|  |  |
| --- | --- |
| Изображение выглядит как текст, керамические изделия, фарфор  Автоматически созданное описание | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

Факультет «Фундаментальные науки»

Кафедра «Техническая физика»

**Лабораторная работа №7**

по курсу «Вычислительная физика»

Выполнили: Коберник Т. Н., Плетенёв Б. А.

Группа: ФН4-71Б

Преподаватели: Хасаншин Р.Х., Ивлиев П.А.

Москва, 2022 г.

Содержание

[1. Теоретическая часть 3](#_Toc118067027)

[1.1. Вычисление многомерных интегралов 3](#_Toc118067028)

[1.2. Основные преимущества ММК. 9](#_Toc118067029)

[1.3. О применении метода Монте-Карло 9](#_Toc118067030)

[2. Постановка задачи 10](#_Toc118067031)

[3. Программа 11](#_Toc118067032)

[4. Результаты 12](#_Toc118067033)

[5. Выводы 12](#_Toc118067036)

1. **Теоретическая часть**

## **. Вычисление многомерных интегралов**

Пускай необходимо вычислить интеграл:

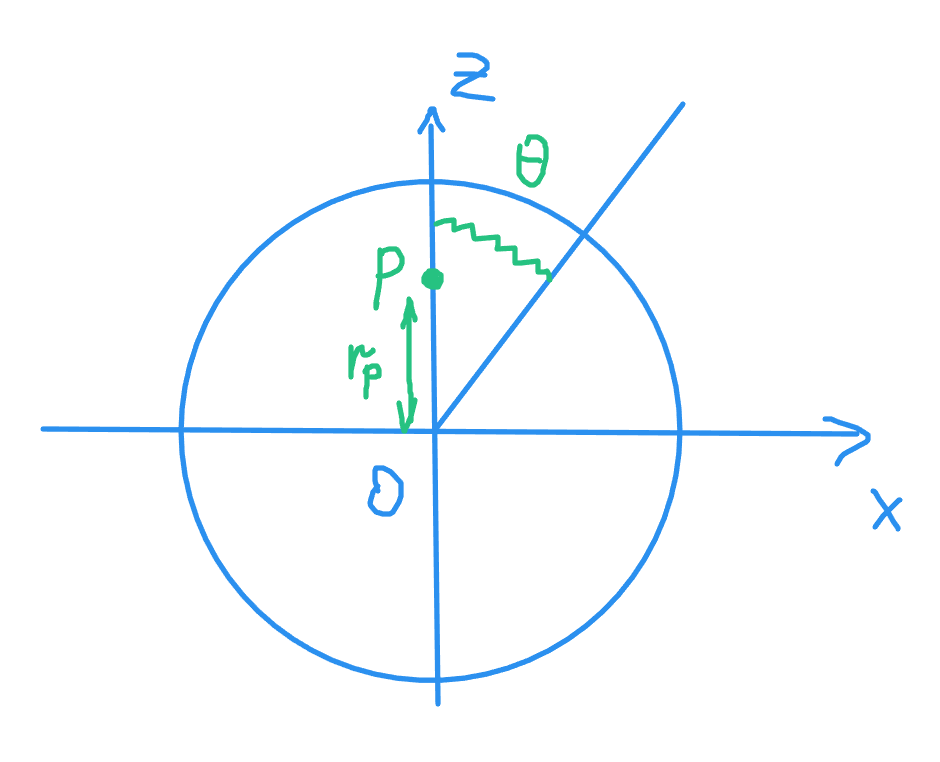
Простейшие методы нахождения интегралов для одномерного случая, которые были рассмотрены в предыдущей лабораторной, можно использовать и для N-мерного.

Рассмотрим единичный шар S, точки

Тогда искомый интеграл можно свести к интегралу вида:

Рассмотрим независимые случайные точки P и Q, равномерно распределенные в единичном шаре.

Не теряя общности примем, что .



Возьмём в плоскости Oxz случайное направление . зададим по закону

На этом направлении возьмем точку Q так, что

Тогда по теореме косинусов

Рассмотрим мат. ожидание следующей величины:

С учётом выражения (1)

Таким образом

* 1. **. Вычисление многомерных интегралов c особенностями**

Первый тип

Область интегрирования , а подынтегральная функция содержит особенности

А) Если интеграл

существует, то можно полагать, что дисперсия при вычислении интеграла простейшими методами конечна. Тогда предлагается использовать существенную выборку: взять плостность распределения С.В. с той же особенностью, что и . Этот прием называют ***включением особенности в плотность распределения***.

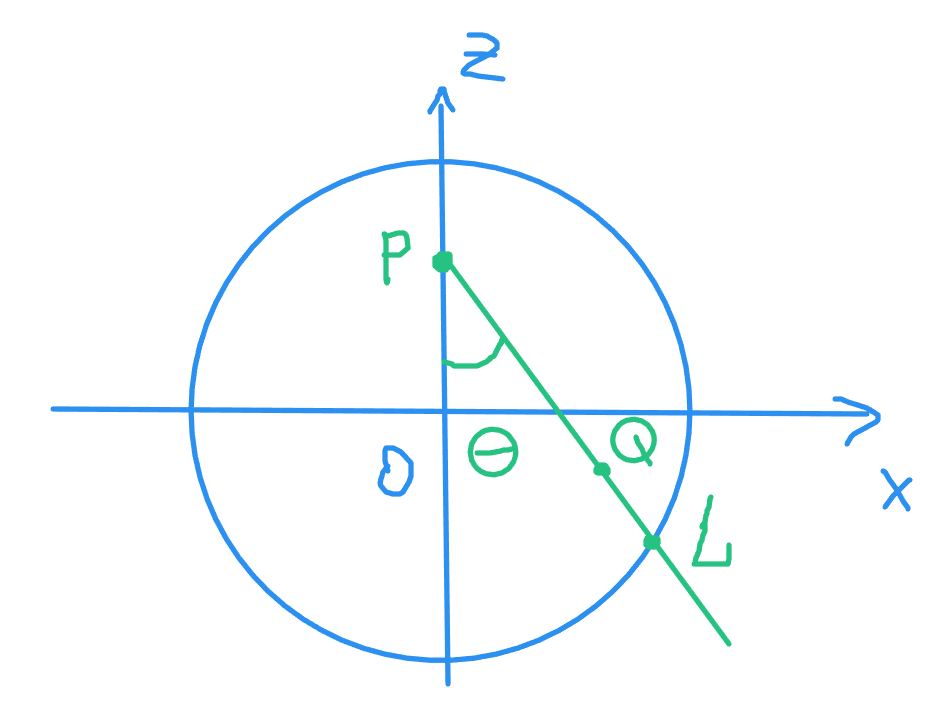
Как было показано выше

Где S – единичный шар, точки

Рассмотрим случайную точку на оси Oz, такую что

А распределение точки Q постараемся взять таким образом, чтобы её плотность распределения

Для этого воспользуемся СК с началом в точке P. Возьмём произвольное направление . зададим по закону



Разыграем точку Q, равномерно распределенную на отрезке . Тогда

Из треугольника PLO

**Б) *Выборка по группам***

Некий аналог существенной выборки, в которой предлагается определить оптимальную плотность распределения по всей области определения.   
В выборке по группам указывается, в какой области выбирать какое количество точек.

В каждой подобласти указывается, какое количество испытаний необходимо провести.

На этом этапе обсуждается ценность областей для исходного интеграла. В зависимости от вклада проводится соответствующее количество испытаний.

Рассмотрим С.В. c плотностью распределения .

Это значение получено из условия нормировки

Рассмотрим

Если есть возможность проинтегрировать его по части переменных аналитически, то это уменьшает дисперсию. Однако мы можем получить очень громоздкую подынтегральную функцию для дальнейших вычислений.

Второй тип

Область G не ограничена.

А) Отбросить часть интеграла от достаточно удалённой части области

Б) Преобразовать интеграл к случаю 1 (преобразование координат)

В) Использовать плотность распределения С.В., спадающую на удалённой части с большой скоростью.

## **. Основные преимущества ММК.**

1. В многомерном случае при применении метода Монте-Карло число вычисляемых значений подынтегральной функции растет значительно медленнее относительно  (– точность вычисляемого интеграла), чем в квадратурных формулах.
2. Точность метода Монте-Карло не зависит от гладкости подынтегральной функции.
3. Простая приспособляемость к форме области интегрирования. Например, полагаем , где  – квадрат, содержащий данную область , , для применения формулы вида , где  – независимые реализации равномерно распределенных в (0,1) случайных величин  соответственно.

# **1.4. О применении метода Монте-Карло**

1. Функция  «достаточно гладкая» и область  – «достаточно хорошая». Такие интегралы на практике вычисляют методом Монте-Карло при  ( - кратность интеграла).
2. Функция  «кусочно гладкая». Интегралы от таких функций вычисляют методом Монте-Карло при . Сюда относится также случай «плохой» ограниченной области , так как ее можно всегда заключить в куб, доопределив функцию нулем.
3. Функция  «очень плохая». Можно указать столь широкий класс функций, что для них нельзя построить квадратурную формулу с порядком сходимости лучшим, чем . В случае функций одной переменной таким классом является, например,  класс функций , удовлетворяющих условию Липшица порядка  при :

.

Если о функции  известно лишь то, что она принадлежит некоторому классу, более широкому, чем  при  (и в этом смысле она «очень плохая»), то, как правило, вычислять интеграл от нее лучше методом Монте-Карло (даже при ).

1. **Постановка задачи**

Вычислить ММК многомерные интегралы.

Используя оценки  и 

приближенно вычислить (при  и ) следующий интеграл , (т. е. , , , )

где точки  и  принадлежат единичному шару  трехмерного пространства; . Эти интегралы  могут быть вычислены аналитически. Сравните точные значения с результатами расчетов. Сделайте выводы.

1. **Программа**
2. **from** numpy **import** random
3. **import** math
5. **def** function1(r):
6. **return** 1 **/** r
8. **def** function2(r):
9. **return** 1 **/** (r **\*\*** 2)
11. **def** function3(r):
12. **return** r
14. **def** for\_ideal\_functions(function, a, b, n):
15. sum **=** 0
16. **for** i **in** range(n):
17. r\_p, r\_q, theta **=** random.uniform(a, b), random.uniform(a, b), random.uniform(**-**b, b)
18. r **=** (r\_p **\*\*** 2 **+** r\_q **\*\*** 2 **-** 2 **\*** theta **\*** r\_p **\*** r\_q) **\*\*** (1 **/** 2)
19. sum **+=** function(r)
20. **return** 16 **\*** sum **/** (n **\*** 9)
22. **def** for\_non\_ideal\_functions(function, a, b, n):
23. **if** function **==** function2:
24. parameter **=** 1 **/** 3
25. **else**:
26. parameter **=** 1
27. sum **=** 0
28. **for** i **in** range(n):
29. r\_p, r\_q, theta **=** random.uniform(a, b) **\*\*** (parameter), random.uniform(a, b), 2 **\*** random.uniform(a, b) **-** 1
30. r **=** ((r\_p **\*\*** 2) **+** (r\_q **\*\*** 2) **-** 2 **\*** theta **\*** r\_p **\*** r\_q) **\*\*** (1 **/** 2)
31. l **=** r\_p **\*** theta **+** (1 **-** (r\_p **\*\*** 2) **\*** (1 **-** theta **\*\*** 2)) **\*\*** (1 **/** 2)
32. sum **+=** ((r **\*\*** 2) **\*** function(r) **\*** l)
33. **return** 16 **\*** sum **/** (n **\*** 3)
35. print(f'Метод вычисления среднего при m = 1: {for\_ideal\_functions(function1, 0, 1, 100)}')
36. print(f'Метод вычисления среднего при m = 2: {for\_ideal\_functions(function2, 0, 1, 100)}')
37. print(f'Метод вычисления среднего при m = -1: {for\_ideal\_functions(function3, 0, 1, 20000)}')
39. print(f'Метод вычисления среднего c особенностью при m = 1: {for\_non\_ideal\_functions(function1, 0, 1, 50000)}')
40. print(f'Метод вычисления среднего c особенностью при m = 2: {for\_non\_ideal\_functions(function2, 0, 1, 50000)}')
41. print(f'Метод вычисления среднего c особенностью при m = -1: {for\_non\_ideal\_functions(function3, 0, 1, 50000)}')
42. **Результаты**

Точные значения интегралов: , , 

В результате работы программы были получены следующие результаты

Метод вычисления среднего при m = 1: 3.4103873458521585  
Метод вычисления среднего при m = 2: 25.67706086181594  
Метод вычисления среднего при m = -1: 1.3336066525800243  
Метод вычисления интеграла c особенностью при m = 1:2.9417313809954257  
Метод вычисления интеграла c особенностью при m = 2:3.9950764647526076  
Метод вычисления интеграла c особенностью при m = -1: 1.663436913341074

1. **Выводы**

По результатам проведенных вычислений можно сделать вывод о том, что рассмотренные в теоретической части формулы для вычисления многомерных интегралов с применением метода Монте-Карло дают недостаточно точные результаты (точность достигает только первого порядка). При этом Первый метод совершенно не применим к первым двум подынтегральным функциям, т.к. они имеют особенности в нулевой точке. Результаты вычислений для этих функций оказались сильно далеки от действительности. В противовес этому, второй метод дал удовлетворительные значения. Лучше всего он рассчитал интеграл для второй функции (с m = 2).