Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Петрозаводский государственный университет»

Институт математики и информационных технологий
Кафедра Прикладной математики и Кибернетики

Шубин Владислав Вадимович

Многочлены Бернштейна и преобразование Мак-Вильямс

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Направление подготовки бакалавра
01.03.01 Математика
Форма обучения - очная

Допущена к защите.	
Заведующий кафедрой,	Руководители:
	к. ф.–м. н.
к. ф.–м. н.	Гогин Н. Д.,
Пешкова И. В.	д. т. н.
	Кузнецов В. А.
________________	________________
«____»__________2023 г.	«____»__________2023 г.

Петрозаводск — 2023

Оглавление

Введение

Многочлены Бернштейна являются, по-видимому, исторически первым примером конструктивного доказательства теоремы Вейерштрасса о плотности подпространства многочленов в пространстве непрерывных функций $C[0,\, 1]$ [3], другими словами, справедлива следующая теорема.

Теорема 1 (Вейерштрасса) *Для любой непрерывной функции $f(x)$ на интервале $[a,\, b]$ и для $\epsilon > 0$ существует многочлен $p(x)$ такой что*

$$|f(x) - p(x)| \leq \epsilon \textit{ для всех } x \in [a,\, b].$$

Многочлены Бернштейна широко применяются для задач аппроксимации функций наравне с другими методами (например, методом наименьших квадратов) и играют важную роль в компьютерной графике, поскольку являются одной из форм аналитической записи кривых Безье [2], рассчитываемых алгоритмом де Кастельжо. В последнее время появились многочисленные приложения многочленов Бернштейна в области математической статистики, машинного обучения и нейронных сетей [18, 19, 20, 21, 22].

Количество методов вычисления многочленов Бернштейна весьма значительно, для примера укажем методы матриц Паскаля [15] и быстрого преобразования Фурье [16]. В данной работе представлен новый метод вычисления многочленов Бернштейна с помощью многочленов Кравчука и преобразования Мак-Вильямс.

Многочлены Кравчука и преобразование Мак-Вильямс известны в алгебраической теории кодирования, криптографии и теории вероятностей [4, 17].

Целью данной работы является доказательство нового метода вычисления многочленов Бернштейна и исследование возможных его приложений. Для достижения цели, поставленны следующие задачи:

1. Ввести необходимые определения.
2. Предложить новый подход к вычислению многочленов Бернштейна и дать ему обоснование.
3. Исследовать возможность его приложения к вычислению многочленов Чебышева.
4. Реализовать предложенный метод в системах компьютерной алгебры Wolfram Mathematica и SageMath.
5. Представить возможные направления применения предложенного метода в других областях.

Работа состоит из двух частей. В первой приводятся основные определения и дается обоснование нового метода. Во второй части исследуется приложение этого метода к многочленам Чебышева первого рода.

В ходе работы использовались системы компьютерной алгебры Wolfram Mathematica и SageMath. Программный код находится в Приложение. Для работы с Wolfram Mathematica и SageMath были использованны следующие учебные пособия [7, 10, 8].

Литература, использованная при подготовке работы, приведена в библиографическом списке.

— Биномиальные коэффициенты обладают удивительно большой выразительной силой.

Ю. В. Матиясевич, [13].

Многочлены Бернштейна и преобразование Мак-Вильямс

1.1 Необходимые определения

Значение функции $f(x)$ при некотором выделенном значении аргумента часто называется *отсчетом* функции $f(x)$ в этой точке. Приведем классическое определение многочлена Бернштейна, для вектора отсчетов функции $f(x)$ на промежутке $[0,\, 1]$.

Определение 1 *Пусть* $f(x) \in C[0,\, 1]$. *Многочленом Бернштейна* $B_n(f;\, x)$ *степени* n *для вектора отсчетов* $(f(0),\, f(1/n),\, \ldots,\, f(1))$ *на равномерной сетке* $x_r = r/n$, $r = 0,\, 1,\, \ldots,\, n$ *называется многочлен*

$$B_n(f;\, x) = \sum_{r=0}^{n} \binom{n}{r} f_r x^r (1-x)^{n-r}, \tag{1.1}$$

где $f_r = f(x_r)$. *Произведения* $\binom{n}{r} x^r (1-x)^{n-r}$ *называются базисными многочленами Бернштейна или многочленами Безье.*

Аппроксимационное свойство этих многочленов выражает следующая теорема [11].

Теорема 2 (Бернштейн) *Если* $f(x)$ *непрерывная функция на промежутке* $[0,\, 1]$, *то при* $n \to \infty$ *последовательность многочленов* $B_n(f;\, x)$ *сходится равномерно на промежутке* $[0,\, 1]$ *к функции* $f(x)$.

Для иных промежутков необходимо воспользоваться линейной заменой аргумента функции. Для целей настоящей работы введем новую переменную

$$t = 2x - 1 \quad \text{т. е.} \quad x = \frac{t+1}{2}. \tag{1.2}$$

Легко видеть, что при такой замене промежуток $[0,\, 1]$ переходит в промежуток $[-1,\, 1]$. Формула (1.1) в соответствии с заменой (1.2) преобразуется к виду

$$B_n(f;\, t) = \frac{1}{2^n} \sum_{r=0}^{n} \binom{n}{r} f_r (1+t)^r (1-t)^{n-r}, \tag{1.3}$$

где $(f_r) = (f(t_r))_{r=0,\,\dots,\,n}$ — $(n+1)$-вектор отсчетов функции f в точках $t_r = \frac{2r-n}{n}$, $r = 0,\, \dots,\, n$.

Определение 2 *Коэффициенты при степенях z в многочлене $(1+z)^{n-r}(1-z)^r$ являются многочленами от r; они называются многочленами Кравчука порядка n [4]:*

$$(1+z)^{n-r}(1-z)^r = \sum_{s=0}^{n} K_s^{(n)}(r)\, z^s. \tag{1.4}$$

Так как

$$(1+t)^r(1-t)^{n-r} = (1+t)^{n-(n-r)}(1-t)^{n-r},$$

то формула (1.3), учитывая (1.4), преобразуется к следующему виду

$$B_n(f;\, t) = \frac{1}{2^n} \sum_{r=0}^{n} \binom{n}{r} f_r \sum_{s=0}^{n} K_s^{(n)}(n-r) t^s, \tag{1.5}$$

так как $\binom{n}{r} = \binom{n}{n-r}$, изменив порядок суммирования, получим окончательно

$$B_n(f;\, t) = \frac{1}{2^n} \sum_{s=0}^{n} \left(\sum_{r=0}^{n} \binom{n}{r} \, f_{n-r} K_s^{(n)}(r) \right) \cdot t^s, \tag{1.6}$$

что и представляет собой разложение многочлена Бернштейна по степеням переменной t.

В следующем разделе мы укажем замкнутую форму для коэффициентов при t^s в (1.6), используя определение преобразования Мак-Вильямс, широко применяемого в теории кодирования.

1.2 Матрицы Мак-Вильямс и преобразование Мак-Вильямс

Определение 3 *Квадратная* $(n+1) \times (n+1)$*-матрица* M_n*, где*

$$(M_n)_{ij} = K_i^{(n)}(j), \quad 0 \le i,\, j < n \tag{1.7}$$

называется матрицей Мак-Вильямс [5].

Для любого вектора-столбца $u = (u_0,\, u_1,\, \ldots,\, u_n)^T$ *размерности* $(n+1)$ *его преобразованием Мак-Вильямс порядка* n *называется произведение вида*

$$\mathcal{M}_n = M_n\, u.$$

Из свойств многочленов Кравчука ([4], [5]) выводятся свойства матриц Мак-Вильямс. Приведем здесь некоторые из этих свойств.

Пусть $C = \operatorname{diag}\left(\binom{n}{0},\, \binom{n}{1},\, \ldots,\, \binom{n}{n}\right)$, $\mathbf{I}$ — единичная матрица. Тогда верны следующие соотношения:

1. Явная формула: $K_s^{(n)}(r) = \sum_{l=0}^{s} (-1)^l \binom{n-r}{s-l}\binom{r}{l}$.

2. Свободный член многочлена Кравчука: $K_r^{(n)}(0) = \binom{n}{r}$.

3. Ортогональность: $\sum_{i=0}^{n} \binom{n}{i} K_r^{(n)}(i) K_s^{(n)}(i) = 2^n \binom{n}{r} \delta_{r,\,s}$, т.е. $M_n C M_n^T = 2^n C$.

4. Инволютивность: $\sum_{i=0}^{n} K_r^{(n)}(i) K_i^{(n)}(s) = 2^n \delta_{r,\,s}$, т.е. $M_n^2 = 2^n\, \mathbf{I}$ и $M_n^{-1} = \frac{1}{2^n} M_n$.

5. Формула взаимности: $\binom{n}{r} K_s^{(n)}(r) = \binom{n}{s} K_r^{(n)}(s)$, т.е. $M_n^T = C^{-1} M_n C$.

Испльзуя явную формулу для многочленов Кравчука, построим примеры матриц Мак-Вильямс (для их построения использовалась программа из раздела приложения Python/SageMath под номером 2).

Пример 1 *Матрицы Мак-Вильямс порядков $n = 1, \ldots, 4$:*

$$M_0 = [1], \quad M_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \\ 1 & -1 & 1 \end{bmatrix},$$

$$M_3 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 3 & 1 & -1 & 3 \\ 3 & -1 & -1 & 3 \\ 1 & -1 & 1 & -1 \end{bmatrix}, \quad M_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 4 & 2 & 0 & -2 & -4 \\ 6 & 0 & -2 & 0 & 6 \\ 4 & -2 & 0 & 2 & -4 \\ 1 & -1 & 1 & -1 & 1 \end{bmatrix}.$$

Пусть ${}^{\beta}f = \left(\binom{n}{r} \cdot f_{n-r}\right)^T_{r=0,\ldots,n}$ — вектор-столбец отсчетов функции f, взятых с биномиальным весом и $\mathfrak{T}_n(f)$ — вектор коэффициентов многочлена Бернштейна $B_n(f;\, t)$ в базисе t^s.

Тогда легко видеть, что с помощью введенных обозначений, формула (1.6) представляется в виде

$$B_n(f;\, t) = \sum_{s=0}^{n} (\mathfrak{T}_n(f))_s t^s. \tag{1.8}$$

что позволяет представить вектор коэффициентов многочлена Бернштейна как

$$\mathfrak{T}_n(f) = \frac{1}{2^n} \mathrm{M}_n\, {}^{\beta}f = \mathrm{M}_n^{-1}\, {}^{\beta}f, \tag{1.9}$$

Предыдущие наши рассмотрения можно сформулировать в виде следующего предложения.

Предложение 1 $(n+1)$—*мерный вектор коэффициентов многочлена Бернштейна $\mathfrak{T}_n(f)$ является преобразованием Мак-Вильямс вектора отсчетов функции f, взятых с биномиальным весом, и деленных на два в степени n.*

1.3 Пирамида Паскаля-Мак-Вильямс

В этом разделе мы покажем, что совокупность матриц Мак-Вильямс можно наглядно представить в виде четырехгранной пирамиды, горизонтальные сечения которой суть матрицы M_n, причем каждое такое сечение представляет собой алгебраическую сумму сдвигов предыдущего сечения, подобно тому, как это имеет место для строк треугольника Паскаля, что и оправдывает название этой пирамиды [1].

Для такого представления мы используем результаты работы [5]. Введем следующие обозначения, которые используются в указанной работе.

Определение 4 *Для любой матрицы A, определим ее окаймленную нулями матрицу $\overline{|A}$, где горизонтальная (вертикальная) черта понимается как строка (столбец) из нулей:*

$$\overline{|A} = \left[\begin{array}{c|ccc} 0 & 0 & \dots & 0 \\ \hline 0 & & & \\ \vdots & & A & \\ 0 & & & \end{array}\right].$$

Обозначения $\overline{A|}$, $\underline{|A}$ и $\underline{A|}$ имеют аналогичный смысл.

В этих обозначениях конструкция пирамиды Паскаля-Мак-Вильямс представлена следующей теоремой.

Теорема 3 *Для матриц Мак-Вильямс M_n верно следующее рекуррентное соотношение (подробное доказательство содержится в [5, p. 7])*

$$M_{n+1} = \left(\underline{M_n|} + \overline{M_n|} + \underline{|M_n} - \overline{|M_n}\right)\cdot \mathrm{diag}(1,\ 1/2,\ \dots,\ 1/2,\ 1),\ n \geq 0,\ M_0 = [1]. \tag{1.10}$$

Пример 2 *Приведем матрицы Мак-Вильямс от первого до четвертого порядка из примера 1, вычисляя их по формуле* (1.10)*:*

$$M_0 = [1];$$

$$M_1 = \left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}\right) \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix};$$

$$M_2 = \left(\begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \right) \cdot$$

$$\cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \\ 1 & -1 & 1 \end{bmatrix};$$

$$M_3 = \left(\begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 0 & -2 & 0 \\ 1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 2 & 0 & -2 & 0 \\ 1 & -1 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 2 & 0 & -2 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} - \right.$$

$$\left. - \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 2 & 0 & -2 \\ 0 & 1 & -1 & 1 \end{bmatrix} \right) \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 3 & 1 & -1 & 3 \\ 3 & -1 & -1 & 3 \\ 1 & -1 & 1 & -1 \end{bmatrix}.$$

Для больших вычислений может быть использована программа из раздела приложения Wolfram Mathematica под номером 3

1.4 Заключительные замечания

1. Предложение 1 можно переформулировать следующим образом.

   Предложение 2 *Пусть $V_n = \mathbb{Z}_2 \oplus \cdots \oplus \mathbb{Z}_2$, есть диадная группа размерности n, т. е. n-мерное векторное пространство над полем $\mathbb{Z}_2$. Пусть, также, $f : V_n \to \mathbb{R}$ вещественнозначная функция на этом пространстве, определенная формулой:*

   $$\forall v \in V_n \quad f(v) = f(|v|) = f_k, \tag{1.11}$$

   где $k = |v|$ есть вес Хэмминга вектора v, $0 \le k \le n$.

   Из чего следует, что нахождение коэффициентов многочлена Бернштейна $T_n(f)$ функции отсчетов $f_r = \{f(-1), \ldots, f(1)\}$, $r = 0, \ldots, n$ на промежутке $[-1, 1]$ является задачей дискретного гармонического анализа на диадной группе V_n, поскольку многочлены Кравчука $K_r^{(n)}$ в свою очередь суть преобразования Фурье (или по-другому, преобразования Адамара) на группе V_n характеристических функций сфер Хэмминга

$$S_r = \{v \in V_n \mid |v| = r\},\ 0 \leq r \leq n.$$

Подробнее о таких методах написано в [6, 9, 26].

2. Аналогично тому, что треугольник Паскаля можно рассматривать, как результат последовательных состояний некоторого одномерного линейного клеточного автомата [1, 23], пирамиду Паскаля-Мак-Вильямс также можно интерпретировать как результат работы подобного, но уже двумерного, автомата [24], что в свое время (2004) было представлено научным руководителем автора настоящей работы в Wolfram Library Archive: `https://library.wolfram.com/infocenter/MathSource/5223/`. Современная версия программы, разработанная автором, представлена в Приложение.

Приложение метода к вычислениям многочленов Чебышева

2.1 Многочлены Чебышева

В этом разделе мы покажем пример приложения представленного подхода применительно к многочленам Чебышека первого рода $T_n(t)$.

Определение 5 *Многочленом Чебышева первого рода называется многочлен $T_n(t)$ степени n такой что*

$$T_n(\cos\alpha) = \cos n\alpha.$$

В статье [14] приводится следующая формула

$$T_n(t) = (2n-1)!! \sum_{k=0}^{n} \frac{(-1)^{n-k}}{(2k-1)!!(2n-2k-1)!!} b_{k,n}(t). \tag{2.1}$$

Данная формула позволяет сформулировать следующее предложение

Предложение 3

$$T_n(t) = \frac{1}{2^n} \sum_{s=0}^{n} \left(\sum_{k=0}^{n} (-1)^k \binom{2n}{2k} K_s^{(n)}(k) \right) t^s.$$

Доказательство.

Напомним, что произведение $\binom{n}{r} x^r (1-x)^{n-r}$ называется базисным многочленом Бернштейна. Обозначим его как $b_{n,r}$. Тогда из (1.6)

$$b_{n,r} = \frac{1}{2^n} \binom{n}{r} \sum_{s=0}^{n} K_s^{(n)}(r) t^s. \tag{2.2}$$

Подставив формулу (2.2) в (2.1) получаем

$$T_n(t) = (2n-1)!! \sum_{k=0}^{n} \frac{(-1)^{n-k}}{(2k-1)!!(2n-2k-1)!!} \left(\frac{1}{2^n} \binom{n}{k} \sum_{s=0}^{n} K_s^{(n)}(n-k) t^s \right). \tag{2.3}$$

Поскольку для любого четного $n = 2k$ имеет место $n!! = 2^k k!$ и, очевидно, $n!! = \frac{n!}{(n-1)!!} = \frac{(n+1)!}{(n+1)!!}$, то для $n = 2k-1$ будем иметь

$$n!! = \frac{(2k)!}{2^k k!}. \tag{2.4}$$

Формула (2.3) с учетом (2.4) преобразуется следующим образом

$$T_n(t) = \frac{(2n)!}{2^n n!} \sum_{k=0}^{n} \frac{(-1)^{n-k}}{\frac{(2k)!}{2^k k!} \frac{(2n-2k)!}{2^{n-k}(n-k)!}} \frac{1}{2^n} \frac{n!}{k!(n-k)!} \sum_{s=0}^{n} K_s^{(n)}(n-k) t^s =$$

$$= \frac{(2n)!}{2^n n!} \sum_{k=0}^{n} (-1)^{n-k} \frac{2^k k! 2^{n-k} (n-k)!}{(2k)!(2n-2k)!} \frac{1}{2^n} \frac{n!}{k!(n-k)!} \sum_{s=0}^{n} K_s^{(n)}(n-k) t^s =$$

$$= \frac{(2n)!}{2^n} \sum_{k=0}^{n} (-1)^{n-k} \frac{1}{(2k)!(2n-2k)!} \sum_{s=0}^{n} K_s^{(n)}(n-k) t^s =$$

$$= \frac{1}{2^n} \sum_{k=0}^{n} (-1)^{n-k} \binom{2n}{2k} \sum_{s=0}^{n} K_s^{(n)}(n-k) t^s.$$

При замене k на $n-k$ получим

$$T_n(t) = \frac{1}{2^n} \sum_{s=0}^{n} \left(\sum_{k=0}^{n} (-1)^k \binom{2n}{2k} K_s^{(n)}(k) \right) t^s. \tag{2.5}$$

$\square$

Программа под номером 5 из раздела приложения Python/SageMath может быть использована для вычисления многочленов Чебышева первого рода.

Имеет место следующее следствие предложения 3.

Следствие 1 *Для простого числа* $p > 2$ *и любого* $x \in \mathbb{Z}$

$$T_p(x) \equiv x \mod p. \tag{2.6}$$

Следствие 1 имеет важное значение в криптографии, поскольку существуют системы, использующие многочлены Чебышева. Подобный представленному обзор связей мночленов Чебышева и тестов на простоту чисел представлены в работах [25, 27]. Подобные факты и гипотезы собраны проектом Primus: `projectprimus.wordpress.com`.

Заключение

В ходе выполнения работы были достигнуты следующие результаты:

1. Разработан новый подход для вычисления коэффициентов многочленов Бернштейна.

2. Представлены приложения разработанного подхода.

3. Показаны возможные приложения представненного подхода в криптографии, теории апроксимации, разработках в области искусственного интеллекта и т.д.

4. Разработаны программы для вычисления матриц Мак-Вильямс и создания демонстрационного материала по теме исследований.

Результаты работы были представлены на конференции Polynomial Computer Algebra 2023 в Санк-Петербурге и опубликованы в тезисах конференции [12].

В дальнейшем планируется обобщить представленный подход для многомерного случая и применения в области гармонического анализа.

Приложение

3.1 Wolfram Mathematica

В приложении содержится код программ, разработанных автором в ходе исследования. К каждому листину приводится короткий пояснительный комментарий.

1. Определение базисного многочлена Бернштейна.

```
'Bernstein polynomial'
bb[n_, k_, t_] := Binomial[n, k] t^(n - k) (1 - t)^k;
```

2. Аппроксимация функции многочленом Бернштейна

```
'Approximation by Bernstein polynomials'
s = N[Expand[Sum[f[i/n]*bb[n, i, t], {i, 0, n}]]];
s = s /. {t -> (1 - t)}
```

3. Построение матриц Мак-Вильямс с помощью пирамиды Паскаля-Мак-Вильямс

```
'MW function'
MW[n_ /; n >= 1] :=
  Module[{MW, M},
   MW[1] = {{1, 1}, {1, -1}};
   M = MW[1];
   For[i = 2, i <= n, i++,
    M = ArrayPad[M, {0, 1}] + ArrayPad[M, {{1, 0}, {0, 1}}] +
      ArrayPad[M, {{0, 1}, {1, 0}}] - ArrayPad[M, {1, 0}];
    M[[All, 2 ;; -2]] = BitShiftRight[M[[All, 2 ;; -2]], 1];
    ]; M];
```

4. Аппроксимация функции матрицей Мак-Вильямс

'Approximation by MacWilliams matrices'

```
vecf[f_, n_Integer] :=
  MW[n].(1/2^n*Table[Binomial[n, r] * f[(2 r)/n - 1], {r, n, 0, -1}])
     ;
MWApprox[f_, n_Integer] := vecf[f, n].Table[x^s, {s, 0, n}];
```

5. График аппроксимации

'Plot of approximation'

```
ShowMWApprox[f_,
   n_Integer] := {Legended[
    Show[Plot[f[x], {x, -1, 1}, AxesLabel -> {"t", "f(t)"},
       PlotStyle -> Blue],
     Plot[MWApprox[f, n], {x, -1, 1}, AxesLabel -> {"t", "f(t)"},
       PlotStyle -> Red], AxesLabel -> {"t", "f(t)"}],
    SwatchLegend[{Blue, Red}, {"f", "Bernf"}]],
   Legended[Show[
     Plot[Abs[f[x] - MWApprox[f, n]], {x, -1, 1}, PlotStyle -> Purple
        ],
     AxesLabel -> {"t", "f(t)"}, GridLines -> Automatic],
    SwatchLegend[{Purple}, {"Inaccurancy"}]]}
```

6. Матрицы Мак-Вильямс по модулю некоторого целого числа

'MacWilliams modulo'

```
MWModulo[n_ /; n >= 0, m_ /; m >= 0] :=  Module[{MW, M},
    MW[0] = {1};
    MW[1] = {{1, 1}, {1, -1}};
    M = MW[1];
     s = {1, 1};
     For[i = 2, i <= n, i++,
      s = Insert[s, Round[(m + 1)/2], 2];
      M = (ArrayPad[M, {0, 1}] + ArrayPad[M, {{1, 0}, {0, 1}}] +
          ArrayPad[M, {{0, 1}, {1, 0}}] - ArrayPad[M, {1, 0}]) .
        DiagonalMatrix[s];
      M = Mod[M, m];
      ];  M];
```

7. Визуализация матриц Мак-Вильямс

'Vizualization of MacWilliams matrices modulo m'

```
hubr[x_] :=
 Hue[If[0 <= x < 0.5, 0.77, 0.414] (1 - x), 1,
```

```
If[0 <= x < 0.1, 5 x + 0.5, If[0.9 < x <= 1, -5 x + 5.5, 1]]]

VizualMWModulo[n_, m_] := Module[{P},
  P = Mod[MWModulo[n, m], m]/m;
  Image[Map[hubr, P , {2}], ImageResolution -> 1000] ]
```

3.2 Python/SageMath

1. Явная формула многочленов Кравчука

'Krawtchouk polynomials'

```
def krawtchouk(x, s, n):
    return sum([((-1) ** l) * binomial(n - x, s - l) * binomial(x, l)
        for l in range(0, s+1)])
```

2. Построение матриц Мак-Вильямс с помощью явной формулы для многочленов Кравчука

'MacWilliams mactrices'

```
def MW(n : int, p : int):
    mw = matrix(ZZ, n+1, n+1, 0)
    for i in range(0, n+1):
        for j in range(0, n+1):
            mw[i, j] = mod(krawtchouk(i, j, n), p)
    return mw
```

3. Скрипт для создания видео матриц Мак-Вильямс

'Makevideo.py'

```
#!/usr/bin/env python3

import os, glob, time
import sys

from multiprocessing import Process

def save(n, name):
    while len(glob.glob("*.png")) < n:
        print(round(100 * len(glob.glob("*.png"))/(n-1)), "%")
        time.sleep(0.2)
    #   os.system("clear")
        if len(glob.glob("*.png")) >= n-1:
```

```
            return

def makeMWs(n0, n1, p):
    if len(glob.glob("*.png")) > 0:
        os.system("rm *.png")
    os.system(f"./mws.py {n0} {n1} {p}")

def main(argv):
    p1 = Process(target = makeMWs, args=(sys.argv[1], sys.argv[2],
        sys.argv[3]))
    p2 = Process(target = save, args=(int(sys.argv[2]), sys.argv[4]))
    p2.start()
    p1.start()

    p1.join()
    p2.join()

    os.system("sips -Z 320 hue_*.png")
    os.system(f"ffmpeg -r 1 -i %01d.png -vcodec mpeg4 -y {sys.argv
        [4]}.mp4")
    os.system(f"ffmpeg -r 1 -i hue_%01d.png -vcodec mpeg4 -y hue{sys.
        argv[4]}.mp4")

    os.system("rm *.png")

    os.system(f"open {sys.argv[4]}.mp4")
if __name__ == '__main__':
    main(sys.argv)
```

4. Скрипт для создания кадров из матриц Мак-Вильямс

'mws.py'

```
#!/usr/bin/env sage -python

from sage.all import *
from sage.repl.image import Image

from music21 import note, stream

import sys

def diagonal_matrix(arr, n):
    m = matrix(ZZ, [1] + (n-1) * [0])
```

```
    for i in range(1, n):
        m = m.insert_row(i, [Integer(0)] * i + [Integer(arr[i])] + [
            Integer(0)] * (n - i - 1))
    return m

def _MW(n, p):
    mw = matrix(ZZ, [[1, 1], [1, -1]])
    for i in srange(2, n+1):
        mw = ((mw.insert_row(i, [Integer(0)]*i)).transpose().
            insert_row(i, [Integer(0)]*(i+1)).transpose() +\
        (mw.insert_row(0, [Integer(0)]*i)).transpose().insert_row(i,
            [Integer(0)]*(i+1)).transpose() +\
        (mw.insert_row(i, [Integer(0)]*i)).transpose().insert_row(0,
            [Integer(0)]*(i+1)).transpose() -\
        (mw.insert_row(0, [Integer(0)]*i)).transpose().insert_row(0,
            [Integer(0)]*(i+1)).transpose())
        mw = mw * diagonal_matrix([1] + ([round((p+1)/2)]*(i-1)) +
            [1], i+1)
        matrix_plot(matrix(IntegerModRing(p), mw), cmap='magma').save
            (f'{i}.png', dpi=250)
    return matrix(IntegerModRing(p), (mw * diagonal_matrix([1] + ([
        round((p+1)/2)]*(n-1)) + [1], n+1)))

def hu(x):
    return 0.77 if 0 <= x < 0.5 else 0.414

def br(x):
    return 5 * x + 0.5 if 0 <= x < 0.1 else -5 * x + 5.5 if 0.9 < x
        <= 1 else 1

def hrbr(n, p):

    mw = matrix(ZZ, [[1, 1], [1, -1]])
    for i in range(2, n+1):
        mw = ((mw.insert_row(i, [Integer(0)]*i)).transpose().
            insert_row(i, [Integer(0)]*(i+1)).transpose() +\
        (mw.insert_row(0, [Integer(0)]*i)).transpose().insert_row(i,
            [Integer(0)]*(i+1)).transpose() +\
        (mw.insert_row(i, [Integer(0)]*i)).transpose().insert_row(0,
            [Integer(0)]*(i+1)).transpose() -\
        (mw.insert_row(0, [Integer(0)]*i)).transpose().insert_row(0,
            [Integer(0)]*(i+1)).transpose())
        mw = mw * diagonal_matrix([1] + ([round((p+1)/2)]*(i-1)) +
```

```
            [1], i+1)

        img = Image('RGB', (i+1, i+1))
        pixels = img.pixels()

        for j in range(0, i+1):
            for k in range(0, i+1):
                H = hue(hu(int(mod(mw[j, k], p))/p) * (1 - int(mod(mw
                    [j, k], p))/p), 1, br(int(mod(mw[j, k], p))/p))
                pixels[j, k] = (int(round(255*H[0])), int(round(255*H
                    [1])), int(round(255*H[2])))
        img.save(f'hue_{i}.png')
        matrix_plot(matrix(IntegerModRing(p), mw), cmap='magma').save
            (f'{i}.png', dpi=250)
    return

def main(argv):
    hrbr(Integer(sys.argv[2]), Integer(sys.argv[3]))
if __name__ == '__main__':
    main(sys.argv)
```

5. Многочлен Чебышева первого рода через многочлены Кравчука

'Chebyshev polynomials'

```
def my_cheb(x, n):
    rslt = 0
    for s in range(0, n+1):
        for k in range(0, n+1):
            rslt += (-1)**(k) * binomial(2*n, 2*k) * krawtchouk(k, s,
                n) * x**s
    return rslt/(2**n)
```

Библиографический список использованной литературы

1. Бондаренко Б. А. *Обобщенные треугольники и пирамиды Паскаля, их фракталы, графы и приложения.* / Б. А. Бондаренко. — Ташкент : Фан, 1990. — 190 с. — Текст : непосредственный.

2. Gerald E. Farin. *Curves and Surfaces for Computer-Aided Geometric Design. A Practical Guide.* / Gerald E. Farin. — 3rd edition. — San Diego : Academic Press, 2014. — 473 с. — Текст : непосредственный.

3. Rida T Farouki. *The Bernstein polynomial basis: A centennial retrospective.* Computer Aided Geometric Design 29, 6 (2012), 379–419.

4. Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. *Теория кодов, исправляющих ошибки: Пер. с англ.* / Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. — Москва: Связь, 1979. — 746 с. — Текст : непосредственный.

5. Gogin Nikita and Hirvensalo Mika. *Recurrent construction of MacWilliams and Chebyshev matrices.* / Nikita Gogin, Mika Hirvensalo. — Текст : непосредственный // Fundam. Informaticae. — 2012. — № 16(1–4). — С. 93–110.

6. Gogin Nikita and Hirvensalo Mika. *On the Moments of Squared Binomial Coefficients.* PCA'2020 : Polynomial Computer Algebra, 2020, Международный математический институт им. Леонарда Эйлера, Санкт-Петербург, 12–17 Октября, 2020, `https://pca-pdmi.ru/2020/files/10/GoHi2020ExtAbstract.pdf` (дата обращения: 09.06.2023).

7. Н. А. Вавилов. *Mathematica для нематематика.* / Н. А. Вавилов, В. Г. Халин, А. В. Юрков. — Москва : МЦНМО, 2021. — 483 с. — Текст : электронный.

8. *Computational Mathematics with SageMath.* / P. Zimmermann, A. Casamayou, N. Cohen [и др.]. — Nancy : SIAM - Society for Industrial and Applied Mathematics, 2019. — 465 с. — Текст : непосредственный.

9. В. Н. Сачков. *Введение в комбинаторные методы дискретной математики.* / В. Н. Сачков. — 2-е изд. испр, и доп. — Москва : МЦНМО, 2004. — 424 с. — Текст : непосредственный.

10. О. А. Иванов, Г. М. Фридман. *Дискретная математика и программированние в Wolfram Mathematica.* / О. А. Иванов, Г. М. Фридман. — Санкт-Петербург : «Питер», 2020. — 352 с. — Текст : непосредственный.

11. Натансон И. П. *Конструктивная теория функций.* / И. П. Натансон. — Москва : Государственное издательство технико-теоретической литературы, 1949. — 688 с. – Текст : электронный.

12. Nikita Gogin and Vladislav Shubin. *Bernstein Polynomials and MacWilliams transform.* PCA'2023 : Polynomial Computer Algebra, 2023, Международный математический институт им. Леонарда Эйлера, Санкт-Петербург, 17–21 Апреля, 2023, `https://pca-pdmi.ru/2023/files/17/Gogin-Shubin-2023.pdf` (дата обращения: 09.06.2023).

13. Ю. В. Матиясевич. *Гипотеза Римана как чётность специальных биномиальных коэффициентов.* Чебышевский сборник, 2018, т. 19, вып. 3, с. 46–60.

14. Rababah, Abedallah. *Transformation of Chebyshev-Bernstein Polynomial Basis.* Comp. Meth. Appl. Math. 3 (4): 608–622, 2003.

15. L.H. Bezerra, L.K. Sacht. *On computing Bézier curves by Pascal matrix methods.* Appl. Math. Comput. 217 (2011) 10118–10128.

16. L.H. Bezerra. *Efficient computation of Bézier curves from their Bernstein–Fourier representation.* Appl. Math. Comput. 220 (2013) 235–238.

17. Philip Feinsilver, Jerzy Kocik. *Krawtchouk polynomials and Krawtchouk matrices.* Recent advances in applied probability, Springer-Verlag, October, 2004.

18. M. Arpogaus, M. Voss, B. Sick, M. Nigge-Uricher and O. D. *Short-Term Density Forecasting of Low-Voltage Load using Bernstein-Polynomial Normalizing Flows.* IEEE Transactions on Smart Grid, doi: 10.1109/TSG.2023.3254890.

19. Wael Fatnassi and Yasser shoukry. *PolyARBerNN: A Neural Network Guided Solver and Optimizer for Bounded Polynomial Inequalities.* arXiv:2204.05365 [cs.LO]

20. Wael Fatnassi, Haitham Khedr, Valen Yamamoto, Yasser Shoukry. *BERN-NN: Tight Bound Propagation For Neural Networks Using Bernstein Polynomial Interval Arithmetic.* arXiv:2211.14438 [cs.LG]

21. Rachid Kharoubi, Abdallah Mkhadri, Karim Oualkacha. *High-Dimensional Penalized Bernstein Support Vector Machines.* arXiv:2303.09066 [stat.ML]

22. T. Hothorn, L. Möst, and P. Bühlmann. *Most Likely Transformations.* Scandinavian Journal of Statistics, vol. 45, no. 1, pp. 110–134, 2018, ISSN: 1467-9469. DOI: 10.1111/sjos.12291.

23. J.-P Allouche; F von Haeseler; H.-O Peitgen; G Skordev *Linear cellular automata, finite automata and Pascal's triangle.* Discrete Applied Mathematics 1996-apr vol. 66 iss. 1.

24. Deepak Ranjan Nayak, Prashanta Kumar Patra, Amitav Mahapatra *A Survey on Two Dimensional Cellular Automata and Its Application in Image Processing.* arXiv:1407.7626 [cs.CV].

25. Mohamed O. Rayes, Vilmar Trevisan, Paul S. Wang. *Chebyshev Polynomials and Primality Tests.* (1999).

26. Тао Т. *Структура и случайность: Пер. с англ.* / Т. Тао. — 2-е изд. стереотип. — Москва : МЦНМО, 2017. — 360 с. — Текст : непосредственный.

27. Daisaburo Yoshioka and Kento Kawano *The period of Chebyshev polynomial sequences modulo a prime power p^k.* NOLTA'2016: International Symposium on Nonlinear Theory and Its Applications, Yugawara, Japan, November 27th-30th, 2016.