Муниципальное бюджетное общеобразовательное учреждение
Лицей №185

Открытый городской конкурс исследовательских проектов
Учащихся 9-11 классов

Направление: математический проект

«Метод Монте-Карло при решения
нестандартных задач»

Работу выполнила
ученица 11 «М» класса
Кашевская Анна Алексеевна
Руководитель пректа:
Учитель математики
Глушкова Татьяна Александровна

г. Новосибирск, 2020

Оглавление

Введение.. 3
Теоретические аспекты . История создания метода Монте Карло 4
Алгоритм Бюффона для определения числа Пи.. 4
Методы Монте-Карло 4
Связь стохастических процессов и дифференциальных уравнений 5
Рождение метода Монте-Карло в Лос-Аламосе 6
Дальнейшее развитие и современность 10
Использование метода 10
Вычислить площадь фигуры.. 10
Факторизация чисел 13
Решение интегралов.. 16
Заключение 17

Введение

В наше время наука прогрессирует быстро и стремительно. Существует огромное количество методов решения задач в математике, однако требуются такие, которые будут делать это быстро, эффективно и будут способны дать решения необычным задачам, которые решить «классическим» путём может быть трудно или невозможно. Один из таких методов это Метод Монте Карло, который может позволить нам это сделать. Именно он меня и заинтересовал своей необычностью и огромными возможностями в решениях задач разного рода.

Цель: исследовать метод Монте Карло и найти применение в нестандартных задачах по математике, ИКТ, физике.

Задачи:

-Ознакомиться с историей метода

-Изучить основные принципы метода

-Понять какие разделы математики необходимы для работы с методом

-Познакомиться с тем, где уже применяется этот метод

-Провести анализ задач

Теоретические аспекты . История создания метода Монте Карло . Алгоритм Бюффона для определения числа Пи

Методы Монте-Карло (ММК) — группа численных методов для изучения случайных процессов. Суть метода заключается в следующем: процесс описывается математической моделью с использованием генератора случайных величин, модель многократно обсчитывается, на основе полученных данных вычисляются вероятностные характеристики рассматриваемого процесса. Например, чтобы узнать методом Монте-Карло, какое в среднем будет расстояние между двумя случайными точками в круге, нужно взять координаты большого числа случайных пар точек в границах заданной окружности, для каждой пары вычислить расстояние, а потом для них посчитать среднее арифметическое.**[1]**

Случайные величины использовались для решения различных прикладных задач достаточно давно. Примером может служить способ определения числа Пи, который был предложен Бюффоном еще в 1777 году. Суть метода была в бросании иглы длиной L на плоскость, расчерченную параллельными прямыми, расположенными на расстоянии r друг от друга. Вероятность того, что отрезок пересечет прямую, связана с числом Пи:

$$p = \int_0^{\pi} \int_0^{l\sin\theta} \frac{1}{r\pi} dA d\theta$$

, где

- A — расстояние от начала иглы до ближайшей к ней прямой;
- θ — угол иглы относительно прямых.

Этот интеграл просто взять: $p = \frac{2L}{r\pi}$ (при условии, что $r > L$), поэтому подсчитав долю отрезков, пересекающих прямые, можно приближенно определить это число. При увеличении количества попыток точность получаемого результата будет увеличиваться.

В 1864 году капитан Фокс, выздоравливая после ранения, чтобы как-то занять себя, реализовал эксперимент по бросанию иглы. Результаты представлены в следующей таблице:

	Число бросаний	Число пересечений	Длина иглы	Расстояние между прямыми	Вращение	Значение Пи
Первая попытка	500	236	3	4	отсутствует	3.1780
Вторая попытка	530	253	3	4	присутствует	3.1423
Третья попытка	590	939	5	2	присутствует	3.1416

Комментарии:

- Вращение плоскости применялось (и как показывают результаты — успешно) для того, чтобы уменьшить систематическую ошибку.
- В третьей попытке длина иглы была больше расстояния между линиями, что позволило не увеличивая числа бросаний эффективно увеличить число событий и повысить точность.

Связь стохастических процессов и дифференциальных уравнений

Создание математического аппарата стохастических методов началось в конце XIX века. В 1899 году лорд Релей показал, что одномерное случайное блуждание на бесконечной решётке может давать приближенное решение параболического дифференциального уравнения. Андрей Николаевич Колмогоров в 1931 году дал большой толчок развитию стохастических подходов к решению различных математических задач, поскольку он сумел доказать, что цепи Маркова связаны с некоторыми интегро-дифференциальными уравнениями. В 1933 году Иван Петровский показал, что случайное блуждание, образующее Марковскую цепь, асимптотически связано с решением эллиптического дифференциального уравнения в частных производных. После этих открытий стало понятно, что стохастические процессы можно описывать дифференциальными уравнениями и, соответственно, исследовать при помощи хорошо на тот момент разработанных математических методов решения этих уравнений.

Стохастичность (др. -греч. στόχος — цель, предположение) означает случайность. **Случайный (стохастический) процесс** — это процесс, поведение которого не является детерминированным, и последующее состояние такой системы описывается как величинами, которые могут быть предсказаны, так и случайными.**[2]**

Дифференциа́льное уравне́ние — уравнение, в которое входят производные функции, и может входить сама функция, независимая переменная и параметры. Порядок входящих в уравнение производных может быть различен.**[3]**

Рождение метода Монте-Карло в Лос-Аламосе

Сначала Энрико Ферми в 1930-х годах в Италии, а затем Джон фон Нейман и Станислав Улам в 1940-х в Лос-Аламосе предположили, что можно использовать связь между стохастическими процессами и дифференциальными уравнениями «в обратную сторону». Они предложили использовать стохастический подход для аппроксимации многомерных интегралов в уравнениях переноса, возникших в связи с задачей о движении нейтрона в изотропной среде.

Идея была развита Уламом, который, по иронии судьбы, также как и Фокс боролся с вынужденным бездельем во время выздоровления после болезни, и, раскладывая пасьянсы, задался вопросом, какова вероятность того, что пасьянс «сложится». Ему в голову пришла идея, что вместо того, чтобы использовать обычные для подобных задач соображения комбинаторики, можно просто поставить «эксперимент» большое число раз и, таким образом, подсчитав число удачных исходов, оценить их вероятность. Он же предложил использовать компьютеры для расчётов методом Монте-Карло.

Появление первых электронных компьютеров, которые могли с большой скоростью генерировать **псевдослучайные числа**, резко расширило круг задач, для решения которых стохастический подход оказался более эффективным, чем другие математические методы. После этого произошёл большой прорыв и метод Монте-Карло применялся во многих задачах, однако его использование не всегда было оправдано из-за большого количества вычислений, необходимых для получения ответа с заданной точностью.

Годом рождения метода Монте-Карло считается 1949 год, когда в свет выходит статья Метрополиса и Улама «Метод Монте-Карло». Название метода происходит от названия города в княжестве Монако, широко известного своими многочисленными казино, поскольку именно рулетка является одним из самых широко известных генераторов случайных чисел. Станислав Улам пишет в своей автобиографии «Приключения математика», что название было предложено Николасом Метрополисом в честь его дяди, который был азартным игроком.**[4]**

Доказательство эффективности метода

Пусть необходимо вычислить линейный функционал $J_n = (\varphi, h)$, где $\varphi = K\varphi + f$, причём для интегрального оператора K с ядром $K(x', x)$ выполняется условие, обеспечивающее сходимость ряда Неймана: $\left\|K^{n_0}\right\| < 1$. Цепь Маркова $\{x_n\}$ определяется начальной плотностью $\pi(x)$ и переходной плотностью $p(x', x) = p(x' \to x)$; вероятность обрыва цепи в точке x' равна $g(x') = 1 - \int p(x', x)dx$. N – случайный номер последнего состояния. Далее определяется функционал от траектории цепи, математическое ожидание которого равно J_h. Чаще всего используется так называемая оценка по столкновениям $\xi = \sum_{n=0}^{N} Q_n h(x_n)$, где $Q_0 = \frac{f(x_0)}{\pi(x_0)}$, $Q_n = Q_{n-1}\frac{k(x_{n-1}, x_n)}{p(x_{n-1}, x_n)}$. Если $p(x', x) \neq 0$ при $k(x', x) \neq 0$, и $\pi(x) \neq 0$ при $f(x) \neq 0$, то при некотором дополнительном условии $E\xi = \sum_{n=0}^{\infty}(K^n f, h) = (\varphi, h) = \int_X \varphi(x)h(x)dx$. Важность достижения малой дисперсии в знакопостоянном случае показывает следующее утверждение: если $\pi(x) = \frac{f(x)\varphi^*(x)}{(f, \varphi^*)}$ и $p(x', x) = \frac{k(x', x)\varphi^*(x)}{[K^* p^*](x')}$, где $\varphi^* = K^*\varphi^* + h$, то $D\xi = 0$, а $E\xi = J_h$. Моделируя подходящую цепь Маркова на ЭВМ, получают статистическую оценку линейных функционалов от решения интегрального уравнения второго рода. Это даёт возможность и локальной оценки решения на основе представления: $\varphi(x) = (\varphi, h_x) + f(x)$, где $h_x(x') = k(x', x)$. Методом Монте-Карло оценка первого собственного значения интегрального оператора осуществляется интерационального методом на основе соотношения $M[Q_n h(x_n)] = (K^n f, h)$. Все рассмотренные результаты почти автоматически распространяются на системы линейных алгебраических уравнений вида $x + H_x = h$. Решение дифференциальных уравнений осуществляется методом Монте-Карло на базе соответствующих интегральных соотношений.

Способ усреднения подынтегральной функции

В качестве оценки определённого интеграла $I = \int_a^b \varphi(x)dx$ принимают

$$I_1^* = (b-a)\frac{\sum_{i=1}^{n}\varphi(x_i)}{n},$$

где n – число испытаний; x_i - возможные значения случайной величины X, распределённой равномерно в интервале интегрирования (a,b), их разыгрывают по формуле $x_i = a + (b-a)r_i$, где r_i - случайное число.

Дисперсия усредняемой функции $\varphi(X)$ равна

$$\sigma^2 = \int_a^b \varphi^2(x)f(x)dx - \{M[\varphi(X)]\}^2,$$

где $M[\varphi(X)] = \int_a^b \varphi(x)f(x)dx$, $f(x) = \frac{1}{b-a}$. Если точное значение дисперсии вычислить трудно или невозможно, то находят выборочную дисперсию (при n>30) $D_B = \sum\frac{u_i^2}{n} - [\sum\frac{u_i}{n}]^2$, или исправленную дисперсию (при n<30) $s^2 = \frac{1}{n-1}[\sum u_i^2 - \frac{[\sum u_i]^2}{n}]$, где $u_i = \varphi(x_i)$.

Эти формулы для вычисления дисперсии применяют и при других способах интегрирования, когда усредняемая функция не совпадает с подынтегральной функцией.

В качестве оценки интеграла $I = \iint_D f(x,y)dxdy$, где область интегрирования D принадлежит единичному квадрату $(0 \le x \le 1$, $0 \le y \le 1)$, принимают

$$I_1^* = S\frac{\sum_{i=1}^{N} f(x_i, y_i)}{N}, \quad (*)$$

где S – площадь области интегрирования; N – число случайных точек (x_i, y_i), принадлежащих области интегрирования.

Если вычислить площадь S трудно, то в качестве её оценки можно принять $S^* = \frac{N}{n}$; в этом случае формула (*) имеет вид

$$I_1^* = \frac{\sum_{i=1}^{N} f(x_i, y_i)}{n},$$

где n – число испытаний.

В качестве оценки интеграла $I = \iiint\limits_V f(x,y,z)dxdydz$, где область интегрирования V принадлежит единичному кубу $(0 \le x \le 1$, $0 \le y \le 1$, $0 \le z \le 1)$, принимают $I_1^* = V \frac{\sum_{i=1}^{N} f(x_i, y_i, z_i)}{N}$, где V – объём области интегрирования, N – число случайных точек (x_i, y_i, z_i), принадлежащих области интегрирования.

Если вычислить объём трудно, то в качестве его оценки можно принять $V^* = \frac{N}{n}$, в этом случае формула (**) имеет вид $I_1^* = \frac{\sum_{i=1}^{N} f(x_i, y_i, z_i)}{n}$, где n – число испытаний.

Задача: найти оценку I_1^* определённого интеграла $I = \int_1^3 (x+1)dx$.

Решение. Используем формулу $I_1^* = (b-a)\frac{\sum_{i=1}^{n} \varphi(x_i)}{n}$. По условию, a=1, b=3, $\varphi(x) = x+1$. Примем для простоты число испытаний n=10.Тогда оценка $I_1^* = (3-1)\frac{\sum_{i=1}^{10} (x_i+1)}{10} = 2\frac{\sum_{i=1}^{10} (x_i+1)}{10}$, где возможные значения x_i разыгрывается по формуле $x_i = a + (b-a)r_i = 1 + (3-1)r_i = 1 + 2r_i$.

Результаты десяти испытаний приведены в таблице 1.

Случайные числа r_i взяты из таблицы приложения.

Таблица 1.

Номер i	r_i	$x_i = 1 + 2r_i$	$\varphi(x_i) = x_i + 1$
1	0,100	1,200	2,200
2	0,973	2,946	3,946
3	0,253	1,506	2,506
4	0,376	1,752	2,752
5	0,520	2,040	3,040
6	0,135	1,270	2,270
7	0,863	2,726	3,726
8	0,467	1,934	2,934
9	0,354	1,708	2,708
10	0,876	2,752	3,752

[5]

Дальнейшее развитие и современность

В 1950-х годах метод использовался для расчётов при разработке водородной бомбы. Основные заслуги в развитии метода в это время принадлежат сотрудникам лабораторий ВВС США и корпорации RAND.

В 1970-х годах в новой области математики — теории вычислительной сложности было показано, что существует класс задач, сложность (количество вычислений, необходимых для получения точного ответа) которых растёт с размерностью задачи экспоненциально. Иногда можно, пожертвовав точностью, найти алгоритм, сложность которого растёт медленнее, но есть большое количество задач, для которого этого нельзя сделать (например, задача определения объёма выпуклого тела в n-мерном евклидовом пространстве) и метод Монте-Карло является единственной возможностью для получения достаточно точного ответа за приемлемое время.

В настоящее время основные усилия исследователей направлены на создание эффективных Монте-Карло алгоритмов различных физических, химических и социальных процессов для параллельных вычислительных систем.**[6]**

Использование метода

Вычислить площадь фигуры.

Для простейших фигур: прямоугольник, многоугольник, круг и т.п. все просто – в известные формулы надо подставить исходные данные.

Но как быть, если фигура имеет сложную форму?

- Задача. *Дана фигура сложной формы. Вычислить ее площадь.*

Возможные модели для решения задачи:

- *С помощью палетки* (на фигуру накладывается клетчатая прозрачная бумага (палетка) и подсчитывается количество квадратиков, попавших в фигуру. Предполагается, что чем меньше клетки, тем точнее будет результат независимо от того, каким образом наложить палетку на фигуру;
- *«физическая» модель*: скопировать фигуру на картон, аккуратно вырезать его, взвесить и поделить на вес единичного квадрата из этого же картона;
- с помощью *интегралов.*

Однако эти модели не для расчетов на ЭВМ.

- Математическая модель метода Монте-Карло

Поместим данную фигуру в квадрат. Будем наугад (случайным образом) бросать точки в этот квадрат. Естественно предположить, что чем больше площадь фигуры, тем чаще в нее будут попадать точки.

Представим себе квадратный дворик и в нем детскую площадку. Ясно, что во время снегопада количество снежинок, попавших на детскую площадку, пропорционально ее площади.

Т.о., можно сделать допущение: при большом числе точек, наугад выбранных внутри квадрата, доля точек, содержащихся в данной фигуре, приближенно равна отношению площади этой фигуры к площади квадрата.

Исходные данные:

F – данная фигура.

a – сторона квадрата, содержащего фигуру F.

N – количество точек, которые мы будем случайным образом выбирать внутри квадрата

Результат:

S – площадь фигуры F.

M – число точек, которые содержатся в фигуре F.

Связи между исходными данными и результатами:

- $S \approx a^2 M/N$ $\left[\frac{S}{a^2} \approx \frac{M}{N}\right]$
- математические соотношения позволяющие определить попала ли точка в фигуру ***F***;

- Проверим данную модель для приближенного нахождения площади круга радиуса R.

Формула площади круга: $S = \pi R^2$. Однако, эта формула дана нам без доказательства. Проверим ее с помощью ЭВМ.

Ввести N – количество точек

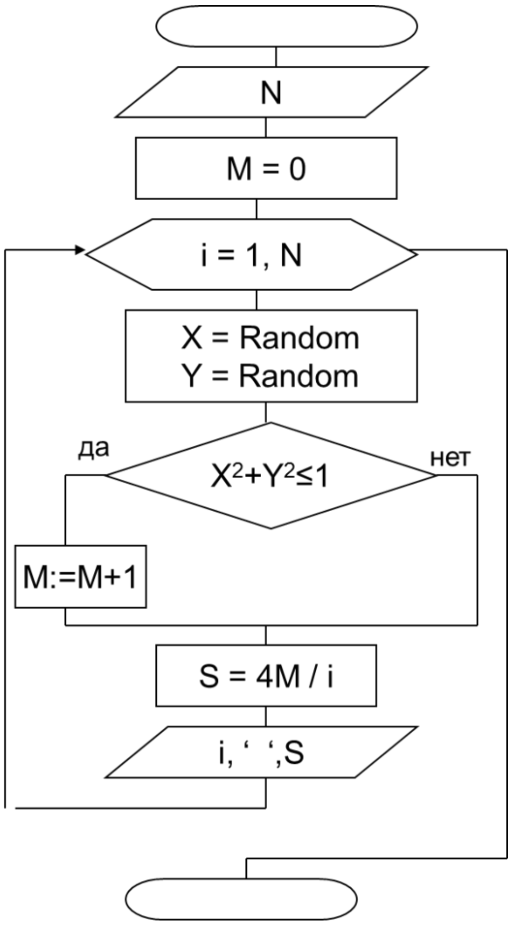
Пусть $\boldsymbol{R = 1;\ a = 2}$.

Выбрать точку – значит задать ее координаты: числа $\boldsymbol{x}$ и $\boldsymbol{y}$.

Точка принадлежит квадрату, если $\boldsymbol{-1 \le x \le 1}$ ***и*** $\boldsymbol{-1 \le y \le 1}$.

Теорема Пифагора “подсказывает”: если $\boldsymbol{x^2 + y^2 \le 1}$, то точка попадет в круг $\boldsymbol{F}$, иначе она вне круга.

Это и есть математическое соотношение, позволяющее для каждой точки определить, лежит ли она в $\boldsymbol{F}$.



-
- Количество точек в круге *M = 0*
- *i* будет последовательно принимать значения *1, 2, 3, …, N*
- Стандартная функция Random вырабатывает случайные числа в интервале *[0,1)*
- Если точка попадет в круг – значение *M* увеличим на *1*
- Присвоим *S* примерное значение площади.
- Для каждого *i* печатаем очередное *S*

• Количество точек (N)

Число точек надо брать достаточно большим – не менее 300

• Точность результатов вычислений

Зависит не только от того, является модель вероятностной или нет, - это зависит и от точности исходных данных, и от точности вычислений.

Таким образом, неточные результаты могут получиться и при вычислениях по тем моделям, в которых случайность вроде бы не присутствует.**[7]**

Факторизация чисел

Факториза́цией натурального числа называется его разложение в произведение простых множителей. Существование и единственность (с точностью до порядка следования множителей) такого разложения следует из основной теоремы арифметики.**[8]**

Рассмотрим один из способов факторизации чисел, предложенный Поллардом. Они легки в использовании и подойдут для больших составных чисел. Время работы метода зависит корня квадратного от данного числа m и минимального простого числа, делящего m. То есть в худшем случае когда число m есть произведение 2 простых чисел примерно одного порядка, число m раскладывается на множители за время, равное корню 4 степени из m. Алгоритм очень прост, его запись короче объяснения. Слово Монте-Карло в название метода присутствует из-за того что алгоритм зависит от случайного выбора начального числа.

Рассмотрим отображение f кольца Zm в Zm:

f: Zm --> Zm

f(x) = x^2 + 1 (mod m)

Выберем случайным образом элемент b_0 кольца Zm. Рассмотрим бесконечную последовательность элементов кольца Zm:

b_0, b_1 = f(b_0), b_2 = f(b_1), b_3 = f(b_2), ... (2)

Последовательность представляет собой орбиту элемента b_0 при отображении f. Поскольку все элементы последовательности принадлежат конечному множеству, последовательность циклическая -- точнее, она содержит начальный апериодический отрезок и далее бесконечно повторяющийся период.

Пусть p -- делитель числа m. Рассмотрим элементы последовательности (2) по модулю p (т.е. образы элементов b_i при каноническом эпиморфизме Zm --> Zp):

c_0 == b_0 (mod p), c_1 == b_1 (mod p), c_2 == b_2 (mod p), ... (3)

Так как в Zp меньше элементов, чем в Zm, то с большой вероятностью период последовательности (3) меньше, чем период последовательности (2). Следовательно, найдется пара индексов i, j такая, что

c_i = c_j, b_i != b_j.

Это означает, что

b_i == b_j (mod p), b_i != b_j (mod m).

Отсюда вытекает, что (b_i - b_j) делится на p, но не делится на m. Следовательно, НОД(b_i - b_j, m) нетривиален, и нам удалось разложить m на множители.

Итак, алгоритм Полларда сводится к поиску цикла в бесконечной рекурсивной последовательности, состоящей из элементов конечного множества. При этом вместо того, чтобы сравнивать между собой два элемента, мы вычисляем наибольший общий делитель их разности и числа m. Алгоритм завершается, когда наибольший общий делитель нетривиален.

Можно предложить 2 способа решения задачи поиска цикла в последовательности. **[9]**

Я рассмотрю один из них. представив алгоритм решения в своей работе, его можно перевести в программу на любом языке программирования для увеличения скорости расчётов

Алгоритм:

- Дано число m, нам известно, что оно составное
- Случайно выберем число a_0,условиz для этого числа таковs: $a_0>1$,

$a_0<m$

- Вычислим число b_0:

  b_0=ост($(a_0^2+1)/m$)
- Найдём НОД, которому присвоим обозначение d_0
- Если $d_0=1$, то повторим алгоритм
- Если $d_0\neq1$, то d_0 это множитель

Данный алгоритм можно представить в программе. Приведу пример рабочей программы на языке Pascal:

Free pascal

ВВОД ДАННЫХ

```
533
```

ВЫВОД ОШИБКИ КОМПИЛЯЦИЯ СООБЩЕНИЯ

```
введите составное число m
шаг=0 a0=295 b0=147 d=1
шаг=1 a0=147 b0=420 d=13
Множитель d0=13
```

Главная Файл Выполнить [F9] Pascal online

main.pas

```
program Pollard;
//факторизация составного числа m методом Монте-Карло( алгоритм Полларда)
var m, a0, b0, d0, max_count, count: integer;
function gcd(m, n: integer): integer;
    var modulo: integer;
    begin
        modulo := m mod n;
        if modulo = 0 then
            gcd := n
        else
            gcd := gcd (n, modulo)
    end;
begin
  max_count:=10000;
  count:=0;
  writeln('введите составное число m');
  read(m);
  randomize;
  a0:=random(m-1)+1;
  b0:= (a0*a0+1) mod m;
  d0 := gcd(abs(b0 - a0), m);
  writeln('шаг=', count, ' a0=', a0, ' b0=', b0, ' d=', d0 );
  while d0=1 do
  begin
    a0:= (a0*a0+1) mod m;
    b0:= (b0*b0+1) mod m;
    b0:= (b0*b0+1) mod m;
    d0:=gcd(abs(b0-a0), m);
    count:=count+1;
    writeln('шаг=', count, ' a0=', a0, ' b0=', b0, ' d=', d0 );
  end;
  writeln('Множитель d0=',d0);
end.
```

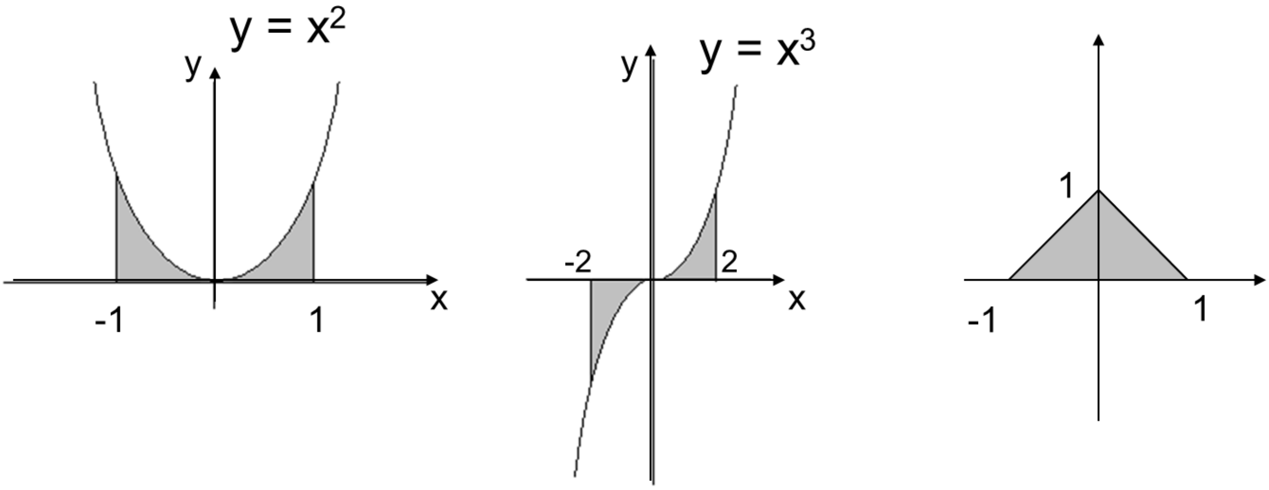
Минусом данной программы является то, что:

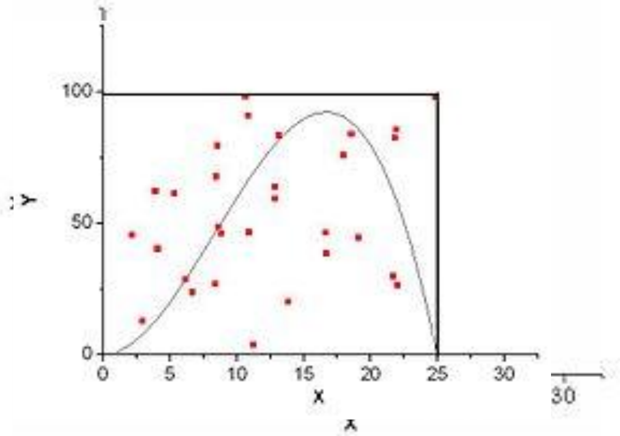
- Не факторизует числа состоящие из произведения простых чисел в степени и их комбинаций. Например, число как 100(2*2*5*5), однако этот недочёт можно исправить, дополнив программу.

Решение интегралов

Алгоритм:

1. Выбрать нижний предел A
2. Выбрать верхний предел B
3. Ввести количество точек N
4. Повторять от 0 до N-1
5. Точка X случайная в интервале [A,B]
6. $S_n=S_0+f(x)$; S_0- текущее значение площади(с каждым разом она будет обновляться)
7. Повторить 4, 5, 6, 7 пункты
8. $S=((B-A)*S_n)/N$ – конечная площадь под кривой





Заключение

Метод Монте-Карло используется очень часто в задачах, которые при решении обычными методами неэффективны и затратны по времени. Однозначно можно выделить положительные стороны этого метода:

- Подходит для решения задач с помощью современных компьютеров.
- Он приводит к выполнимости задачи даже в сложном случае, когда классическое решение не даёт приемлемого времени решения.
- Его легко применять без предварительного анализа задачи.
- Метод универсален.

Также стоит упомянуть о недостатках данного метода, а именно:

- Границы ошибки не определены точно, но включают некую случайность. Это, однако, более психологическая, чем реальная, трудность.
- Необходимо иметь «хороший» генератор случайных чисел.
- Статическая погрешность убывает медленно.
- Необходимость иметь случайные числа.

Таким образом, я ознакомилась с историей метода, его принципами. Могу подвести итог того, что работать с этим методом можно и в будущем, так как раскрытие его потенциала в рамках школьной программы весьма скудно из-за его сложности и наличия множества терминов, которые выходят за рамки средней школы. Применять его можно: при поиске площадей фигур. факторизации чисел, нахождении интегралов. Цель своей работы я считаю достигнутой.

Литература

[1]https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%9C%D0%BE%D0%BD%D1%82%D0%B5-%D0%9A%D0%B0%D1%80%D0%BB%D0%BE

[2]https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%BE%D1%85%D0%B0%D1%81%D1%82%D0%B8%D1%87%D0%BD%D0%BE%D1%81%D1%82%D1%8C

[3] https://math.semestr.ru/math/diffur.php

[4] machinelearning.ru/wiki/index.php?title=Метод_Монте-Карло

[5] Курсовая работа Зубанова М. А., студента 3 курса очного отделения физико-математического факультета АГПИ имени А.П.Гайдара Кафедра математического анализа

[6] N. Metropolis, S. Ulam The Monte Carlo Method, — J. Amer. statistical assoc. 1949 **44** № 247 335—341

[7] А.Г. Гейн, и др. Информатика. Учебник для 8-9 классов.Москва, «Просвещение», 1997

[8]https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D1%86%D0%B5%D0%BB%D1%8B%D1%85_%D1%87%D0%B8%D1%81%D0%B5%D0%BB

[9] http://algolist.ru/maths/teornum/factor/monte.php