула Симпсона (число элементарных отрезков четно :

Оценка погрешности:

Формула правых прямоугольников:

Формула левых прямоугольников:

Оценка погрешности:

Формула трапеций:

Оценка погрешности:

Ньютоново правило трех восьмых (разобьем отрезок на отрезков точками):

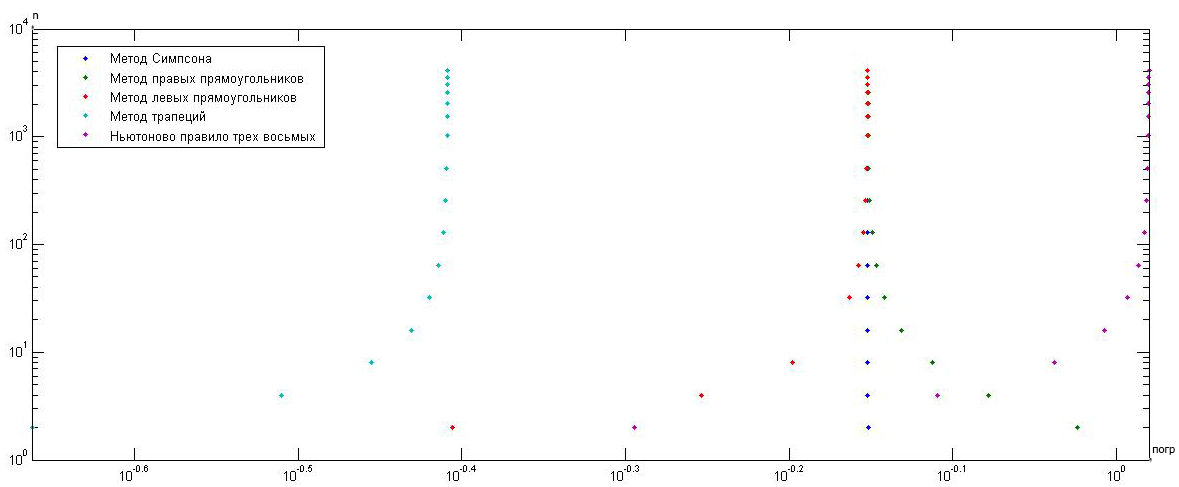
Оценка погрешности:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Название методов | n-число  узлов | 2 | | 4 | 8 | | 16 | 32 | | 64 |
| **Метод**  **Симпсона** | Погрешность | 0.705309 | | 0.704815 | 0.704791 | | 0.704789 | 0.704789 | | 0.704789 |
| Оценка погрешности | 7.8∙10-8 | | 3.9∙10-8 | 1.9∙10-8 | | 9.8∙10-9 | 4.9∙10-9 | | 2.4∙10-9 |
| **Метод правых**  **прямоугольников** | | Погрешность | 0.946273 | | 0.834747 | 0.771897 | | 0.738866 | 0.721958 | | 0.713406 |
| Оценка  погрешности | 0.0417 | | 0.0104167 | 0.002604 | | 0.000651 | 0.0000162 | | 4.0∙10-5 |
| **Метод левых**  **прямоугольников** | | Погрешность | 0.392699 | | 0.55796 | 0.633503 | | 0.370712 | 0.68736 | | 0.696107 |
| Оценка погрешности | 0.0417 | | 0.0104167 | 0.002604 | | 0.000651 | 0.0000162 | | 4.0∙10-5 |
| **Метод**  **трапеций** | | Погрешность | | 0.217388 | 0.308872 | 0.350691 | 0.370712 | | 0.380505 | 0.385347 | |
| Оценка погрешности | | 0.013333 | 0.003333 | 0.000833 | 0.0002083 | | 5.2083∙10-5 | 1.302∙10-5 | |
| **Ньютоново правило трех восьмых** | | Погрешность | | 0.507535 | 0.77768 | 0.916431 | 0.983751 | | 1.01619 | 1.03193 | |
| Оценка погрешности | | 1.53∙10-6 | 9.6∙10-8 | 6.00∙10-9 | 3.75∙10-10 | | 5.20∙10-5 | 1.30∙10-5 | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Название методов | n-число  узлов | 128 | 256 | 512 | 1024 | | 1536 |
| **Метод**  **Симпсона** | Погрешность | 0.704789 | 0.704789 | 0.704789 | 0.704789 | | 0.704789 |
| Оценка погрешности | 1.2∙10-9 | 6.1∙10-10 | 3.06∙10-10 | 1.53∙10-10 | | 1.02∙10-10 |
| **Метод правых**  **прямоугольников** | | Погрешность | 0.709106 | 0.70695 | 0.70587 | 0.70533 | | 0.70515 |
| Оценка  погрешности | 1.01∙10-5 | 2.5∙10-6 | 6.3∙10-7 | 1.5∙10-7 | | 7.06∙10-8 |
| **Метод левых**  **прямоугольников** | | Погрешность | 0.700456 | 0.702625 | 0.703708 | 0.704249 | | 0.704429 |
| Оценка погрешности | 1.01∙10-5 | 2.5∙10-6 | 6.3∙10-7 | 1.5∙10-7 | | 7.06∙10-8 |
| **Метод**  **трапеций** | | Погрешность | 0.387755 | 0.388955 | 0.389554 | | 0.389854 | 0.389954 |
| Оценка погрешности | 3.255∙10-6 | 8.138∙10-7 | 2.034∙10-7 | | 5.086∙10-8 | 2.260∙10-8 |
| **Ньютоново правило трех восьмых** | | Погрешность | 1.03964 | 1.04344 | 1.04532 | | 1.04626 | 1.04657 |
| Оценка погрешности | 3.25∙10-6 | 8.138∙10-7 | 2.034∙10-7 | | 5.086∙10-8 | 2.260∙10-8 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Название  методов | n-число  узлов | 2048 | 2560 | | 3072 | 3584 | 4096 | |
| **Метод**  **Симпсона** | Погрешность | 0.704789 | 0.704789 | | 0.704789 | 0.704789 | 0.704789 | |
| Оценка погрешности | 7.6∙10-11 | 6.1∙10-11 | | 5.1∙10-11 | 4.3∙10-11 | 3.8∙10-11 | |
| **Метод правых**  **прямоугольников** | Погрешность | 0.70506 | 0.705005 | | 0.704969 | 0.704944 | 0.704924 | |
| Оценка  погрешности | 3.9∙10-8 | 2.5∙10-8 | | 2.2∙10-9 | 1.7∙10-8 | 9.9∙10-9 | |
| **Метод левых**  **прямоугольников** | Погрешность | 0.704519 | | 0.704573 | 0.704609 | 0.704635 | 0.704654 | |
| Оценка погрешности | 4.9∙10-8 | | 2.5∙10-8 | 2.2∙10-9 | 1.7∙10-8 | 9.9∙10-9 | |
| **Метод**  **трапеций** | Погрешность | 0.390004 | | 0.390034 | 0.390053 | 0.390068 | | 0.390078 |
| Оценка  погрешности | 1.2715∙10-8 | | 8.138∙10-9 | 5.65∙10-9 | 4.15∙10-9 | | 3.17∙10-9 |
| **Ньютоново правило трех восьмых** | Погрешность | 1.04673 | | 1.04682 | 1.04689 | 1.04693 | | 1.04696 |
| Оценка  погрешности | 1.2715∙10-8 | | 8.138∙10-9 | 5.65∙10-9 | 4.15∙10-9 | | 3.17∙10-9 |

График абсолютной погрешности вычисления интеграла:

****

Вывод: Для данной функции метод Симпсона дает лучше погрешность.

1. **Вычисление интеграла**

а). Сначала вычислим интеграл аналитически:

б) Приближенное вычисление

Формула Симпсона (число элементарных отрезков четно :

Оценка погрешности:

Формула правых прямоугольников:

Формула левых прямоугольников:

Оценка погрешности:

Формула трапеций:

Оценка погрешности:

Ньютоново правило трех восьмых (разобьем отрезок на отрезков точками):

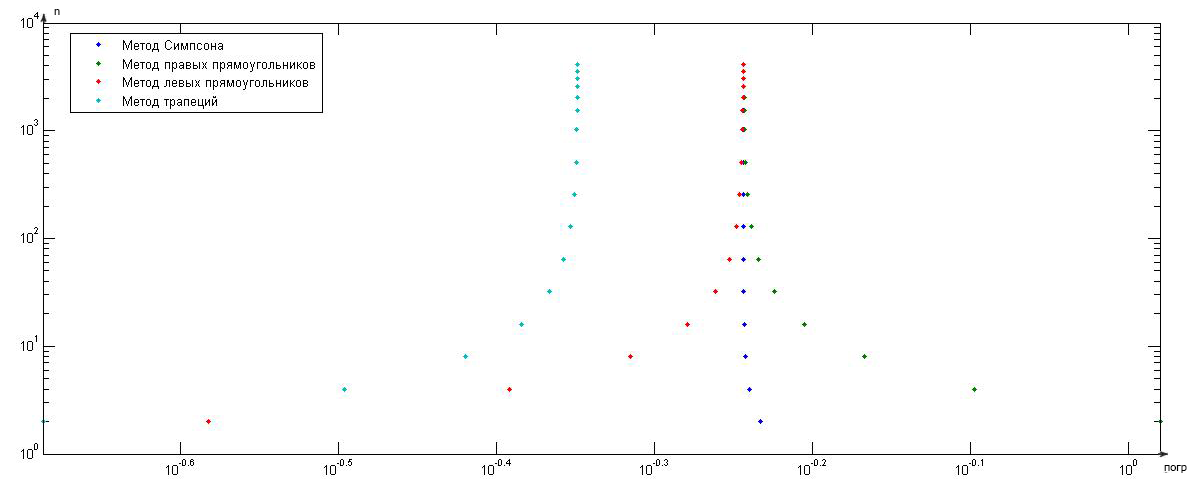
Оценка погрешности:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Название методов | n-число  узлов | 2 | | 4 | | 8 | | 16 | | 32 | | 64 | | |
| **Метод**  **Симпсона** | Погрешность | 0.58508 | | 0.57586 | | 0.572589 | | 0.57143 | | 0.571021 | | 0.570876 | | |
| Оценка погрешности | 0 | | | | | | | | | | | | |
| **Метод правых**  **прямоугольников** | | Погрешность | 1.0472 | | 0.798784 | | 0.680679 | | 0.624155 | | 0.596883 | | | 0.583622 | |
| Оценка  погрешности | 0 | | | | | | | | | | | | |
| **Метод левых**  **прямоугольников** | | Погрешность | 0.261799 | | 0.406085 | | 0.484329 | | 0.52598 | | 0.547796 | | 0.559078 | | |
| Оценка погрешности | 0 | | | | | | | | | | | | |
| **Метод трапеций** | | Погрешность | | 0.205617 | | 0.318939 | | 0.380391 | | 0.413104 | | 0.430238 | 0.439099 | |
| Оценка погрешности | | 0 | | | | | | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Название методов | n-число  узлов | 128 | 256 | 512 | 1024 | | 1536 |
| **Метод**  **Симпсона** | Погрешность | 0.570824 | 0.570806 | 0.5708 | 0.570798 | | 0.570797 |
| Оценка погрешности | 0 | | | | | |
| **Метод правых**  **прямоугольников** | | Погрешность | 0.57713 | 0.573935 | 0.572355 | 0.571572 | | 0.571313 |
| Оценка  погрешности | 0 | | | | | |
| **Метод левых**  **прямоугольников** | | Погрешность | 0.564858 | 0.567799 | 0.569287 | 0.570038 | | 0.57029 |
| Оценка погрешности | 0 | | | | | |
| **Метод**  **трапеций** | | Погрешность | 0.443639 | 0.445948 | 0.447117 | | 0.447707 | 0.447905 |
| Оценка погрешности | 0 | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Название  методов | n-число  узлов | 2048 | 2560 | | 3072 | 3584 | 4096 | |
| **Метод**  **Симпсона** | Погрешность | 0.570797 | 0.570797 | | 0.570797 | 0.570797 | 0.570797 | |
| Оценка погрешности | 0 | | | | | | |
| **Метод правых**  **прямоугольников** | Погрешность | 0.571183 | 0.571105 | | 0.571054 | 0.571017 | 0.570989 | |
| Оценка  погрешности | 0 | | | | | | |
| **Метод левых**  **прямоугольников** | Погрешность | 0.570416 | | 0.570492 | 0.570542 | 0.570579 | 0.570606 | |
| Оценка погрешности | 0 | | | | | | |
| **Метод**  **трапеций** | Погрешность | 0.448004 | | 0.448063 | 0.448103 | 0.448131 | | 0.448153 |
| Оценка  погрешности | 0 | | | | | | |

Приведен график абсолютной погрешности вычисления интеграла:



## Интерполяция

Данная глава посвещена истории интерполяции и сформулирована общая интерполяционная задача

**Определение**. Интерполяция - это метод поиска новых значений для любой функции с использованием заданного набора значений. Неизвестное значение в определенной точке может быть найдено с использованием многих формул интерполяции. Если новое значение должно быть найдено из двух заданных точек, тогда используется формула линейной интерполяции, тогда как при наличии набора чисел новое значение определяется с использованием интерполяционной формулы Лагранжа.

**Линейная интерполяционная формула задается**:

**Интерполяционная формула Лагранжа задается следующим образом**:

Одной из самых простых процедур аппроксимации функции является интерполяция. Значения функции в разных известных точках , ,.., , как известно, состоят в определении свободных параметров , , ... функции заданного вида. Условие интерполяции () = = () выполняются при

В первых производных от в некоторых точках также известны, можно также попытаться определить три параметра в такие, что для и с

Очевидно, что самыми простыми функциями являются полиномы. Интерполяция, возникла в значительной степени благодаря Джону Уоллсу[1656] (1616-1703), аппроксимация функции имеет давнюю историю, которая была начата после ранней попытки Томаса Харриота [1611] (1560-1621) Генри Бриггсом (1561-1630), когда он строил свою таблицу логарифмов. Его метод, описанный в его книге Arithmetica Logarithmica [1624], предполагал, что точки интерполяции равноудалены и используют конечные разности первого и второго порядка, и но не было дано никаких доказательств или предположений о том, как он получил свой метод. Это соответствует интерполяции полиномом второй степени. Схема для квадратичной интерполяции также была значительно полезна великому арабскому математику Аль-Бируни (973-1048) и, конечно же, многими его последователям.

Интерполяция полиномом любой степени (еще для равноудаленных точек), известная как формула Грегори-Ньютона, была обсуждена Джеймсом Грегори (1638-1675) в письме Джону Коллинзу (1625-1683) от 23 ноября 1670 года и Исаак Ньютон (1642-1727) в лемме 5 книги 3 его Principia Mathematica (1687), но формула была известна ему, по крайней мере, в 1676 году, как указал Фрейзер [1927]. Затем Ньютон распространил его на случай произвольных точек интерполяции, используя разделенные разности, термин впервые использовал Август де Морган (1806-1871) в своей книге «Дифференциальный и интегральный исчисление». Формула была написана в более подходящей форме Роджером Котсом (Roger Cotes, 1708). Другие интерполяционные формулы, названные в честь Гаусса, Стирлинга и Бесселя, уже были известны Ньютону. Похоже, что Готфрид Вильгельм Лейбниц (1646-1716) также получил аналогичные результаты независимо от 1672. Интерполяция в точках геометрической прогрессии приводит к упрощению, аналогичному случаю арифметической прогрессии. Эта проблема была впервые изучена Джеймсом Стирлингом [1730]. Затем Карл-Генрих Шелльбах (1864) представил результат Стирлинга в изящную алгоритмическую форму. Этот процесс был вновь открыт Карлом Рунге.

Отметим, что интерполяция, ныне представленная как чисто численный метод, имеет большое значение в развитии исчисления, поскольку формула Ньютона использовалась независимо Григорием и Ньютоном для получения биномиальной теоремы. Он также использовался Брук Тейлором (Brook Taylor, 1715) для получения разложения функций в бесконечный ряд.

Интерполяционные полиномы в точках могут быть рекурсивно вычислены для варьирования и , используя алгоритм, первый из которых принадлежит Александру Крейгу Айткену [1932] и улучшенному Эрику Гарольду Невиллу [1934]. Это схема Невилла-Айткена, которая была расширена Гюнтером Мулбахом [1973] для интерполяции любым семейством функций, образующих систему Чебышева. Эти две формулы приводят соответственно к ускорению путем экстраполяции

Интерполяционная задача в своей полной общности трактовалась Эрмитом [1878]. Теперь она называется интерполяционной проблемой Эрмита. Как уже упоминалось, Эрмит дал интегральное представление ошибки, которая была введена в обычном виде Стилтьесом [1882]. Интерполяция с некоторыми ненужными последовательными производными улучшается, называется интерполяцией Эрмита-Биркгофа. Он широко трактуется Лоренцем, Джеттером и Рименшнейдером [1983],

Пусть - векторное пространство размерности . Общая интерполяционная задача состоит в нахождении такой, что

для

Где - линейный функционал на . Если образуют базис то можно записать как и условия предшествующей интерполяции приводят к для являющейся системой из уравнений неизвестных

## Интерполяционная задача Эрмита

Задача для гладкой функции , характеризуется последовательностью линейных функционалов , где различные точки. Число функционалов равно и должно быть размерностью подмножества . Когда интерполяция осуществляется с помощью алгебраических многочленов, основная теорема следующая.

**Теорема 1**

Пусть гладкая функция на отрезке и различные точки из этого интервала. Для произвольных неотрицательных целых чисел существует единственный алгебраический многочлен , , который удовлетворяет условиям

## Список литературы

1. Ciarlet P.G. *HandBook of Numerical Analysis. Vol. III.* Elsevier Science.1994. 778 p.
2. Мысовских И. П. *О теореме Чакалова*. // Ж. вычисл. матем. и матем. физ. 1975, том 15, № 6. C. 1589–1593.

## Примечание

Приведены программы, встречающиеся в работе.

**Метод Симпсона**:

#include <cmath>

#include <iostream>

#include <conio.h>

#include <limits>

const int size=55000;

using namespace std;

int main()

{

setlocale(LC\_ALL,"");

double a,b;

int n;

cout << "Введите b " << endl;

cin >> b;

cout << "Введите a " << endl;

cin >> a;

cout << "Введите n " << endl;

cin >> n;

double z[size];

double p[size];

double result1=0;

double result2=0;

double result3=0;

double result4=0;

double k= (b-a) / (6\*n);

for (int i=0; i<=2\*n; i++)

{

z[2\*i-1]=(a+((b-a)\*(2\*i-1) /(2\*n)));

p[2\*i]=(a+((b-a)\*(2\*i) /(2\*n)));

if(i>0&&i<=n)

result1=result1+atan(2\*z[2\*i-1]);

if(i>0&&i<=n-1)

result2=result2+atan(2\*p[2\*i]);

}

for(int i=0; i<=2\*n;i++)

{

z[2\*i-1]=(a+((b-a)\*(2\*i-1) /(2\*n)));

p[2\*i]=(a+((b-a)\*(2\*i) /(2\*n)));

if(i>0&&i<=n)

result3=result3+atan(z[2\*i-1]);

if(i>0&&i<=n-1)

result4=result4+atan(p[2\*i]);

}

result3=k\*(cos(a)+4\*cos(result1)+2\*cos(result2)+cos(b));

cout << result2 << endl;

cout << "arctg(x): " << endl;

cout <<k\*(atan(a)+4\*result3+2\*result4+atan(b))<<endl; //arctg(x)

cout << "arctg(2x): " << endl;

cout <<k\*(atan(2\*a)+4\*result1+2\*result2+atan(2\*b))<<endl;//arctg(2\*x)

cout << "Оценка погрешности для arctg(2x): " << endl;

double s=(b-a)\*(b-a)\*(b-a)\*(b-a)\*(b-a);

cout << (2304\*1)/(625\*11750000\*2\*n);

getch();

return 0;

}

**Методы правых и левых прямоугольников и метод трапеций**:

#include <cmath>

#include <iostream>

#include <conio.h>

const int size=15000;

using namespace std;

int main()

{ double a,b,n;

setlocale(LC\_ALL,"");

double x[size];

double tr[size];

double result1=0;

double result2=0;

double result3=0;

double result4=0;

double result5=0;

double result6=0;

cout << "Введите b: " << endl;

cin >> b;

cout << "Введите a: " << endl;

cin >> a;

cout << "Введите n: " << endl;

cin >> n;

double h=((b-a)/n);

double h1=((b-a)/3\*n);

double k=(b-a)/(n);

for(int i=1;i<=n;i++)

{

x[i]=(a+(i\*(b-a)/(n)));

// cout << x[i] << endl;

result1=result1+atan(2\*x[i]);

}

cout << "Формула метода правых прямоугольников для функции arctg(2\*x): " << endl;

cout << k\*result1 << endl;

for(int j=0;j<=n-1;j++)

{

x[j]=(a+(j\*(b-a)/(n)));

// cout << x[i] << endl;

result2=result2+atan(2\*x[j]);

}

cout << "Формула метода левых прямоугольников для функции arctg(2\*x): " << endl;

cout << k\*result2;

cout << endl;

cout << "Формула трапеций для функции arctg(2\*x): " << endl;

for(int j=1;j<=n-1;j++)

{

tr[j]=a+((b-a)\*j)/n;

result3=result3+atan(2\*tr[j]);

}

cout << h\*( ((atan(2\*a)+atan(2\*b))/2)\*result3);

cout << endl;

for(int i=1;i<=n;i++)

{

x[i]=(a+(i\*(b-a)/(n)));

// cout << x[i] << endl;

result4=result4+atan(x[i]);

}

cout << "Формула метода правых прямоугольников для функции arctg(x): " << endl;

cout << k\*result4 << endl;

for(int j=0;j<=n-1;j++)

{

x[j]=(a+(j\*(b-a)/(n)));

// cout << x[i] << endl;

result5=result5+atan(x[j]);

}

cout << "Формула метода левых прямоугольников для функции arctg(x): " << endl;

cout << k\*result5 << endl;

cout << "Формула трапеций для функции arctg(x): " << endl;

for(int j=1;j<=n-1;j++)

{

tr[j]=a+((b-a)\*j)/n;

result6=result6+atan(tr[j]);

}

cout << h\*( ((atan(a)+atan(b))/2)\*result6);

cout << endl;

cout << "Оценка погрешности для функции arctg(2\*x) методом Трапеций: " << endl;

cout << 16\*(b-a)\*(b-a)\*(b-a) / (300\*n\*n) << endl;

cout << "Оценка погрешности: " << endl;

cout << h\*h\*(b-a)/6 << endl;

cout << "Оценка погрешности: " << endl;

double kg=(1513\*24/2799360)\*(b-a)\*(b-a)\*(b-a)\*(b-a)\*(b-a)\*h1\*h1\*h1\*h1;

cout << kg;

getch();

return 0;

}

**Ньютоново правило трех восьмых**:

#include <iostream>

#include <conio.h>

#include <math.h>

const int size=15000;

using namespace std;

int main()

{ setlocale(LC\_ALL,"");

int n;

double a,b;

double f[size];

double ff[size];

double fff[size];

double result1=0;

double result2=0;

double result3=0;

double result4=0;

double result5=0;

double result6=0;

cout << "Введите b: " << endl;

cin>>b;

cout << "Ввведите a: " << endl;

cin>>a;

cout << "Введите n: " << endl;

cin >> n;

for(int i=1;i<=n-1;i++)

{

fff[3\*i+1]=a+((b-a)\*(3\*i+1))/3\*n;

result3=result3+atan(fff[3\*i+1]);

}

for(int i=1;i<=n-1;i++)

{

ff[3\*i+2]=a+((b-a)\*(3\*i+2))/3\*n;

result2=result2+atan(ff[3\*i+2]);

}

for(int i=0;i<=n-1;i++)

{

f[3\*i]=a+((b-a)\*3\*i)/3\*n;

result1=result1+atan(f[3\*i]);

}

double k=2\*(b-a)/(24\*n);

cout << "Погрешность для функции arctg(x): ";

cout <<k\*(atan(a)+atan(b)+3\*result3+3\*result2+2\*result1);

cout << endl;

cout << "Оценка метода: " << endl;

double h1=((b-a)/3\*n);

double ks=(b-a)\*(b-a)\*(b-a)\*(b-a)\*(b-a);

double g=h1\*h1\*h1\*h1;

float kg=(1513\*24/2799360)\*ks\*g;

cout <<(1513/2799360) << endl;

for(int i=1;i<=n-1;i++)

{

fff[3\*i+1]=a+((b-a)\*(3\*i+1))/3\*n;

result4=result4+atan(2\*fff[3\*i+1]);

}

for(int i=1;i<=n-1;i++)

{

ff[3\*i+2]=a+((b-a)\*(3\*i+2))/3\*n;

result5=result5+atan(2\*ff[3\*i+2]);

}

for(int i=0;i<=n-1;i++)

{

f[3\*i]=a+((b-a)\*3\*i)/3\*n;

result6=result6+atan(2\*f[3\*i]);

}

cout << "Погрешность для функции arctg(2\*x): ";

cout <<k\*(atan(2\*a)+atan(2\*b)+3\*result4+3\*result5+2\*result6)<<endl;

cout << "Оценка погрешности для Ньютоново правила трех восьмых: " << endl;

cout << 3485952\*powl((b-a),9) / (141717600000\*powl(n,4));

getch();

return 0;

}

Графики:

x=[0.438884 0.438828 0.438825 0.438825 0.438825 0.438825 0.438825 0.438825 0.438825 0.438825 0.438825 0.438825 0.438825 0.438825 0.438825 0.438825];

y=[2 4 8 16 32 64 128 256 512 1024 1536 2048, 2560 3072 3584 4096];

x2=[0.624523 0.534381 0.48726 0.463205 0.451056 0.44495 0.44189 0.440358 0.439591 0.439208 0.43908 0.439016 0.438978 0.438952 0.438934 0.43892];

x3=[0.231824 0.338032 0.389085 0.414118 0.426512 0.432678 0.435754 0.43729 0.438057 0.438441 0.438569 0.438633 0.438671 0.438697 0.438715 0.438729];

x4=[0.091037 0.132745 0.152793 0.162624 0.167491 0.169912 0.17112 0.171723 0.172025 0.172175 0.172226 0.172251 0.172266 0.172276 0.172283 0.172288];

x5=[0.436408 0.740571 0.901706 0.978313 1.01418 1.03116 1.03933 1.0433 1.04526 1.04623 1.04656 1.04672 1.04681 1.04713 1.04688 1.04696];

loglog(x,y,'.',x2,y,'.',x3,y,'.',x4,y,'.',x5,y,'.')

legend('Метод Симпсона','Метод правых прямоугольников', 'Метод левых прямоугольников', 'Метод трапеций', 'Ньютоново правило трех восьмых');

x=[0.705309 0.704815 0.704791 0.704789 0.704789 0.704789 0.704789 0.704789 0.704789 0.704789 0.704789 0.704789 0.704789 0.704789 0.704789 0.704789];

y=[2 4 8 16 32 64 128 256 512 1024 1536 2048, 2560 3072 3584 4096];

x2=[0.946273 0.834747 0.771897 0.738866 0.721958 0.713406 0.709106 0.70695 0.70587 0.70533 0.70515 0.70506 0.705005 0.704969 0.704944 0.704924];

x3=[0.392699 0.55796 0.633503 0.370712 0.68736 0.696107 0.700456 0.702625 0.703708 0.704249 0.704429 0.704519 0.704573 0.704609 0.704635 0.704654];

x4=[0.217388 0.308872 0.350691 0.370712 0.380505 0.385347 0.387755 0.388955 0.389554 0.389854 0.389954 0.390004 0.390034 0.390053 0.390068 0.390078];

x5=[0.507535 0.77768 0.916431 0.983751 1.01619 1.03193 1.03964 1.04344 1.04532 1.04626 1.04657 1.04673 1.04682 1.04689 1.04693 1.04696];

loglog(x,y,'.',x2,y,'.',x3,y,'.',x4,y,'.',x5,y,'.')

legend('Метод Симпсона','Метод правых прямоугольников', 'Метод левых прямоугольников', 'Метод трапеций', 'Ньютоново правило трех восьмых');