УДК 519.1 + 004.4

Алгоритм аналитического дифференцирования комбинаторных полиномов разбиений

В.А. Мельникова

Филиал Иркутского государственного университета, ул. Ленина 34, Братск, Россия
Vera_smart@rambler.ru
Статья поступила 16.05.2013, принята 26.08.2013

Комбинаторные полиномы разбиений широко используются для обращения комбинаторных сумм и моделирования дискретных распределений. К рассматриваемому классу относятся полиномы Белла и Платонова. В литературных источниках представлены различные соотношения, которые могут быть использованы для построения комбинаторных полиномов разбиений. Однако проблема разработки специального программного обеспечения, предназначенного для автоматизированного построения изучаемых полиномов, до сих пор актуальна. В статье решается задача разработки алгоритма аналитического дифференцирования комбинаторного полинома разбиений. Разработанный алгоритм предназначен для подготовки комбинаторного полинома разбиений к его использованию в рекуррентном соотношении, это необходимо для построения матрицы из полиномов. Алгоритм входит в состав программно-алгоритмического комплекса, предназначенного для автоматизированного построения изучаемых полиномов и вычисления на их основе параметров дискретных процессов восстановления. Структура и назначение основных программных методов комплекса рассмотрены в других работах автора. Представлена блок-схема алгоритма и изложены некоторые аспекты его программной реализации средствами системы Nokia Qt Creator. Также приводится описание структуры для хранения параметров полиномов в оперативной памяти, рассмотрены рекуррентные соотношения для построения полиномов. В заключительной части статьи приведены результаты оценки производительности разработанного программного комплекса.

Ключевые слова: комбинаторный полином разбиений, спецификация, алгоритм аналитического дифференцирования.

Algorithm for partition polinomials analytical derivation

V.A. Mel’nikova

Irkutsk State University, Bratsk Branch, 34 Lenin st., Bratsk, Russia
Vera_smart@rambler.ru
Received 16.05.2013, accepted 26.08.2013

Combinatorial partition polynomials are widely used for combinatorial sums conversion and discrete distribution modeling. Bell and Platonov's polynomials belong to this class. In literature, there are some expressions, which can be used for partition polynomials construction. However, the problem of developing special software designed for the computer-assisted partition polynomials construction has been of current interest so far. In this article, a special algorithm to solve the problem of developing the partition polynomials analytical derivation has been presented. The developed algorithm is designed to prepare the partition polynomials to be used in recurrence relationships in the process of polynomials matrix construction. The algorithm is included in the special program complex aimed at the computer-assisted partition polynomials construction and the discrete renewal processes parameters based on these polynomials are calculated. The structure and main program methods dispatch of the algorithmic complex have been considered in other author’s articles. Besides, the block diagram has been presented and some aspects of its software implementation in Nokia Qt Creator have been revealed. The description of the structure for storing polynomials parameters in RAM is given and the recurrence relationships for partition polynomial construction are considered. In the final part of the article, the results of the developed software performance evaluation have been demonstrated.

Keywords: combinatorial partition polynomial, specification, analytical derivation algorithm.

Введение. Однородные полиномы Белла (A-полиномы) представляют собой полиномы вида [1]:

$$A_{n,k}(g) = n!\sum_{n,k}\prod_{i=1}^{n-k+1} g_i^{r_i}\left[r_i!(i!)^{r_i}\right]^{-1}, n \geq 1, 1 \leq k \leq n, \quad (1)$$

где сумма берется по всем разбиениям натурального числа n на k целых неотрицательных слагаемых, т. е. по всем таким наборам $(r_1, r_2, ..., r_{n-k+1})$ неотрицательных чисел, что

$$\sum_{i=1}^{n-k+1} ir_i = n, \qquad \sum_{i=1}^{n-k+1} r_i = k.$$

Рассматриваемые полиномы относятся к классу комбинаторных полиномов разбиений, определение которого было введено Беллом в [2]. Они представляют собой выражение последовательных производных сложной функции через производные ее компонент и

названы полиномами Белла в [3].

С полиномами Белла сопряжены однородные полиномы Платонова (B-полиномы), которые в явном виде (при $g_1 \neq 0$) можно представить следующим образом [4]:

$$B_{n,k}(g) = (-1)^{n-k}[(k-1)!\,g_1^{2n-k}]^{-1} \sum_{2n-k,n-k} (-1)^{r_1} r_1! \times$$

$$(2n-k-r_1-1)! \prod_{i=1}^{n-k+1} g_i^{r_i} [r_i!(i!)^{r_i}]^{-1},$$

$$n \geq 2, 1 \leq k \leq n-1, \qquad (2)$$

где суммирование ведется по всем разбиениям натурального числа $(2n-2k)$ на $(n-k)$ целых неотрицательных слагаемых.

Дополнительно полагаем:

$$B_{n,n}(g) = g_1^{-n}, n \geq 1. \qquad (3)$$

В настоящее время на основе различных математических соотношений и алгоритмов разрабатываются компьютерные модели процессов и явлений, протекающих в сложных системах, использование которых позволяет проводить вычислительные эксперименты (см., например, [5]).

Комбинаторные полиномы разбиений широко применяются для моделирования дискретных распределений, в частности, известны модели дискретных процессов восстановления, составленные с использованием нормированных матриц изучаемых полиномов ([1, 6]). Следовательно, задача разработки программно-алгоритмического комплекса, предназначенного для автоматизированного построения полиномов разбиений, является актуальной. При этом важное значение имеет эффективность разрабатываемых алгоритмов с точки зрения их безопасности и компактности хранения данных.

В работе [7] описывается структура разработанного программного комплекса, а также рассмотрено назначение каждого из алгоритмов, входящего в его состав и реализованного в виде специальных программных методов.

В данной работе рассмотрен вопрос разработки одного из основных алгоритмов программного комплекса, предназначенного для аналитического дифференцирования полиномов разбиений в процессе построения матриц из рассматриваемых полиномов.

I. Рекуррентные соотношения для построения A- и B-полиномов. Для комбинаторных полиномов разбиений известны различные соотношения ([1, 6]).

Часть из них, например соотношения (1) и (2), позволяют получать полиномы в явном виде на основе найденных разбиений натуральных чисел на целые неотрицательные слагаемые. Построение матриц из полиномов разбиений указанным способом не представляется эффективным с точки зрения быстродействия алгоритма, так как различные разбиения требуются каждый раз для построения очередного полинома.

Более рациональный способ формирования матриц из рассматриваемых полиномов предполагает построение очередного полинома на основе различных рекуррентных соотношений ([1]):

$$A_{n.k}(g) = g_1 A_{n-1,k-1}(g) + DA_{n-1,k}(g),\ n,k \geq 1, k \leq n, \qquad (4)$$

$$B_{n,k}(g) = \frac{B_{n-1,k-1}(g) + DB_{n-1,k}(g)}{g_1}, \qquad (5)$$

где $D = g_2 \partial / \partial g_1 + g_3 \partial / \partial g_2 + \ldots$

$$\frac{\partial}{\partial g_i} B_{n.,k}(g) = -\binom{k+i-1}{i} B_{n+1,k+i}(g),$$

$$n,k \geq 1, k \leq n. \qquad (6)$$

В случае построения первого столбца нижней треугольной матрицы, состоящей из изучаемых полиномов, рекуррентные соотношения (4) и (5) упрощаются следующим образом:

– для матрицы из A-полиномов – $A_{n,1}(g) = g_n$; (7)

– для матрицы из B-полиномов –

$$B_{n,1}(g) = \frac{DB_{n-1,1}(g)}{g_1}. \qquad (8)$$

Для построения главной диагонали матрицы из B-полиномов применяем соотношение (3), в случае A-полиномов используем следующее соотношение:

$$A_{n,n}(g) = g_1^n\ . \qquad (9)$$

В работе [7] приведены алгоритмы, предназначенные для построения матриц из полиномов разбиений, которые составлены с использованием приведенных соотношений. Указанные алгоритмы предназначены для построения полиномов вручную, однако они были использованы как основа при разработке специальных методов программно-алгоритмического комплекса, предназначенного для автоматизированного построения рассматриваемых полиномов.

II. Спецификация полинома и структура данных для ее хранения. Каждому полиному разбиений поставим в соответствие его спецификацию.

При этом под *спецификацией* комбинаторного полинома понимается числовое множество S, образованное следующими элементами:

– коэффициенты слагаемых полинома;

– индексы множителей;

– степени множителей.

В работе [7] предложен способ расположения элементов спецификации и обоснована целесообразность его применения в рамках разрабатываемого программного комплекса.

Продемонстрируем применение спецификации на примере явного вида полиномов $A_{7,5}(g)$ и $B_{5,1}(g)$, взятых из таблиц источника [6]:

$$A_{7,5}(g) = 35 g_3 g_1^4 + 105 g_2^2 g_1^3;$$

$$B_{5,1}(g) = 105 g_1^{-9} g_2^4 - 105 g_1^{-8} g_2^2 g_3 + 10 g_1^{-7} g_3^2 +$$

$$+15g_1^{-7}g_2g_4 - g_1^{-6}g_5 .$$

Пример 1. Спецификации указанных полиномов имеют следующий вид:

$$\{(35,1,4,3,1)(105,1,3,2,2)\};$$

$$\{(105,1,-9,2,4)(-105,1,-8,2,2,3,1)(10,1,-7,3,2)$$

$$(15,1,-7,2,1,4,1)(-1,1,-6,5,1)\}.$$

Разработка методов программного комплекса осуществлялась средствами системы Nokia Qt Creator, при этом первоочередной задачей явился выбор структуры для хранения данных спецификации полинома.

В результате сравнительного анализа структурированных типов данных, поддерживаемых системой Nokia Qt Creator, был выбран контейнерный класс QList. Для представления данных спецификации полинома используется вложенная структура из целочисленных элементов QList<QList <int>>. Длина вложенного списка определяется количеством слагаемых полинома. Схемы представления спецификации полиномов $A_{7,5}(g)$ и $B_{5,1}(g)$ указанным способом приведены на рис. 1 и 2 соответственно.

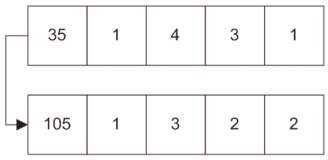

Рис. 1. Структура для хранения данных спецификации A-полинома на примере полинома $A_{7,5}(g)$

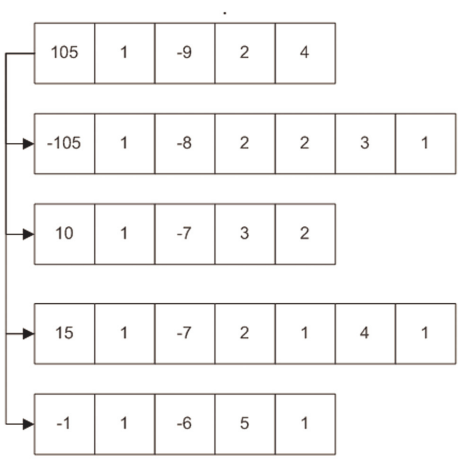

Рис. 2. Структура для хранения данных спецификации B-полинома на примере полинома $B_{5,1}(g)$

Спецификация соответствует A-полиному с приведенными слагаемыми и множителями. Параметры множителей расположены в порядке возрастания их индексов. Алгоритмы, входящие в состав комплекса, составлены с учетом сохранения компактности хранения данных.

При составлении алгоритмов комплекса необходимо было учитывать особенности и возможности выбранного типа данных, а также наличие встроенных средств манипулирования элементами контейнера, подробное описание которых можно найти в литературе по системе Nokia Qt Creator (например, в [8]).

В процессе составления алгоритмов, манипулирующих простыми и вложенными структурами QList, довольно часто возникает необходимость как в обращении к отдельному элементу списка, так и в переборе всех элементов списка. Система NokiaQTCreator поддерживает два способа такого обращения.

1) Обращение по порядковому номеру (индексу). Например, запись List1[2] позволяет обратиться ко второму элементу списка List1, при этом необходимо учитывать, что по умолчанию принята нумерация элементов с 0. Однако обращение по индексу является удобным в случаях, когда нет необходимости перебирать все элементы списка, а нужно указать на конкретный элемент. При составлении алгоритмов также следует учитывать, что индекс элемента списка может изменить свое значение в случае вставки или удаления элементов.

2) Обращение с помощью итератора QList::Iterator. Итератор является указателем на элемент контейнера. Для обхода всех элементов списка достаточно установить итератор на начальную позицию контейнера с помощью метода iterator QList::begin().

Затем необходимо организовать цикл с пред- или постусловием, выполнение тела которого прекращается при достижении итератором позиции конечного элемента, которая определяется с помощью метода iterator QList::end().

Применение нерегулярных циклов, основанных на использовании итераторов, с контролем условия выхода за пределы списка позволяет избежать сбоя в работе составленной программы, который может произойти в результате обращения к несуществующему элементу контейнера по некорректному значению его индекса.

В методах составленного алгоритмического комплекса использованы оба способа обращения к элементам контейнера.

Однотипная структура данных, используемая для хранения специфкации A- и B-полиномов, а также применение одного и того же оператора дифференцирования в рекуррентных соотношениях позволяет часть методов разрабатываемого программно-алгоритмического комплекса сделать универсальными, т. е. подходящими для использования при построении полиномов обоих рассмативаемых видов.

III. Алгоритм аналитического дифференцирования полинома с использованием оператора D. Программной реализацией алгоритма аналитического дифференцирования полинома является метод differentiationOfPolynomial, который входит в состав разрабатываемого комплекса и предназначен для подготовки спецификации полинома к ее использованию в рекуррентном соотношении (4) в случае построения A-полинома и соотношении (5) – в случае построения B-полинома.

Метод имеет следующие параметры:

– QList<QList<int>> polynomial1 – вложенный список целочисленных значений, используемый для хранения элементов спецификации первоначального полинома;

– QList<QList<int>> polynomial2 – вложенный список целочисленных значений, используемый для хранения элементов спецификации полинома, полученного в результате проведения операции дифференцирования D.

В методе использованы следующие локальные идентификаторы:

– it – итератор, предназначенный для обхода элементов списка;

– int y – целочисленная переменная, используемая для хранения значения индекса $i+1$ в случае дифференцирования полинома по переменной g_i;

– int i – целочисленная переменная, счетчик цикла, с помощью которого организован перебор всех множителей каждого слагаемого полинома;

– int stk – целочисленная переменная, предназначенная для хранения номера позиции множителя g_i;

– int coef – целочисленная переменная, предназначенная для хранения значения, на которое умножается коэффициент слагаемого полинома во время нахождения частной производной;

– QList<int> summandList – вспомогательный список, предназначенный для промежуточного хранения данных спецификации слагаемого полинома.

Блок-схема алгоритма представлена на рис. 3.

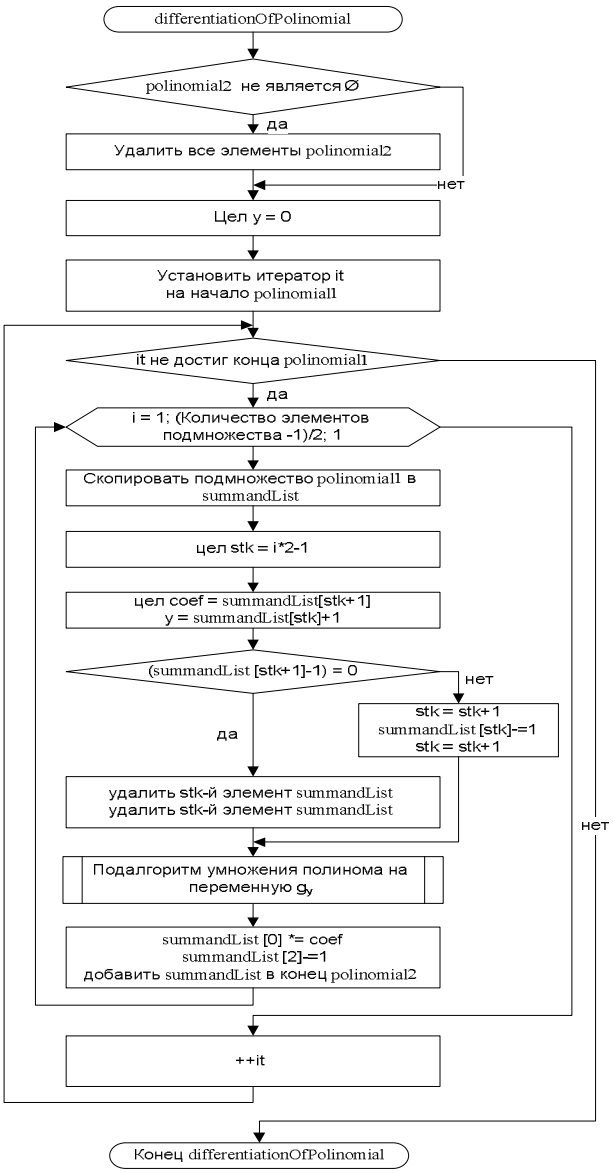

Рис. 3. Блок-схема алгоритма аналитического дифференцирования комбинаторного полинома разбиений

Блок-схема подалгоритма умножения полинома на переменную g_y представлена на рис. 4.

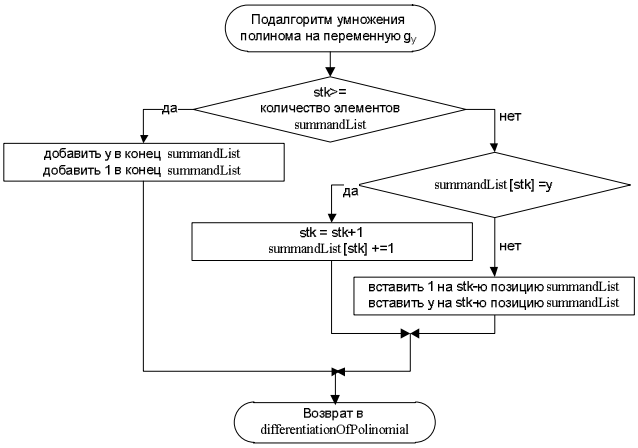

Рис. 4. Блок-схема подалгоритма умножения полинома на переменную g_y

Поскольку с алгебраической точки зрения операция нахождения частной производной подразумевает многократное умножение и деление слагаемых полинома на различные переменные, в результате ее выполнения может возникнуть необходимость приведения подобных слагаемых и множителей.

В процессе выполнения рассматриваемого алгоритма постоянно производится анализ индексов и степеней текущего множителя полинома, в зависимости от значений которых умножение на переменную g_{i+1} или деление на переменную g_i осуществляется различным образом. При умножении в случае наличия в слагаемом полинома множителя с индексом $i+1$ производится увеличение его степени на 1, в противном случае производится вставка соответствующих элементов в контейнер. При делении в случае присутствия множителя с индексом i производится уменьшение его степени на 1, в противном случае производится вставка соответствующих элементов в контейнер. Если при очередном дифференцировании получен множитель с 0-й степенью, то его данные удаляются из контейнера как неинформативные.

Также необходимо отметить, что как приведенный алгоритм, так и остальные методы программного комплекса сохраняют возрастающий порядок нечетных элементов контейнера, что соответствует упорядоченности множителей слагаемого полинома по индексам, это позволяет избежать дополнительных операций сортировки данных, загружающих алгоритмы.

IV. Оценка быстродействия разработанного алгоритма. На начальном этапе тестирования разработанного программно-алгоритмического комплекса были получены в явном виде полиномы, которые затем сравнивались с полиномами, представленными в таблицах источника [6].

В результате можно заключить, что при пробном использовании функции построения матриц из однородных А-полиномов для $n \le 10, k \le n$ полученные полиномы по своему явному виду полностью совпали с представленными в указанном источнике. Аналогичный вывод можно сделать по поводу В-полиномов,

построенных для значений $n \le 8,\ k \le n$.

Несколько B-полиномов, полученные в явном виде и не входящие в число опубликованных в таблице источника [6], приведены в [7].

Поскольку основное назначение разработанного комплекса заключается в автоматизации трудоемкой процедуры формирования комбинаторных полиномов разбиений, целесообразно оценить производительность выполнения указанной операции в автоматизированном варианте.

Полученные результаты измерения времени работы программы представлены в виде графиков на рис. 5 и 6.

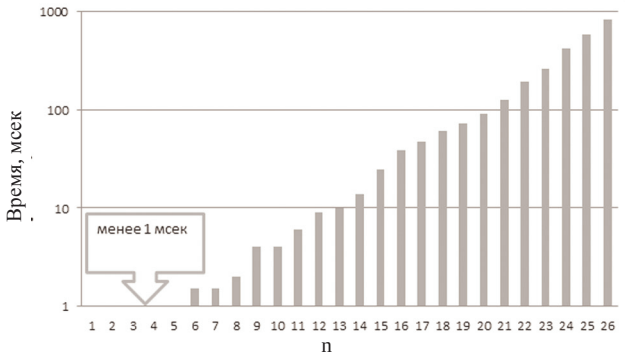

Рис. 5. Диаграмма времени построения матрицы A-полиномов автоматизированным способом при различных значениях *n*

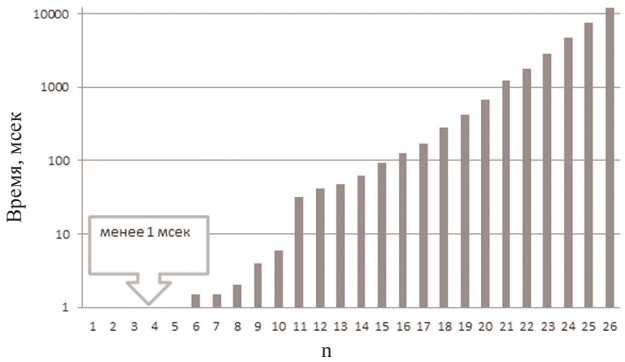

Рис. 6. Диаграмма времени построения матрицы B-полиномов автоматизированным способом при различных значениях *n*

Оценка времени и точности работы программного комплекса позволяет сделать предположение о целесообразность его применения для построения комбинаторных полиномов разбиений.

Заключение

Использование программно-алгорит-мического комплекса позволяет автоматизировать процедуру построения комбинаторных полиномов разбиений, появляется возможность проведения вычислительного эксперимента на основе моделей, составленных с применением указанных полиномов.

Алгоритмы, входящие в состав разработанного комплекса, направлены на экономичное использование вычислительных ресурсов ЭВМ. Это достигается за счет применения метода рекуррентных соотношений и динамической структуры для хранения данных.

Представление алгебраических преобразований полиномов в виде операций над их спецификациями дает возможность разработки универсальных методов программно-алгоритмического комплекса, предназначенных для построения комбинаторных полиномов разбиений различных видов.

Литература

1. Кузьмин О.В. Обобщенные пирамиды Паскаля и их приложения. Новосибирск: Наука; РАН, 2000. 294 с.

2. Bell E. F. Exponential polynomials // Ann. Math. 1934. Vol 35. P.258-277.

3. Риордан Дж. Введение в комбинаторный анализ. М.: Ин. лит., 1963. 287 с.

4.Платонов М.Л. Обращение формулы Бруно // Исследования по геомагнетизму, аэрономии и физике Солнца. М.: Наука, 1975. Вып. 35. С. 32-38.

5. Краковский Ю.М., Михайлова Е.А. Программное обеспечение многофакторного прогнозирования промышленных загрязняющих выбросов // Современные технологии. Системный анализ. Моделирование. 2011. № 3 (31). С. 92-96.

6.Кузьмин О.В. Комбинаторные методы моделирования дискретных распределений. 2-е изд., испр. и доп. Иркутск: Иркут. ун-т, 2006. 138 с.

7.Кузьмин О.В., Мельникова В.А. Алгоритмический комплекс построения однородных полиномов Платонова на основе метода рекуррентных соотношений // Современные технологии. Системный анализ. Моделирование. 2013. № 2 (38). С. 46-51.

8.Бланшет Ж., Саммерфилд М. Qt 4: Программирование GUI на C++. 2-е изд. М.: КУДИЦ-ПРЕСС, 2008. 641 с.

References

1. Kuz'min O.V. The generalized Pascal's pyramids and their applications. Novosibirsk: Nauka; RAN, 2000. 294 s.

2. Bell E. F. Exponential polynomials // Ann. Math, 1934. Vol 35, P. 258-277.

3. Riordan J. An introduction to combinatorial analysis. M.: Inostr. lit., 1963. 287 s.

4. Platonov M.L. The generalized Bruno's formulas // Issledovaniya po geomagnetismu, aeronomii and fizike solntsa. M.: Nauka, 1975. Vyp. 35. S. 32-38.

5. Krakovsky Yu.M., Mikhailova E.A. Software for multifactor forecasting of industrial emissions // Sovremennye tekhnologii. Sistemny analiz. Modelirovaniye. 2011. № 3 (31). S. 92-96.

6. Kuz'min O.V. The combinatorial methods of discreet distribution modeling. Irkutsk: Irkut. un-t, 2006. 138 s.

7. Kuz'min O.V., Mel'nikova V.A. The algorithmic complex to construct the homogeneous Platonov's polinomials based on the recurrence relations method // Sovremennye tekhnologii. Sistemny analiz. Modelirovaniye. 2013. № 2 (38). S. 46-51.

8. Blanchette J., Summerfild M. C++ GUI Programming with Qt 4. M.: "KUDITS-PRESS", 2008. 641 s.

УДК 621.314.222.6