ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

ЖУРНАЛ ПРАКТИКИ

Студента 2 курса Туманова Георгия Артуровича
(Фамилия, имя. отчество)

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 804 «Теория вероятностей и компьютерное моделирование»

Учебная группа М8О-201Б-18

Направление 01.03.02. Прикладная математика и информатика
(шифр) *(название направления)*

Вид практики учебная (вычислительная)
(учебная (вычислительная, исследовательская), производственная (преддипломная))
в Московском авиационном институте (НИУ)
(наименование предприятия, учреждения, организации)

Руководитель практики от МАИ Зайцева О.Б. _____
(ФИО) (Подпись)

Туманов Г.А. / _____ / “12” июля 2020 г.
(ФИО) (подпись студента) (дата)

Москва 2020

1. Место и сроки проведения практики

Дата начала практики *“29” июня 2020 г.*

Дата окончания практики *“12” июля 2020 г.*

Наименование предприятия МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Название структурного подразделения) кафедра 804

2. Инструктаж по технике безопасности

Зайцева О.Б. /_____/ “29” июня 2020 г.

(подпись проводившего) *(дата проведения)*

3. Индивидуальное задание студенту

Тема: Моделирование случайных событий и дискретных случайных величин (в пакете MATLAB)

Вариант 7.

В урне m белых и r чёрных шаров. Из урны извлекают l шаров (с возвращением).

ξ - количество чёрных шаров среди них.

1. Вычислить $\mathbf{P}\{\xi = k\}$, $k \in \mathbb{N} \cup \{0\}$.
2. Найти вероятность события $A = \{\xi$ — нечётное$\}$.
3. Провести моделирование эксперимента и получить элементарное событие. Повторить эксперимент n раз. После каждой итерации вычислять частоту события A. ($n \in \{100; 1600; 10000\}$)
4. Для каждого их 3х значений n построить график зависимости частоты события A от количества экспериментов.
5. По графику, построенному в п. 4, определить при каком n частота не будет отличаться от вероятности более чем на $\varepsilon \in \{0{,}1; 0{,}05; 0{,}025; 0{,}01\}$ в течение последующих $N \in \{10; 50; 100\}$ итераций.

Предусмотреть возможность программного задания параметров эксперимента. При моделировании в системе matlab используйте функцию rand. Также изучите возможность использования специальных функций, моделирующих распределение из заданного варианта. Сравните скорость работы программы при использовании функции rand и специальной функции.

4. План выполнения индивидуального задания

1. Ознакомление с местом прохождения практики, средствами обеспечения безопасной работы
2. Составление рабочего плана и графика выполнения задания
3. Поиск и анализ литературных источников по тематике практики
4. Основные принципы работы в MATLAB.
5. Научная графика в MATLAB.
6. Моделирование случайных величин (функция rand)
7. Моделирование распределений с помощью специальных функций
8. Решение индивидуальной задачи
9. Анализ полученных результатов
10. Подготовка к защите практики. Оформление отчета.
11. Защита результатов практики

Руководитель практики от МАИ: __Зайцева О.Б.___ /_____/

(Фамилия, имя, отчество) (Подпись)

___Туманов Г.А._____ /_____/ “29” июня 2020 г.

(ФИО) (подпись студента) (дата)

5. Отзыв руководителя практики

Поставленная задача выполнена полностью. Получено аналитическое решение задачи, которое затем было подтверждено с помощью выборочных оценок. Написана программа для построения выборочных оценок. Материалы, изложенные в отчёте студента, полностью соответствуют индивидуальному заданию. Работа студента Туманова Г. заслуживает оценки «отлично»

Руководитель _____Иванов С.В._____ /_____/

(Фамилия, имя, отчество) (Подпись)

“12” *июля* *2020 г.*

Отчет студента

Решение задачи: теория, формулы, значения для контрольного примера, программный код, графики, выводы (все по заданию)

Задание:

Вариант 7. В урне m белых и r чёрных шаров. Из урны извлекают l шаров (с возвращением). ξ — количество чёрных шаров среди них.

1. Вычислить P(ξ=k), k=0,1,2,…

2. Найти вероятность события A={ξ-нечетное}

3. Провести моделирование эксперимента и получить элементарное событие. Повторить эксперимент n раз. После каждой итерации вычислить частоту события A. (n=100,1600,10000).

4. Для каждого из 3-х значений n построить график зависимости частоты события A от количества экспериментов

5. По графику, построенному в пункте 4, определить при каком n частота не будет отличаться от вероятности не более чем на (0.1,0.05,0.025,0.01) в течение последующих N=10,50,100 итераци

Решение:

1) Рассмотрим серию из l опытов. Опыт считается успешным, если вытащенный шар чёрный. Так как шары вытаскивают с возвращением, данные опыты не зависимы. Получаем схему Бернулли из l опытов. Вероятность вытащить чёрный шар в каждом опыте равна $\frac{r}{m+r}$. Следовательно, ξ распределена по биномиальному закону, а значит:
$$P(\xi = k) = C_l^k \left(\frac{r}{m+r}\right)^k \left(1 - \frac{r}{m+r}\right)^{l-k} = C_l^k \left(\frac{r}{m+r}\right)^k \left(\frac{m}{m+r}\right)^{l-k} = \frac{C_l^k r^k m^{l-k}}{(m+r)^l}$$
2) Вероятность события A равна:
$$P(A) = \sum_{k=0}^{+\infty} P(\xi = 2k+1)$$
Но число отличных от нуля слагаемых конечно. Отбросим все слагаемые, равные нулю (это слагаемые, для которых $2k+1 > l$). Получим:
$$P(A) = \sum_{k=0}^{\left\lfloor \frac{l-1}{2} \right\rfloor} P(\xi = 2k+1) = \sum_{k=0}^{\left\lfloor \frac{l-1}{2} \right\rfloor} \frac{C_l^{2k+1} r^{2k+1} m^{l-2k-1}}{(m+r)^l}$$
Вычислим вероятность для $m = 4$, $r = 6$ и $l = 3$:
$$P(A) = \sum_{k=0}^{\left\lfloor \frac{l-1}{2} \right\rfloor} \frac{C_l^{2k+1} r^{2k+1} m^{l-2k-1}}{(m+r)^l} = \sum_{k=0}^{1} \frac{C_3^{2k+1} 6^{2k+1} 4^{3-2k-1}}{(10)^3} =$$
$$= \frac{C_3^1 6^1 4^{3-1}}{1000} + \frac{C_3^3 6^3 4^{3-3}}{1000} = 0.504$$

3) Процесс будет смоделирован на языке JavaScript. Благодаря технологиям HTML5 можно достаточно легко построить все необходимые графики. Также это позволяет моделировать процесс прямо на веб-странице без скачивания и компилирования программы.

   Программа моделирует случайное событие A, рисует график зависимости частоты A от числа проведённых опытов, вычисляет вероятность P(A), а также находит n для пятого

пункта и выделяет на графике область от n до n+N. Если n не существует, программа пишет n=undefined.

Файл index.html представляет собой саму веб-страницу, на которой находятся график и элементы управления. plot.js содержит код, необходимый для рисования графика и считывания данных, введённых пользователем. Все необходимые вычисления вероятностей находятся в файле math.js.

ξ моделируется следующим образом. Сначала мы вычисляем вероятность p вытащить чёрный шар. После этого мы l раз увеличиваем ξ с вероятностью p. Это делается с помощью генератора псевдослучайных величин от 0 до 1, встроенного в JS.

```
//get value of ksi
function getKsi(m, r, l)
{
	let p = r / (r + m); //probability of black ball

	let ksi = 0; //value of ksi
	for (let i = 0; i < l; i++)
	{
		if (Math.random() < p) ksi++;
	}
	return ksi;
}
```

Смоделировав ξ, вычисляем частоту A. Мы параллельно считаем число успехов success. Если сгенерированное ξ нечётно, мы увеличиваем success на единицу. Тогда частота A равна success, делённое на номер опыта:

```
//carry out n experiments
function experiment(m, r, l, n) //generate array [v1, v2, ..., vn]
{
	let successes = 0; //ammount of successfull experiments
	let v = []; //array of frequences

	for (let i = 1; i <= n; i++)
	{
		//we need to add to successes variable 1 if A and 0 if not A
		//A means ksi is odd
		//so we just add (ksi mod 2) to successes each time
		successes += getKsi(m, r, l) % 2;
		v.push(successes / i);
	}
	return v;
}
```

Теперь вычисляем точное значение вероятности P(A). Мы переносим на JS формулы, полученные в пунктах 1 и 2, а также реализуем вычисление C_n^k:

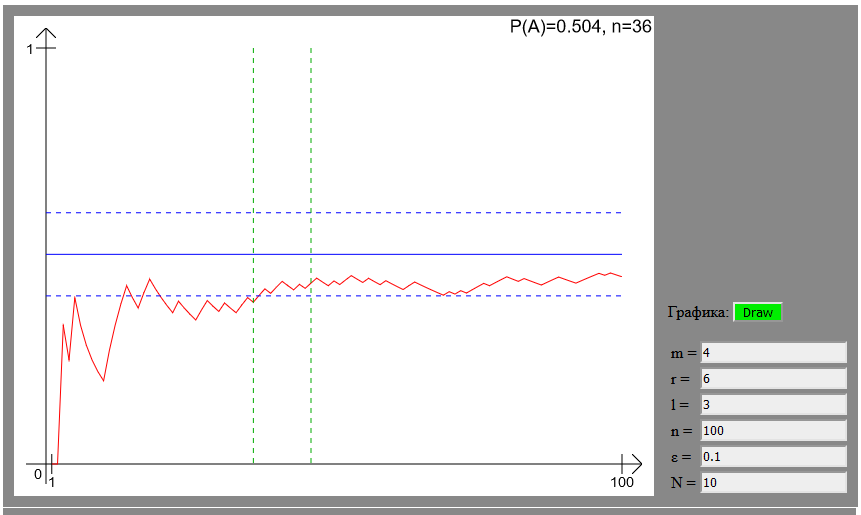
```
//now we need to find propper value of probability
//find C(k, n)
function C(k, n)
{
	let c = 1;
	for (let i = 1; i <= k; i++) c *= (n - k + i) / i;
	return c;
}
//find P(ksi = k)
function Pk(m, r, l, k)
{
	if (k > l) return 0;
	return C(k, l) * Math.pow(r, k) * Math.pow(m, l - k) / Math.pow(m + r, l);
}
//find P(A)
function probability(m, r, l)
{
```

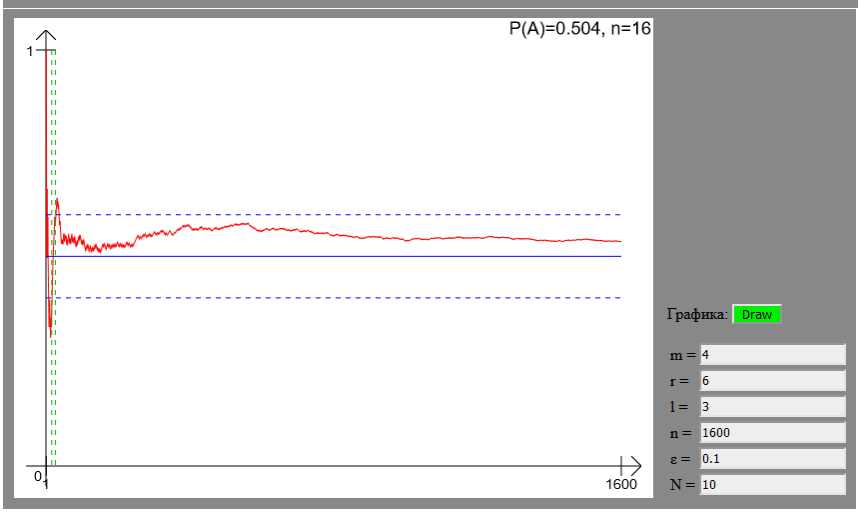
```
        let kmax = Math.floor((l - 1) / 2); //upper limit of summing

        let p = 0; //probability
        for (let k = 0; k <= kmax; k++) p += Pk(m, r, l, 2 * k + 1);
        return p;
}
```

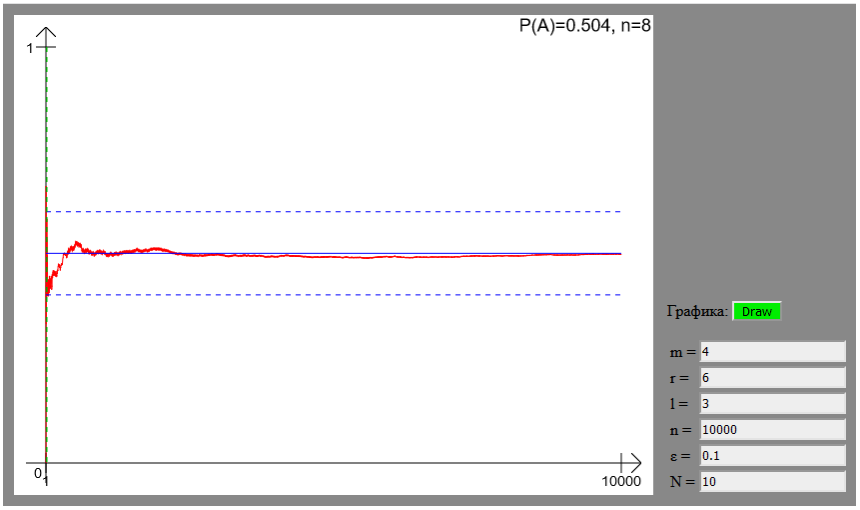
И наконец, найдём n, начиная с которого следующие N шагов разность частоты A и точной вероятности P(A) не превосходит ε. Мы перебираем все n и проверяем, удовлетворяет ли оно необходимому условию. Если да, то возвращаем первое найденное n, иначе возвращаем undefined:

```
//find n, from which...
function checkN(n, N, v, p, eps)
{
        for (let i = n; i < n + N; i++)
                if (Math.abs(v[i] - p) > eps) return false;
        return true;
}
function findN(N, v, p, eps)
{
        for (let i = 0; i < v.length - N; i++)
                if (checkN(i, N, v, p, eps)) return i;
        return undefined;
}
```

4) Построим графики для n=100, n=1600 и n=10000:

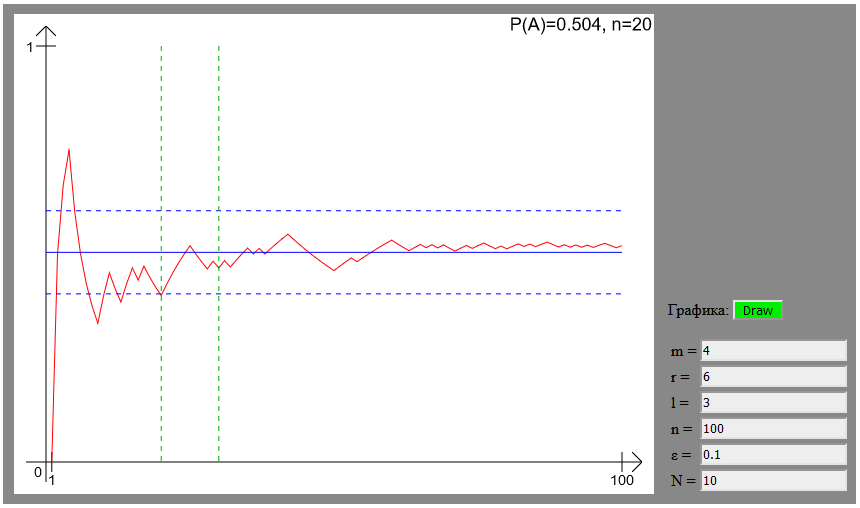




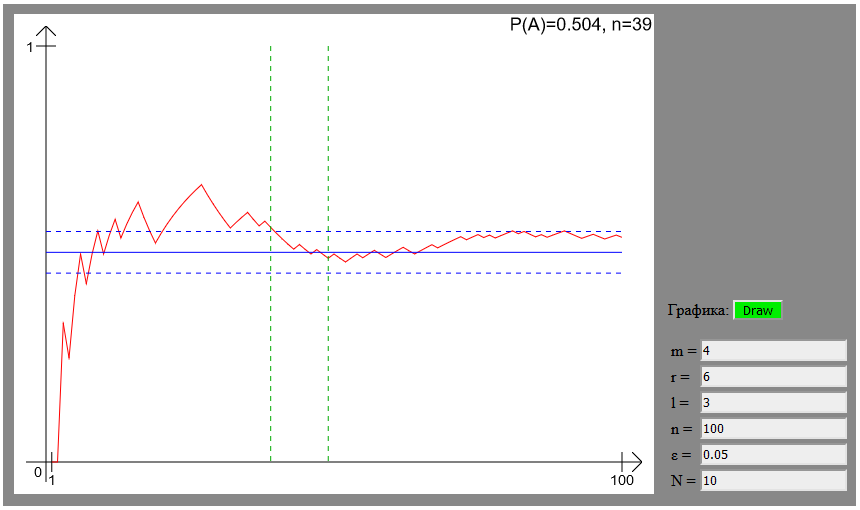


5) Найдём n при разных N и ε:

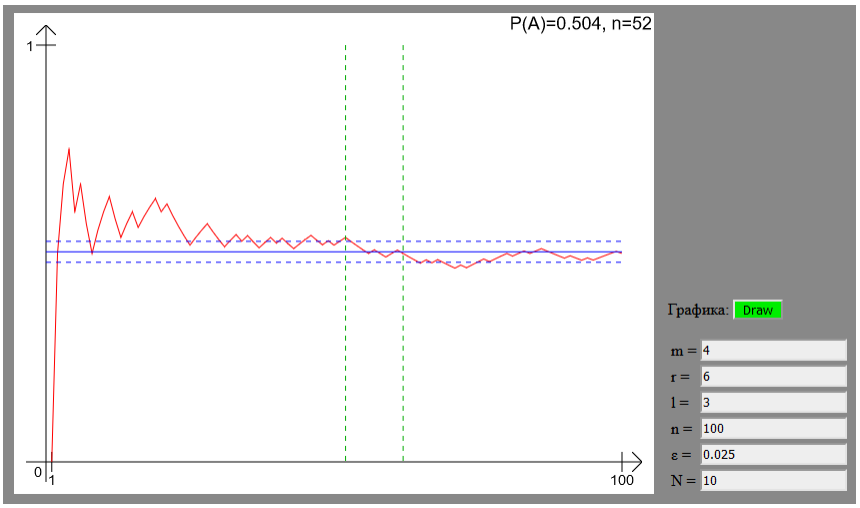
N=10, ε=0.1



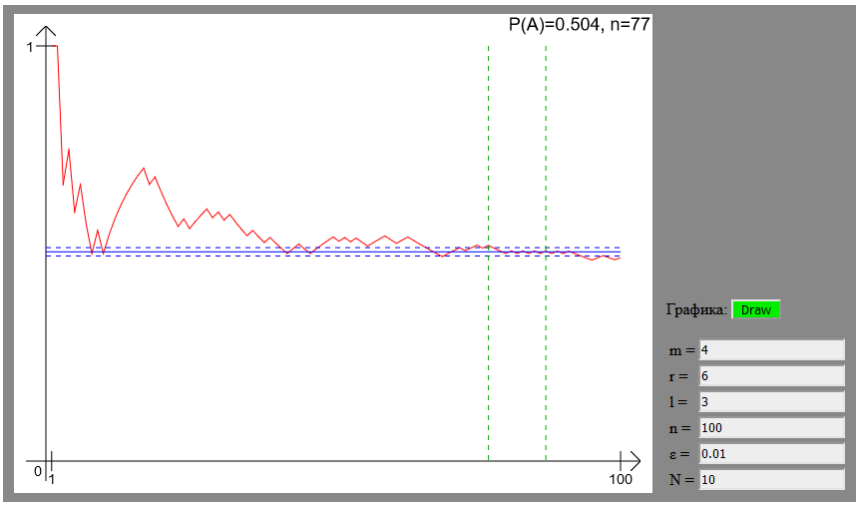
N=10, ε=0.05



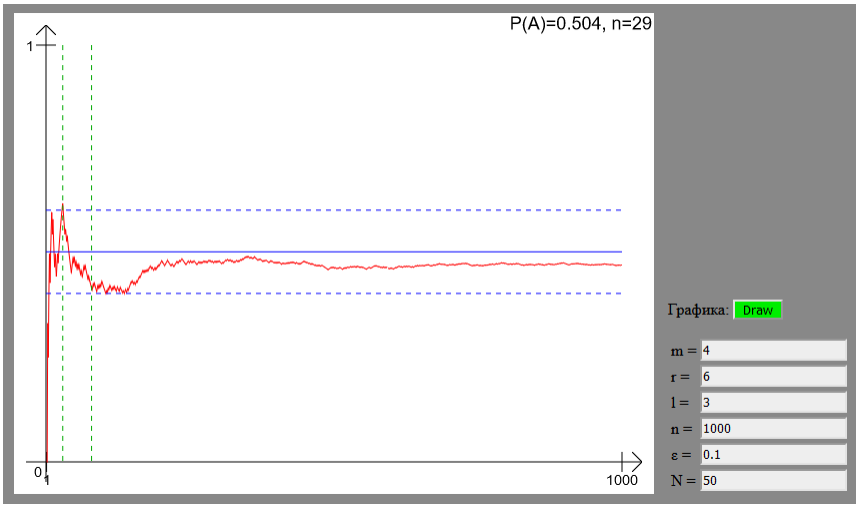
N=10, ε=0.025



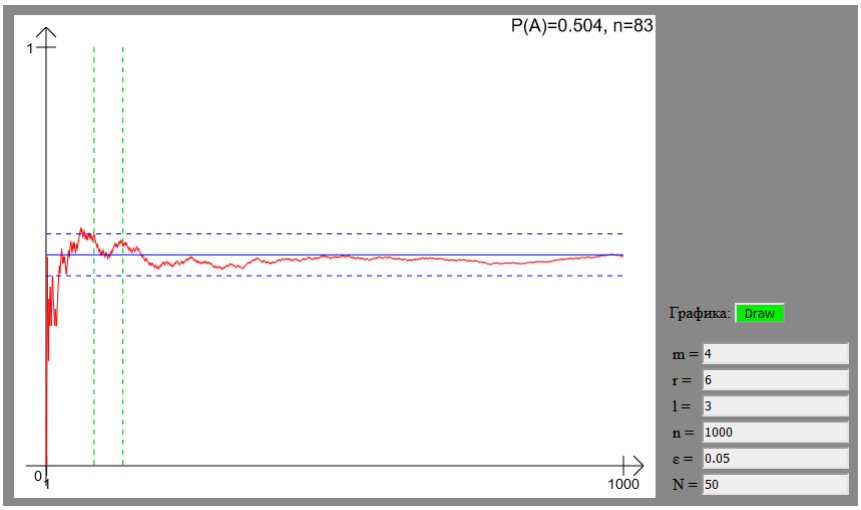
N=10, ε=0.01



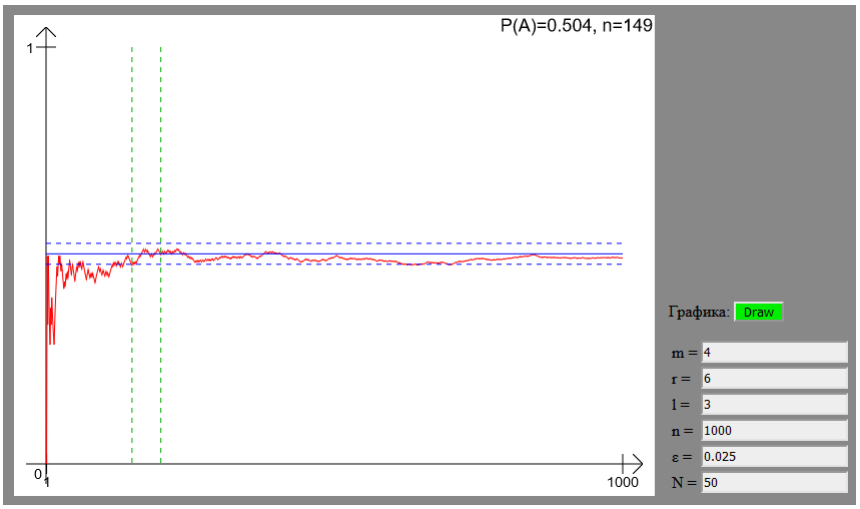
N=50, ε=0.1



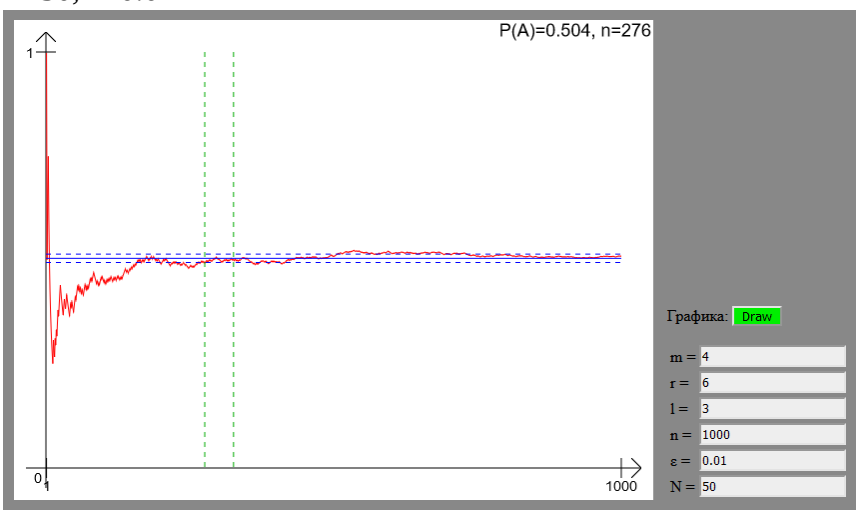
N=50, ε=0.05



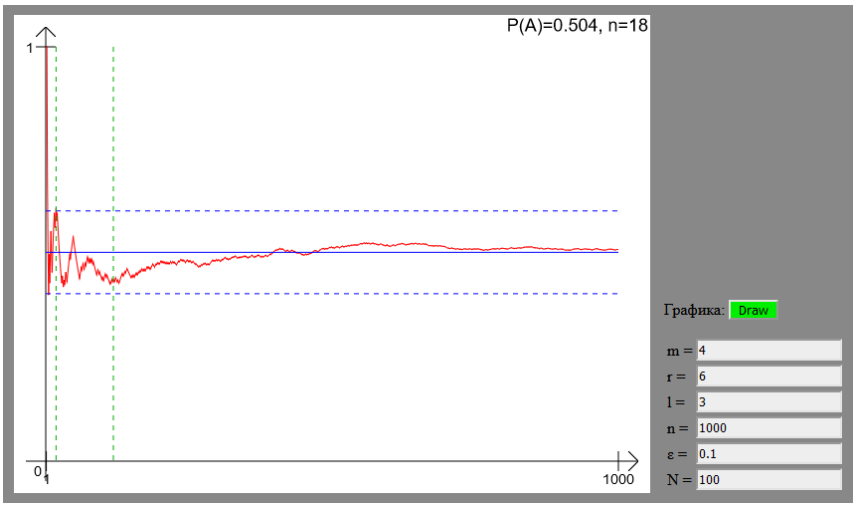
N=50, ε=0.025



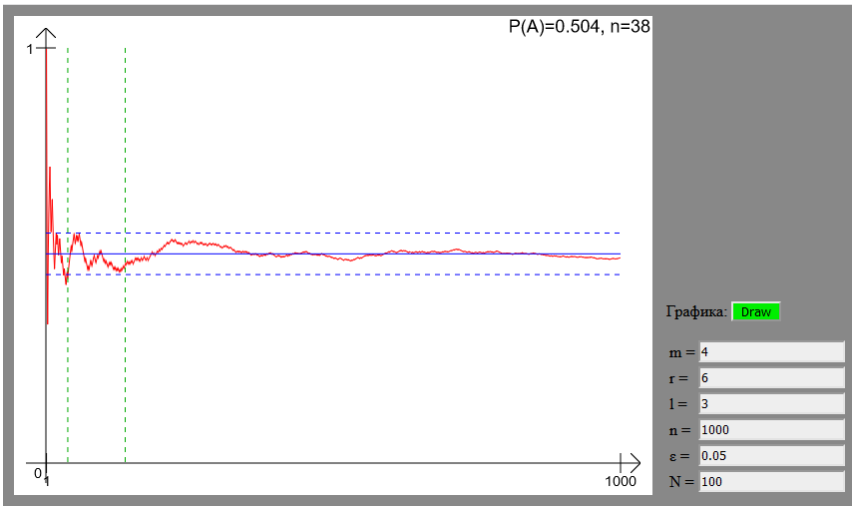
N=50, ε=0.01



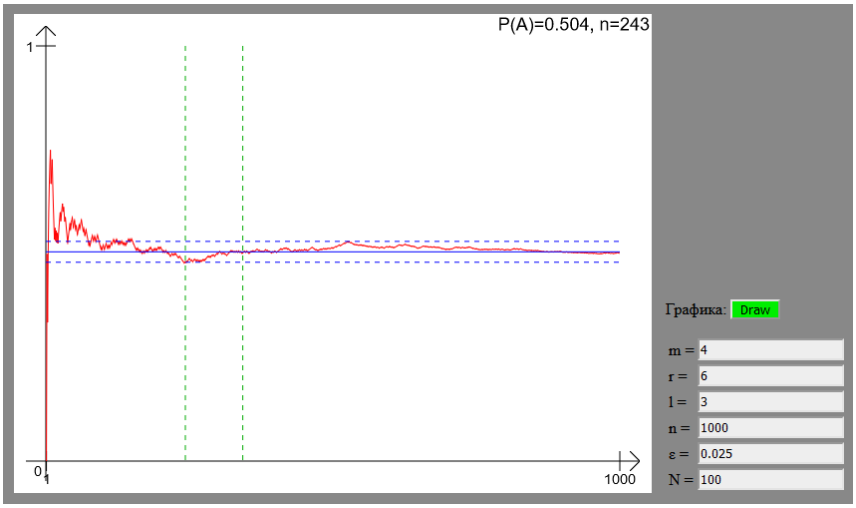
N=100, ε=0.1



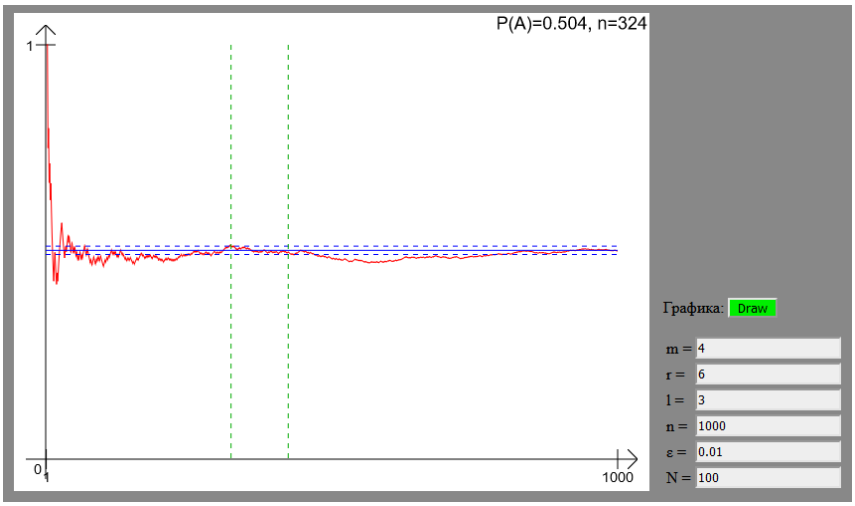
N=100, ε=0.05



N=100, ε=0.025



N=100, ε=0.01



6) Вывод: были смоделированы случайные события (ξ=k) и событие A, заключающееся в том, что ξ нечётное, и дискретная случайная величина ξ, равная числу чёрных шаров, вытащенных из урны, на ЭВМ с помощью языка программирования JavaScript и технологии HTML5. Были построены графики зависимости частоты события A от числа опытов. Также программная реализация позволяет пользователю задавать величины m, r, l, n (число опытов), ε, N и находить вероятность P(A) и такое n, что для всех $k \in [n, n+N]$ выполняется $|\nu(k) - P(A)| < \varepsilon$.

Задача 4 (аналитически):

Пусть P(n) – искомая вероятность. Тогда

$$P(n) = \frac{a_n}{b_n}$$

где a_n – число благоприятных исходов, а b_n – общее число исходов. b_n находится легко:

$$b_n = 2^{2n}$$

И действительно, совершается 2n броска, в каждом из которых может быть 2 варианта: орёл или решка.

Представим броски в виде скобочной записи: открытая скобка означает выпадение орла, а закрытая – решки. Тогда a_n равно числу правильных скобочных последовательностей, что равно числам Каталана:

$$a_n = \frac{1}{n+1} C_{2n}^{n} = \frac{(2n)!}{n!\,n!\,(n+1)}$$

Итого:

$$P(n) = \frac{(2n)!}{2^{2n} n!\,n!\,(n+1)}$$

Задача 5 (аналитически):

Пусть P(n) – искомая вероятность. Тогда

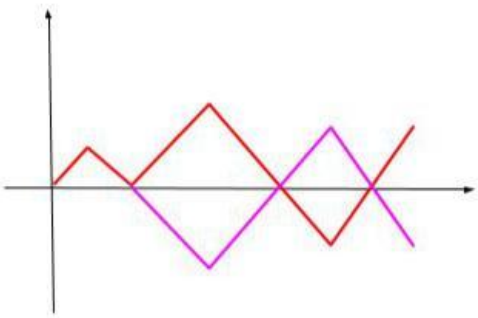
$$P(n) = \frac{a_n}{b_n}$$

где a_n – число благоприятных исходов, а b_n – общее число исходов. b_n находится легко:

$$b_n = 2^{2n}$$

И действительно, совершается 2n броска, в каждом из которых может быть 2 варианта: орёл или решка.

Найдём a_n. Представим задачу в виде графика. Мы идём вверх, если выпадает орёл, и вниз, если решка. Очевидно, первым должен выпасть орёл, потому мы пропускаем первый шаг.



Так как мы отбросили первого орла, число бросков стало нечётным. Значит, график может закончиться либо строго над осью абсцисс, либо под осью. Всего графиков 2^{2n-1}. Нам нужно найти число графиков, заканчивающихся над осью и никогда её не пересекающих.

Рассмотрим графики, пересекающие ось. Они также заканчиваются либо под осью, либо над осью. Однако, если заканчивающийся над осью график отразить по вертикали после его первой точки пересечения с осью, то мы получим график, заканчивающийся под осью. Таким образом, мы получаем биекцию между этими графиками, а значит их количество равно. Следовательно, в сумме число этих графиков равно:

$$\sum_{k=0}^{n-1} C_{2n-1}^{k} + \sum_{k=n+1}^{2n-1} C_{2n-1}^{k}$$

Если убрать графики, пересекающие ось, то останутся только графики, заканчивающиеся сверху и не пересекающие ось. Значит, число этих графиков равно:

$$a_n = 2^{2n-1} - \sum_{k=0}^{n-1} C_{2n-1}^{k} - \sum_{k=n+1}^{2n-1} C_{2n-1}^{k} = \sum_{k=0}^{2n-1} C_{2n-1}^{k} - \sum_{k=0}^{n-1} C_{2n-1}^{k} - \sum_{k=n+1}^{2n-1} C_{2n-1}^{k} = C_{2n-1}^{n}$$

Получаем:

$$P(n) = \frac{C_{2n-1}^{n}}{2^{2n}}$$

Программная реализация.

Программа написана на языке JS с использованием технологий HTML5. Данная программа моделирует случайную величину ξ, равную числу уникальных марок в условиях задач 7 и 8.

Программа вычисляет точные значения P(ξ=n), M[ξ] и D[ξ], а также строит графики приближённого вычисления этих величин. Программа позволяет пользователю вручную задавать значения n и m и число опытов, проводимых при приближенном вычислении.

График P(ξ=n):

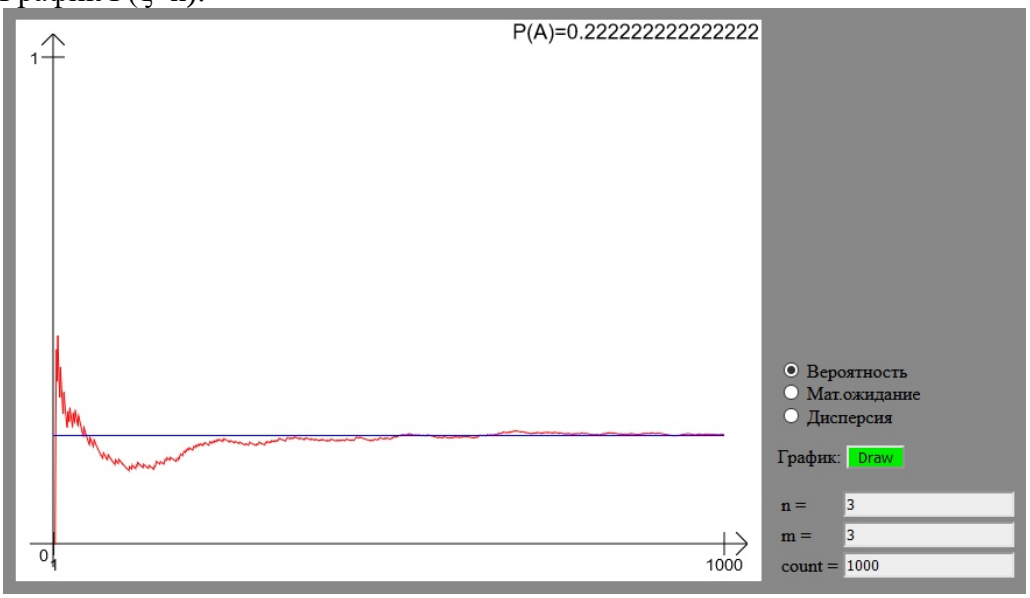


График M[ξ]:

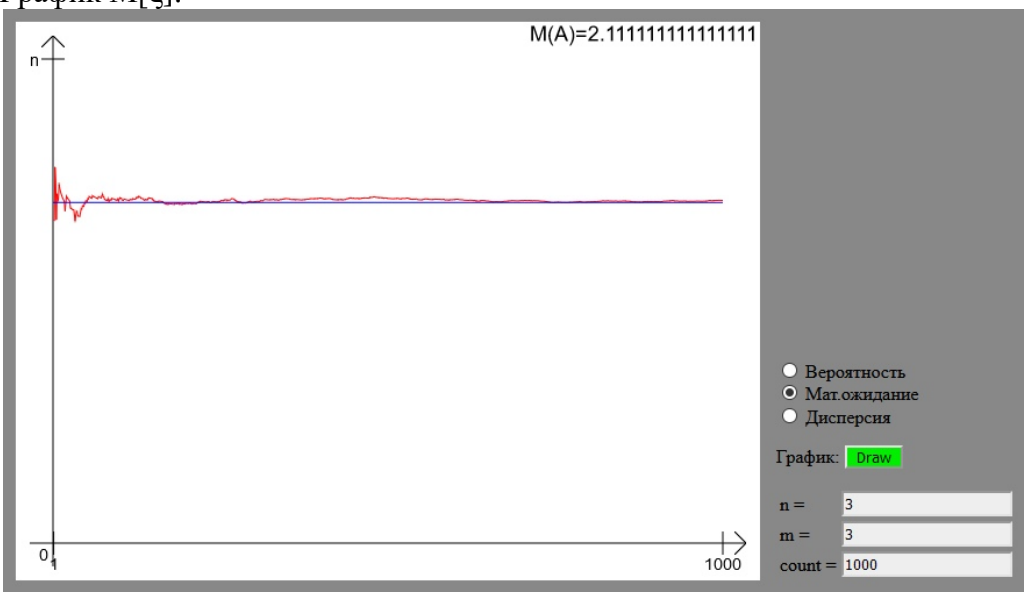
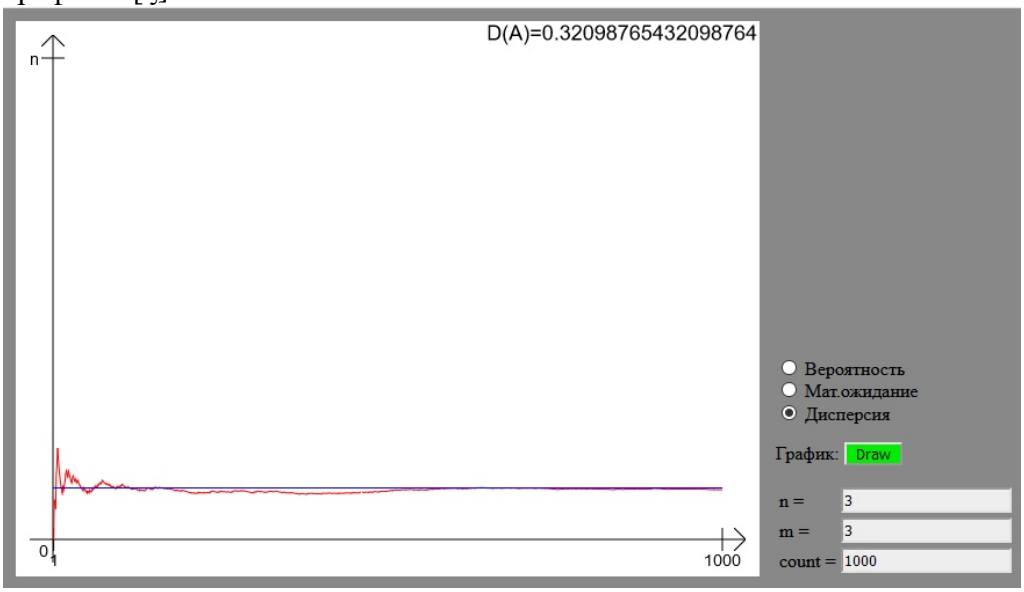


График D[ξ]:



Программная реализация (работает в браузере): https://2manovg.github.io/Marks-probability/