Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное   
учреждение высшего образования

Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

**Отчет по лабораторной работе**

**« Вычисление многомерных интегралов с использованием многошаговой схемы (метод прямоугольников)»**

**Выполнил**:

студент группы 381708-2

Харченко И.Д.

**Проверил**:

Волокитин В.Д.

Нижний Новгород

2019

**Содержание**

1. [Постановка задачи 3](#_Toc26446451)
2. [Метод решения 4](#_Toc26446452)
3. [Схема распараллеливания 6](#_Toc26446453)
4. [Описание программной реализации 7](#_Toc26446454)
5. [Результаты экспериментов 10](#_Toc26446456)
6. [Заключение 11](#_Toc26446457)
7. [Приложение 12](#_Toc26446458)

# Постановка задачи

**Метод прямоугольников** является одним из методов численного интегрирования. Он позволяет вычислять определенные интегралы с заранее заданной степенью точности

Сначала остановимся на сути этого метода численного интегрирования, выведем формулу прямоугольников и получим формулу для оценки абсолютной погрешности метода. Далее по такой же схеме рассмотрим модификации метода прямоугольников, такие как метод правых прямоугольников и метод левых прямоугольников. В заключении рассмотрим подробное решение характерных примеров и задач с необходимыми пояснениями

# Метод решения

Пусть функция y = f(x) непрерывна на отрезке [a; b]. Нам требуется вычислить определенный интеграл формула.

Обратимся к понятию определенного интеграла .Разобьем отрезок [a;b] на n частей формулаточками формула. Внутри каждого отрезка формулавыберем точку формула. Так как по определению определенный интеграл есть предел интегральных сумм при бесконечном уменьшении длины элементарного отрезка разбиения формула, то любая из интегральных сумм является приближенным значением интеграла формула.

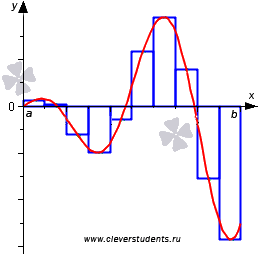
**Суть метода прямоугольников** заключается в том, что в качестве приближенного значения определенного интеграла берут интегральную сумму (далее мы покажем, какую именно интегральную сумму берут в методе прямоугольников).

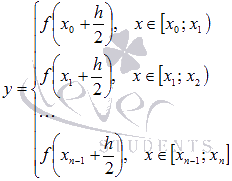
### *Формула метода средних прямоугольников.*

Если отрезок интегрирования [a;b] разбить на РАВНЫЕ части длины h точками формула(то есть формула) и в качестве точек формулавыбрать СЕРЕДИНЫ элементарных отрезков формула(то есть формула), то приближенное равенство формуламожно записать в виде формула. Это и есть **формула метода прямоугольников**. Ее еще называют формулой средних прямоугольников из-за способа выбора точек формула.

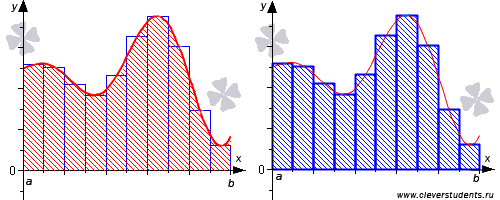
формуланазывают **шагом разбиения отрезка** [a;b].

Приведем графическую иллюстрацию метода средних прямоугольников.



Из чертежа видно, что подынтегральная функция y=f(x) приближается кусочной ступенчатой функцией на отрезке интегрирования.

С геометрической точки зрения для неотрицательной функции y=f(x) на отрезке [a;b] точное значение определенного интеграла представляет собой площадь криволинейной трапеции, а приближенное значение по методу прямоугольников – площадь ступенчатой фигуры.



# Схема распараллеливанияC:\Users\Администратор\Desktop\Сни2222мок.PNG

# Описание программной реализации

#include <stdio.h>

#include "mpi.h"

#include <math.h>

int main(int argc, char\*\* argv)

{

int my\_rank;

int p;

float a = 0.0, b=1.0, h;

int n = 2048;

int mode=3; /\* mode=1,2,3 rectangle, \*/

float local\_a, local\_b, local\_h;

float local\_integral, integral;

int source;

int dest = 0;

int tag = 0;

MPI\_Status status;

/\* function prototypes \*/

void Get\_data(float\* a\_ptr, float\* b\_ptr,

int\* n\_ptr, int my\_rank, int p, int \*mode\_ptr);

float rect(float local\_a, float local\_b, int local\_n);

/\* MPI starts \*/

MPI\_Init(&argc, &argv);

MPI\_Comm\_rank(MPI\_COMM\_WORLD,

&my\_rank);

MPI\_Comm\_size(MPI\_COMM\_WORLD, &p);

Get\_data(&a, &b, &n, my\_rank, p, &mode);

h = (b-a)/n;

local\_h=h/p;

local\_a = a + my\_rank\*local\_h\*n;

local\_b = local\_a + local\_h\*n;

switch(mode)

{

case(1):

local\_integral = rect(local\_a, local\_b, n);

}

if(my\_rank==0)

{

if (mode==1)

printf("Rectangle rule (0-point rule) is selected\n");

}

if (my\_rank == 0)

{

integral = local\_integral;

for (source = 1; source < p; source++)

{

MPI\_Recv(&local\_integral,1,MPI\_FLOAT,source,tag,MPI\_COMM\_WORLD, &status);

integral += local\_integral;}

}

else

{

printf("The intergal calculated from process %d is %f \n", my\_rank,local\_integral);

MPI\_Send(&local\_integral, 1, MPI\_FLOAT, dest, tag, MPI\_COMM\_WORLD);

}

if (my\_rank == 0)

{

printf("With n = %d, the total integral from %f to %f= %f\n",n, a,b,integral);

}

/\* MPI finished \*/

MPI\_Finalize();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Function Get\_data

\* Reads in the user input a, b, and n.

\* Input parameters:

\* 1. int my\_rank: rank of current process\* 2. int p: number of processes.

\* Output parameters:

\* 1. float\* a\_ptr: pointer to left endpoint a.

\* 2. float\* b\_ptr: pointer to right endpoint b.

\* 3. int\* n\_ptr: pointer to number of trapezoids.

3. int\* mode\_ptr: pointer to mode of rule of NewtonCotes methods

\* Algorithm:

\* 1. Process 0 prompts user for input and

\* reads in the values.

\* 2. Process 0 sends input values to other

\* processes.

\*/

void Get\_data(

float\* a\_ptr /\* out \*/,

float\* b\_ptr /\* out \*/,

int\* n\_ptr /\* out \*/,

int my\_rank /\* in \*/,

int p /\* in \*/,

int\* mode\_ptr /\* out \*/)

{

int source = 0; /\* All local variables used by \*/

int dest; /\* MPI\_Send and MPI\_Recv \*/

int tag;

MPI\_Status status ;

if (my\_rank == 0)

{

do

{

printf("Enter a, b, n(1024), and mode(1--rect):\n");

scanf("%f %f %d %d", a\_ptr, b\_ptr, n\_ptr, mode\_ptr);

}

while (\*mode\_ptr<1 || \*mode\_ptr>3);

for (dest = 1; dest < p; dest++)

{

tag = 0;

MPI\_Send(a\_ptr, 1, MPI\_FLOAT, dest, tag,MPI\_COMM\_WORLD);

tag = 1;

MPI\_Send(b\_ptr, 1, MPI\_FLOAT, dest, tag,MPI\_COMM\_WORLD);

tag = 2;

MPI\_Send(n\_ptr, 1, MPI\_INT, dest, tag,MPI\_COMM\_WORLD);

tag = 3;

MPI\_Send(mode\_ptr, 1, MPI\_INT, dest, tag,MPI\_COMM\_WORLD);

}

}

else

{

tag = 0;

MPI\_Recv(a\_ptr, 1, MPI\_FLOAT, source, tag,MPI\_COMM\_WORLD, &status);

tag = 1;

MPI\_Recv(b\_ptr, 1, MPI\_FLOAT, source, tag,MPI\_COMM\_WORLD, &status);

tag = 2;

MPI\_Recv(n\_ptr, 1, MPI\_INT, source, tag,MPI\_COMM\_WORLD, &status);

tag = 3;

MPI\_Recv(mode\_ptr, 1, MPI\_INT, source, tag,MPI\_COMM\_WORLD, &status);

}

} /\* Get\_data \*/

float rect( float local\_a, float local\_b, int local\_n)

{

float local\_integral;

float x;

int i;

float local\_h;

float f(float x);

local\_h=(local\_b-local\_a)/local\_n;

local\_integral = f(local\_a);

x = local\_a;

for (i = 1; i <= local\_n-1; i++)

{

x = x + local\_h;

local\_integral += f(x);

}

local\_integral \*=local\_h;

return local\_integral;

}

float trap( float local\_a, float local\_b, int local\_n)

{

float local\_integral;

float x;

int i;

float local\_h;

float f(float x);

local\_h=(local\_b-local\_a)/local\_n;

local\_integral = f(local\_a) + f(local\_b);

x = local\_a;

for (i = 1; i <= local\_n-1; i++)

{

x = x + local\_h;

local\_integral += 2.0\*f(x);

}

local\_integral \*=local\_h/2.0;

return local\_integral;

}

float simp( float local\_a, float local\_b, int local\_n )

{

float local\_integral;

float x;

int i;

float local\_h;

float f(float x);

local\_h=(local\_b-local\_a)/local\_n;

local\_integral = f(local\_a) + f(local\_b);

x = local\_a;

for (i = 1; i < local\_n; i++)

{

x = x + local\_h;

if (i % 2 == 0) /\* if i is even \*/

local\_integral = local\_integral + 2 \* f(x);

else /\* if i is odd \*/

local\_integral = local\_integral + 4 \* f(x);

}

local\_integral \*=local\_h/3.0;

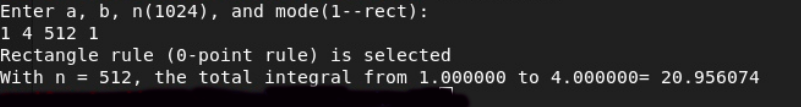
return local\_integral;

}

float f(float x)

{ return x\*x; }

# Результаты экпериментов



Как показано в командной строке, мы вычислили площадь, используя интегральную функцию, с помощью метода треугольников.

Разбив функцию на 512 треугольников, мы вычислили её площадь, которая равняется приблизительно 20.

# Заключение

***Метод прямоугольников*** *– хороший* способ вычисления определённых интегралов для нахождения площади.В ходе работы были реализованы алгоритмы, позволяющие вычислять интеграл с помощью метода прямоугольников.

# Приложение

1.studopedia.su - Студопедия (2013 - 2020)- Вычисление многомерных интегралов (<https://studopedia.su/6_20982_vichislenie-mnogomernih-integralov.html>)

2. Сибирский государственный университет телекоммуникаций и информатики-

Лекция 5. Параллельное численное интегрирование

(<http://www.mkurnosov.net/teaching/uploads/PCT/pct-fall2017-lec5.pdf>)

3. Материал из Википедии — свободной энциклопедии - метод прямоугольников (<https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BF%D1%80%D1%8F%D0%BC%D0%BE%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B8%D0%BA%D0%BE%D0%B2>)