УДК 519.6
DOI: 10.20310/1810-0198-2016-21-6-2026-2041

НОВОЕ ПОКОЛЕНИЕ СИСТЕМ СИМВОЛЬНЫХ ВЫЧИСЛЕНИЙ

© **Г. И. Малашонок**

Тамбовский государственный университет им. Г.Р. Державина
392000, Российская Федерация, г. Тамбов, ул. Интернациональная, 33
E-mail: malaschonok@ya.ru

Определяется новое поколение систем символьных вычислений – облачные математические сервисы, которые появились в последние 10 лет. Основная часть статьи посвящена описанию возможностей одной из таких систем, которая называется MathPartner. Обсуждается влияние этих систем на развитие многих современных технологий, в первую очередь, образовательных технологий. В заключительном разделе приводится обзор других известных облачных систем компьютерной алгебры и вычислительной математики.
Ключевые слова: облачная математика, компьютерная алгебра, MathPartner, символьные вычисления

1. Введение

Развитие систем символьно-численных вычислений, одновременно с развитием облачных технологий, привело к появлению нового поколения систем компьютерной алгебры – математических сервисов широкого назначения.

Одним из первых в этом классе систем является "Math Partner"[1]. Сегодня этот сервис доступен по адресу mathpar.cloud.unihub.ru.

Настоящее сообщение посвящено описанию его особенностей и тем новым возможностям, которые он предоставляет самому широкому кругу пользователей: от профессионалов математиков – до младших школьников, от физиков-теоретиков – до учителей математики и физики.

Можно ожидать, что облачные математические сервисы приведут к кардинальному изменению всей системы образования, к изменению статуса математического знания в современном обществе. Математическим аппаратом можно будет эффективно пользоваться, избегая технической рутины, во всех сферах деятельности человека.

Важным фактом, связанным с появлением облачных математических сервисов, является появление нового письменного языка математики. Этот язык максимально приближен к естественному языку, а его операторы понимает и исполняет сервер. Так как сервер свободно доступен в Интернете, то этим языком может воспользоваться и школьник, и студент, и инженер, и ученый. На таком языке можно создавать учебники и задачники, сохранять их в общедоступных базах. Такими учебниками удобно пользоваться. Можно просто копировать из них формулы и операторы, переносить их в свою рабочую тетрадь.

Для обращения к серверу не требуется приобретать какие-либо дополнительные программы. Достаточно воспользоваться современным браузером, который есть сегодня у всех. Пользователю легко освоиться на сервере, так как сервер снабжен многочисленными выпадающими меню, страницами помощи и выгружаемым руководством пользователя. Выпадающее меню содержит большой запас слов, предложений и операторов, за которыми стоят нужные пользователю математические конструкции.

Основным объектом на сервере является рабочая тетрадь пользователя. Такая тетрадь заводится для каждого пользователя и сохраняет все введенные пользователем предложения. Тексты в такую тетрадь можно загружать и выгружать как обычные текстовые файлы или способом «копировать-вставить», как в любом редакторе.

Можно ожидать, что изменится качество образования в результате индивидуализации и интенсификации. Можно будет усилить обратную связь между учащимся и образовательной системой: автоматически проверять самостоятельные результаты учащегося, при наличии ошибки автоматически рекомендовать исправить или же демонстрировать правильное решение. При этом, все действия учащегося могут сохраняться в его облачном дневнике, а ученику может присваиваться рейтинг, отражающий его знания и умения.

Основная часть этой статьи посвящена описанию возможностей облачного сервиса MathPartner. В заключительном разделе приводится обзор других известных облачных систем компьютерной алгебры и вычислительной математики.

2. О языке сервиса MathPartner

Сервис предоставляет возможность вводить и исполнять программы на языке Mathpar.

Этот язык можно рассматривать как расширение подмножества выражений языка LaTeX, основу которого много лет назад заложил Дональд Кнут.

Текст, содержащий запись на языке – это текстовый файл, который может создаваться, редактироваться и сохраняться так же, как тексты на LaTeX.

Дополнительно введены операторы присваивания, операторы вычисления, операторы управления и создания процедур. Другой смысл и форму записи имеют операторы установки окружения и операторы вывода графиков. Операторы установки окружения определяют основное числовое множество, типы операций в этом множестве, имена переменных и некоторые константы.

Имеется группа операторов вычисления, которые предназначены для использования многопроцессорного кластера при проведении вычислении.

Исходный текст и результаты вычислений могут отображаться в двух видах: в исходном виде и в виде PDF. Они появляются в одном и том же окне, а сменить вид окна можно с помощью кнопки-переключателя, которая расположена над окном.

Принципиальное отличие Mathpar от LaTeX в том, что после исполнения сервером программы результатом будет новое математическое выражение или график, которые появятся как результат вычислений.

Для более подробного знакомства с языком Mathpar можно воспользоваться литературой [2]-[6], загрузить загрузить в компьютер с сайта “Руководство по языку” или воспользоваться страницами ”Помощи” на сайте.

3. Элементы синтаксиса языка

Текст программы состоит из операторов языка, которые отделяются либо точкой с запятой, либо любыми выражениями, заключенными в двойные кавычки. Выражения, заключенные в кавычки, рассматриваются только как комментарии. А в операторах используются апострофы в тех случаях, когда нужно выделить текстовый фрагмент.

3.1. Операторы присваивания

Оператор *присваивания* в левой части содержит имя переменой, а в правой части – математическое выражение или функцию, которая определена в языка Mathpar.

3.2. Процедуры и функции

Основному тексту программы могут предшествовать *процедуры и функции*. Процедура начинается со служебного слова, имеет имя и может иметь аргументы, оператор выхода из процедуры и оператор возвращаемого значения: \ **return** objectName.

Синтаксис процедур и функций такой:
procedure $proc1(arg1, arg2, ..)\{\text{оператор}_1; \text{оператор}_2; ..\}$

3.3. Операторы управления

Кроме операторов присваивания доступны операторы *управления*:
if(){}**else**{} — оператор ветвления;
while(){} — оператор цикла с предусловием;
for(; ;){} — оператор цикла со счетчиком.

Кроме того, имеются специальные операторы: операторы *вывода значения выражений*, операторы *вывода графиков* и операторы *настройки окружения*.

3.4. Вывод результата

Используется простое правило для вывода результата вычислений. Если в исполняемой части программы не встретился оператор вывода, то выводится значение выражения в последнем операторе. Когда выводится результат, то входные выражения и результат вычислений компилируются в PDF-подобный вид и демонстрируются на экране. В частности, все имена, которые начинаются с символа “backslash”, пишутся жирным шрифтом “bold”, а сам знак (\ ) не ставится.

3.5. Некоммутативные объекты

Все переменные, которые заводит пользователь, считаются коммутативными, кроме тех, у которых имена начинаются с *символа backslash и заглавной буквы*. Вот примеры некоммутативных переменных: $\backslash A$, $\backslash Omega$, $\backslash Table$. Поэтому, например, при вводе $a*b-b*a$ результатом будет 0, но при вводе $\backslash A*\backslash B-\backslash B*\backslash A$ результатом будет $\mathbf{A}*\mathbf{B}-\mathbf{B}*\mathbf{A}$.

3.6. Допустимые отступления в записи оператора присваивания

Отметим, что синтаксис допускает следующие три отступления от правил в операторе присваивания. Слева от знака равно, в том месте, где должно быть имя переменной, разрешается размещать не только имя, но и сложное выражение. В одном операторе может присутствовать цепочка знаков "равно". Может совсем отсутствовать знак "равно"и вся левая часть.

Во всех этих некорректных случаях значение выражения, которое стоит справа, будет вычислено и доступно для вывода. Но это выражение не будет присвоено никакой переменной, если слева нет имени переменной. Все выражения, которые стоят до самого правого знака "равно", будут игнорироваться. Если оператор присваивания является последним, то вычисленное значение попадет в вывод.

Такая свобода в языке позволяет использовать сервис просто как тетрадь для записи решения задач. Например, можно ввести выражение "p=f-g=2-2" или " \ sin(a-a)= \ sin(0)". В обоих случаях будет получено значение 0. Число 0 в первом случае запишется в переменную p, а во втором случае 0 никуда не запишется.

3.7. Комментарии

Комментарии – это текст, который заключен в двойные кавычки.

Так как в комментариях кроме обычного текста могут встречаться и математические выражения, то внутри комментариев действуют стандартные правила для текста, которые приняты в LaTeX. Переключение в математическую моду происходит путем окружения знаками доллара: $ выражение $. Выделение выражения в отдельную строку с одновременным центрированием происходит путем окружения двумя знаками доллара: $$ выражение $$.

Специальная разметка текста в комментариях допускается только в пределах действия математической моды. Для красной строки можно использовать вот такое $\ \ \ $ сочетание символов. Жирный шрифт будет располагаться в центре строки, если текст выделить двумя знаками доллара: $$\bf \hbox{жирный шрифт}$$.

В комментариях можно писать шрифтом "италик" ($\it$) и использовать многие стандартные обозначения символов из LaTeXa, надстрочные и подстрочные символы. Эти символы можно найти в "подсказках" в левой панели.

4. Окружение

Окружение определяет пространство, в котором происходят вычисления. Оно позволяет установить основное числовое множество, типы операций в этом множестве, имена переменных и некоторые константы.

По умолчанию определено пространство $R64[x, y, z, t]$. Это пространство четырех переменных, самая младшая – x , самая старшая – t , над множеством приближенных действительных числе, которые хранятся в 64-битном машинном слове. Для таких чисел арифметические операции поддерживаются аппаратно.

Для смены пространства нужно выполнить команду установки нового пространства. Например, SPACE=Q[x] или SPACE=Z[p,q] и т. д.

4.1. Числовые множества

Пользователь может выбирать следующие числовые множества:
Z – множество целых чисел $\mathbb{Z}$,
Zp – конечное поле $\mathbb{Z}/\mathrm{p}\,\mathbb{Z}$, характеристика p задается пользователем (в постоянной MOD);
Zp32 – конечное поле $\mathbb{Z}/\mathrm{p}\,\mathbb{Z}$, характеристика p задается постоянной MOD32 которая должна быть меньше, чем 2^{31} ,
Z64 – подмножество целых чисел $\{z : -2^{63} \le z < 2^{63}\}$,
Q – множество рациональных чисел $\mathbb{Q}$,
R – множество приближенных действительных чисел, у которых число цифр в мантиссе задается пользователем (постоянная ACCURACY),
R64 – множество приближенных действительных чисел (со стандартной 52-разрядной мантиссой и отдельным 11-разрядным полем для хранения порядка),
R128 – множество приближенных действительных чисел, использующих 128 бит: 52-разряда в мантиссе и отдельно 64-разряда для хранения порядка.

Еще 8 числовых множеств получаются в результате комплексификации этих восьми множеств: CZ, CZp, CZp32, CZ64, CQ, C, C64, C128. Всего 16 числовых множеств.

4.2. Переменные

Можно выбирать произвольные имена для основных переменных и порядок на переменных. Например, Z[c,b,a] – это пространство полиномов от трех переменных (c,b,a) над кольцом целых чисел, с таким старшинством c<b<a.

4.3. Смена окружения

Разрешается многократно изменять окружение по своему усмотрению. Однако при этом необходимо каждый раз сменять окно вывода, которое ограничивает размер компилируемой части программы.

Все объекты, которые были созданы в других окружениях, сохраняются. Если выполняются операции над разными типами чисел, то результат автоматически приводится к тому из этих числовых типов, который может иметь погрешность больше. Имеется оператор преобразования объекта к типу чисел текущего окружения: \ toNewRing().

4.4. Постоянные окружения

Можно устанавливать или менять значения следующих постоянных.

MOD32 (268435399)– характеристика поля вычетов Zp32. Она не превосходит 2^{31}.

MOD (268435399)– характеристика поля вычетов Zp. Может быть произвольным простым числом.

RADIAN (1)– определяет меру для углов: радианы (1) или градусы (0).

TIMEOUT (15) – максимальное время (в сек.), которое отводится для вычислений.

STEPBYSTEP (0) – включает режим вывода промежуточных результатов (1) или выключает его (0).

EXPAND (1) – включает режим раскрытия всех скобок во входном выражении (1) или выключает его (0).

SUBSTITUTION (1) – включает режим замены в каждом выражении всех известных переменных их значениями (1) или выключает его (0).

FLOATPOS (2) – число десятичных знаков после запятой, которые выводятся на печать.

MachineEpsilonR64 (36) – машинное эпсилон для чисел типа R64. Число, абсолютное значение которого меньше, чем $2^{-MachineEpsilonR64}$ считается машинным нулем.

ACCURACY (25) – определяет число точных десятичных позиций после запятой для чисел типа R и C в операциях умножения и деления.

MachineEpsilonR (20) – машинное эпсилон для чисел типа R. Число, абсолютное значение которого меньше, чем $10^{-MachineEpsilonR}$ считается машинным нулем. Должно выполнялось неравенство ACCURACY > MachineEpsilonR: $10^{-ACCURACY} < 10^{-MachineEpsilonR}$.

При установке MachineEpsilonR происходит автоматическая установка нового значения ACCURACY = MachineEpsilonR + 5. Если ввести MachineEpsilonR= n/m, то будет установлено: MachineEpsilonR = n, ACCURACY = m.

В командах смены постоянных окружения используется знак равно и целые числа, например, $MOD32 = 17$.

5. Основные классы функций

5.1. Элементарные функции и математические символы

Определены следующие математические символы:
$\backslash i$ – мнимая единица i,
$\backslash e$ – число e,
$\backslash pi$ – число π,
$\backslash infty$ – знак бесконечности ∞,
$\backslash emptyset$ – пустое множество $\emptyset$

Определены следующие элементарные функции:
$\backslash exp()$ – экспоненциальная функция,
$\backslash ln()$ – натуральный логарифм,

$\backslash lg()$ – десятичный логарифм,
$\backslash log(a, b)$ или log_a(b) – логарифм числа b по основанию a,
$\backslash sgrt()$ – корень квадратный,
$\backslash cubrt()$ – корень кубический,
$\backslash rootof(a, n)$ – корень из числа a степени n,
$\backslash sin()$ – синус,
$\backslash cos()$ – косинус,
$\backslash tg()$ – тангенс,
$\backslash ctg()$ – котангенс,
$\backslash arcsin()$ – арксинус,
$\backslash arccos()$ – арккосинус,
$\backslash arctg()$ – арктангенс,
$\backslash arcctg()$ – арккотангенс,
$\backslash sh()$ – гиперболический синус,
$\backslash ch()$ – гиперболический косинус,
$\backslash th()$ – гиперболический тангенс,
$\backslash cth()$ – гиперболический котангенс,
$\backslash arcsh()$ – гиперболический арксинус,
$\backslash arcch()$ – гиперболический арккосинус,
$\backslash arctgh()$ – гиперболический арктангенс,
$\backslash arcctgh()$ – гиперболический арккотангенс,
$\backslash sc()$ – секанс,
$\backslash csc()$ – косеканс,
$\backslash arcsec()$ – арксеканс,
$\backslash arccsc()$ – арккосеканс.

5.2. Операции над числами

$\backslash max(a, b)$ – максимальное из двух чисел,
$\backslash min(a, b)$ – минимальное из двух чисел,
$\backslash sign(a)$ – знак числа или знак старшего коэффициента у полинома,
$\backslash abs(a)$ – абсолютное значение числа или модуль для комплексного числа:
$\backslash floor(a)$ – наибольшее целое число, которое не превосходит данное,
$\backslash ceil(a)$ – наименьшее целое число, которое не меньше данного числа,
$\backslash round(a)$ – округление к ближайшему целому числу,

5.3. Операции над целыми числами

$\backslash divRem(a, b)$ – целая часть частного и остаток при делении a на целое число b,
$\backslash div(a, b)$ – целая часть частного при делении целого числа a на число b,
$\backslash rem(a, m)$ – остаток при делении a на m,
$\backslash mod(a, m)$ – остаток от деления a на нечетное число m, который заключен в интервале от -(m-1)/2 до (m-1)/2,
$\backslash fact(a) = a!$ – факториал целого числа a.

5.4. Операции с дробями и рациональными функциями

$\backslash num(fr)$ – числитель дроби,
$\backslash denom(fr)$ – знаменатель дроби,
$\backslash cancel(fr)$ – сократить дробь,
$\backslash properForm(fr)$ – выделить целую часть и правильную дробь,

$\backslash quotientAndRemainder(fr)$ – частное и остаток при делении числителя дроби на знаменатель,
$\backslash quotient(fr)$ – частное при делении числителя дроби на знаменатель,
$\backslash remainder(fr)$ – остаток при делении числителя дроби на знаменатель.

5.5. Операции над многочленами многих переменных

$\backslash value(f, [x_0, y_0])$ – подставить в многочлен f значения переменных $x = x_0$, $y = y_0$ и вычислить его,
$\backslash expand()$ – раскрытие всех скобок в алгебраическом выражении,
$\backslash factor()$ – разложение многочлена на множители,
$\backslash GCD()$ – вычисление НОД полиномов,
$\backslash LCM()$ – вычисление НОК полиномов,
$\backslash groebner(f_1, f_2, .., f_c)$ – вычисление базиса Гребнера идеала, порожденного многочленами f_1, $f_2, ..,$ f_c.
$\backslash reduceByGB(g, [f_1, f_2, .., f_c])$ – редукция полинома g с помощью полиномов f_1, $f_2, ..,$ f_c. Если эти полиномы составляют базис Гребнера, то результатом будет редукция по идеалу $(f_1, f_2, .., f_c)$, который они порождают.
$\backslash solveNAE(f_1, f_2, .., f_c)$ – решение системы нелинейных алгебраических уравнений ($f_1 = 0$, $f_2 = 0, ..,$ $f_c = 0$),
$\backslash solve(p(x))$ or $\backslash solve(p(x) = 0)$ – найти корни полинома в области, которая определена окружением SPACE: Если SPACE определено над действительными числами, то ищутся все действительные корни, а если над комплексными числами, то ищутся все комплексные корни, в остальных случаях ищутся (при возможности) корни, содержащие знаки радикала и/или буквенные параметры, если они присутствуют в коэффициентах.
$\backslash quotientAndRemainder(a, b)$ – частное и остаток при делении a на b,
$\backslash quotient(a, b)$ – частное при делении a на b,
$\backslash remainder(a, b)$ – остаток при делении a на b.

В трех последних функциях можно добавлять третий параметр – это имя переменной, которая является главной в этой операции.

Оценки сложности полиномиальных алгоритмов можно найти в работе [7].

5.6. Функции нескольких аргументов

Некоторые функции от двух аргументов:
$\backslash log(a, b)$ – логарифм $\log_a(b)$,
$\backslash rootOf(b, n)$ – корень из b степени n,
$\backslash value(f, [x_0, y_0])$ – подставить в функцию f значения переменных $x = x_0$, $y = y_0$ и вычислить ее,
$\backslash value(f, [g_0, g_1])$ – произвести замену переменных в функции f: $x \to g_0$, $y \to g_1$.
$\backslash int(f)\ d\ x$ – интеграл от f по x (первообразная без произвольной постоянной),
$\backslash D(f, x)$ – производная функции f по переменной x,
$\backslash D(f, x\hat{}n)$ – производная порядка n функции f по переменной x,
$\backslash D(f, x\hat{}n\ y\hat{}m)$ – смешанная производная функции f по переменной x порядка n, по переменной y порядка m,
$\backslash lim\_\{x \backslash to\ a\}(f)$ предел функции f при x стремящемся к a,
$\backslash binom(n, k)$ – число сочетаний из n по k.

Матрицы и векторы \ O_{n,m} — нулевая матрица размера $n \times m$; \ I_{n,m} — $n \times m$ матрица с единицами на главной диагонали. \ O_{n}. \ charPolynom(), \ kernel(), \ transpose() или **A**^{**T**} , — транспонированная матрица;

\ conjugate() или **A^{\ast}** — сопряженная матрица;
\ toEchelonForm() — эшелонная (ступенчатая) форма;
\ det() — определитель;
\ inverse() или **A^{−1}** — обратная матрица;
\ adjoint() или **A^{\star}** — присоединенная матрица;
\ genInverse() или **A^{+}** — обобщенная обратная матрица Мурра-Пенроуза;
\ closure() или **A^{\times}** — замыкание, т.е. сумма $I + A + A^2 + A^3 + \ldots$. Для классических алгебр это эквивалентно $(I - A)^{-1}$;
\ LDU() — треугольная LDU-факторизация матрицы. Результатом являются пять матриц $[L, D, U, P, Q]$. Произведение LDU равно исходной матрице. Здесь L — нижняя треугольная матрица, U — верхняя треугольная матрица, D — матрица перестановок, умноженная на диагональную матрицу, P и Q – матрицы перестановок. При этом $P^T LP$ и $Q^T UQ$ – это нижняя и верхняя треугольные матрицы, $P^T DQ$ – это диагональная матрица, у которой все нулевые диагональные элементы собраны в нижних строках.
\ BruhatDecomposition() — разложение Брюа матрицы. Результатом являются три матрицы $[V, w, U]$, V и U — верхние треугольные матрицы, w — матрица перестановок, умноженная на диагональную матрицу, при этом произведение VwU равно исходной матрице.

Здесь использовались матричные алгоритмы из работ [7-11].

5.7. Преобразование Лапласа и решение систем линейных дифференциальных уравнений с постоянными коэффициентами

$\backslash laplaceTransform()$ – прямое преобразование Лапласа,
$\backslash inverseLaplaceTransform()$ – обратное преобразование Лапласа,
$\backslash solveLDE(S, J)$ — решение системы линейных дифференциальных уравнений с постоянными коэффициентами S при начальных условиях J. Вот примеры задания S и J.
$\backslash systLDE(\backslash d(x,t) + x - 4y = t\hat{}2\backslash exp(2t)),\ 7\backslash d(y,t) - 2x - 2y = 3\backslash exp(3t) + t);$ – пример задания системы дифференциальных уравнений S: $\begin{cases} x'_t + x - 4y = t^2 e^{2t}, \\ 7y'_t - 2x - 2y = 3e^{3t} + t. \end{cases}$
$\backslash initCond(\backslash d(x,t,0,0) = 2, \backslash d(y,t,0,0) = 0);$ – пример задания начальных условий J для решения системы дифференциальных уравнений.
Здесь используются обозначения:
$\backslash d(f,t)$ – обозначение для производной функции f по переменной t.
$\backslash d(f,t,k)$ – обозначение для k-той производной функции f по переменной t.
$\backslash d(f,t,k,t_0)$ – обозначение для k-той производной функции f по переменной t в точке t_0. Если $k = 0$, то это обозначение для функции f в точке t_0.

Если система дифференциальных уравнений дана в матричном виде, то можно использовать оператор решения с тремя аргументами:
$\backslash solveLDE(A, B, C)$. Здесь матрица A – это левая часть системы дифференциальных уравнений после преобразования Лапласа, вектор B – правая часть системы после преобразования Лапласа и C – это матрица начальных условий.

С алгоритмами можно ознакомится по работам [12-14].

5.8. Функции теории вероятностей и математической статистики

5.8.1. Функции непрерывных случайных величин

Для непрерывной случайной величины с плотностью распределения $f(x)$, заданной на интервале (a,b), определены следующие функции:
$\backslash mathExpectation(a, b, f(x))$ – математическое ожидание,

$\backslash dispersion(a, b, f(x))$ – дисперсия,
$\backslash meanSquareDeviation(a, b, f(x))$ – среднее квадратичное отклонение.

5.8.2. Функции дискретных случайных величин

Дискретная случайная величина задается двустрочной матрицей: в первой строке записаны значения случайной величины, во второй — соответствующие им вероятности. Сумма всех элементов второй строки равна единице. Например, a=[[1,2,3,4,5],[0.4,0.1,0.1,0.2,0.2]].

Для дискретных случайных величин определены следующие функции:
$\backslash mathExpectation()$ – математическое ожидание,
$\backslash dispersion()$ – дисперсия,
$\backslash meanSquareDeviation()$ – среднее квадратичное отклонение,
$\backslash addQU(a, b)$ – сумма двух дискретных случайных величин,
$\backslash multiplyQU(a, b)$ – произведение двух дискретных случайных величин,
$\backslash covariance(a, b)$ – коэффициент ковариации двух дискретных случайных величин,
$\backslash correlation(a, b)$ – коэффициент корреляции двух дискретных случайных величин,
$\backslash simplifyQU()$ упрощение записи дискретной случайной величины, если в ней присутствуют повторяющиеся значения.

5.8.3. Функции для выборок

Выборка задается как вектор, например, S=[1,7,10,15]. Для выборок определены следующие функции:
$\backslash sampleMean(S)$ – выборочное среднее,
$\backslash sampleDispersion(S)$ – выборочная дисперсия,
$\backslash covarianceCoefficient(S_1, S_2)$ – коэффициент ковариации для двух выборок,
$\backslash correlationCoefficient(S_1, S_2)$ – коэффициент корреляции для двух выборок.

5.9. Создание случайных объектов: чисел, полиномов, матриц. Функция времени.

$\backslash randomNumber(k)$ – случайное число, содержащее k бит.
$\backslash randomPolynom(n_x, n_y, n_z, denP, k)$ – случайный полином, у которого коэффициенты – это числа, содержащее k бит, степени по переменным x, y, z не превосходят n_x, n_y, n_z, соответственно, и плотность полинома равна $denP$ процентов. Плотность полинома – это отношение числа ненулевых коэффициентов к максимально возможному числу коэффициентов $(n_x + + 1)(n_y + 1)(n_z + 1)$.
$\backslash randomMatrix(m, n, denM, k)$ случайная числовая матрица $m \times n$ с плотностью $denM$ (процентов), элементы которой – это k битные числа. Плотность матрицы – это отношение числа ненулевых элементов к общему числу элементов mn.
$\backslash randomMatrix(m, n, denM, n_x, n_y, n_z, denP, k)$ случайная полиномиальная матрица с плотностью $denM$, элементы которой – это случайные полиномы, которые создаются оператором $\backslash randomPolynom(n_x, n_y, n_z, denP, k)$.
$\backslash time()$ – время в миллисекундах. Чтобы измерить время некоторого процесса, нужно вычесть время возвращаемое этой функцией в конце и в начале этого процесса.

6. Вычисления в тропической математике

6.1. Идемпотентные алгебры

Кроме классических числовых алгебр с операциями + , -, * и операцией “делить” для полей, можно производить вычисления в идемпотентных алгебрах (в тропической матема-

тике). В таких алгебрах первая операция является идемпотентной. В названии тропической алгебры сразу указываются алгебраические операции. При выборе такой алгебры происходит переназначение символов + и *, и порядка на числовом множестве. В них вводится порядок, согласованный со сложением:

$$x \le y \Leftrightarrow x \oplus y = y\,.$$

Поэтому нейтральный элемент по сложению $\mathbb{0}$ является наименьшим элементом по порядку, но при этом он может отличаться от числового нуля.

Можно задавать следующие идемпотентные полуполя:
1) ZMaxPlus, ZMinPlus – на множестве целых чисел Z,
2) RMaxPlus, RMinPlus, RMaxMult, RMinMult – на множестве действительных чисел R,
3) R64MaxPlus, R64MinPlus, R64MaxMult, R64MinMult – на множестве действительных чисел R64.

Можно задавать следующие идемпотентные полукольца:
1) ZMaxMin, ZMinMax, ZMaxMult, ZMinMult – на множестве целых чисел Z,
2) RMaxMin, RMinMax – на множестве действительных чисел R,
3) R64MaxMin, R64MinMax – на множестве действительных чисел R64.

Всего 18 разных типов алгебр. Вот примеры операторов устанавливающих тропическое окружение: SPACE = ZMaxPlus [x, y, z]; SPACE = RMaxMin [u, v].

Пример простой задачи в полукольце ZMaxPlus:

$$a = 2;\ b = 9;\ c = a + b;\ d = a * b;\ \backslash print(c, d)$$

В результате будет $c = 9,\ d = 11$, т. к. произошло назначение операций: знаком плюс обозначается бинарная функция максимум, а знаком умножить обозначается операция сложения.

Помимо бинарных операций сложения и умножения доступна унарная операция *замыкания*. Обозначается она так: $\backslash closure(a)$, где a — это число или матрица. В результате вычисляется выражение $1 + a + a^2 + a^3 + \ldots$

6.2. Операторы в тропических алгебрах

В тропических алгебрах имеются следующие операторы:
$\backslash solveLAETropic(A, b)$ — частное решение системы уравнений $Ax = b$.
$\backslash BellmanEquation(A)$ — решение однородного уравнения Беллмана $Ax = x$;
$\backslash BellmanEquation(A, b)$ — решение неоднородного уравнения Беллмана $Ax + b = x$;
$\backslash BellmanInequality(A)$ — решение неравенства $Ax \le x$;
$\backslash BellmanInequality(A, b)$ — решение неравенства $Ax \oplus b \le x$.

Если для графа задана матрица A расстояний между смежными вершинами, то для нее в алгебрах Min-Plus определены еще два оператора:
$\backslash searchLeastDistances(A)$, — находит кратчайшие расстояния между всеми вершинами,
$\backslash findTheShortestPath(A, i, j))$ — находит кратчайший путь между вершинами i и j.

7. Построение 2D и 3D графиков

7.1. Построение графиков функций на плоскости

Графическое окружение задается командой set2D(). У нее могут быть следующие параметры:
$\backslash set2D(x0, x1)$, $\backslash set2D(x0, x1, 'title')$, $\backslash set2D(x0, x1, y0, y1)$, $\backslash set2D(x0, x1, y0, y1, 'title')$,
$\backslash set2D(x0, x1, 'title', 'nameOX', 'nameOY')$, $\backslash set2D(x0, x1, y0, y1, 'title', 'nameOX', 'nameOY')$.

Здесь x0, x1, y0, y1 – это границы графика, title, nameOX, nameOY – заголовок графика и наименование его осей.

Последними в списке параметров могут стоять BW и ES: BW устанавливает черно-белый цвет графика, ES устанавливает равенство масштаба по шкале x и шкале y.

Полный формат для команды set2D() предполагает 3 группы параметров, каждая из которых пишется в квадратных скобках: set2D([x0,x1, y0, y1],['xTitle','yTitle','title'] ,[0,1,12,3,5]).

Первая квадратная скобка определяет границы графика. Эта скобка должна обязательно присутствовать. Если в первой скобке указаны только границы по оси абсцисс, то границы по оси ординат рассчитываются автоматически.

Вторая квадратная скобка – это надписи к осям координат и подпись ко всему рисунку.

Третья квадратная скобка содержит 5 чисел: 1) 1 — означает: установить режим черно-белый (0 - цветной), 2) 1 — означает: установить равный масштаб по обеим осям (0- золотое сечение), 3) это размер шрифта для подписей, 4) это толщина линий графиков, 5) это толщина координатных осей. Есть для этой скобки и 3 сокращенных варианта:['ES'],['BW'],['ESBW']. Они, соответственно, устанавливают в значение 1 или первый параметр, или второй параметр, или оба. Любая из квадратных скобок, кроме первой, может не присутствовать.

Параметрами изображаемой линии могут быть: dash (пунктир), arrow (стрелки) и dashAndArrow (пунктир и стрелки). Эти параметры могут стоять в конце списка параметров функций plot, tablePlot, и paramPlot. Например, $\backslash plot(sin(x), dash)$.

Три способа задания графиков:
$\backslash plot(f)$,где f=f(x) – функция, заданная явно,
$\backslash paramPlot([f, g], [x_0, x_1])$, где f=X(x), g=Y(x) — функции, заданная параметрически, а $[x_0, x_1]$ — интервал изменения параметра.
$\backslash tablePlot(T)$, где $T = [[x_1, .., x_n], [f_1, .., f_n], .., [g_1, .., g_n]]$ – это одна или несколько таблично заданных функций, записанных в виде матрицы: первая строка – значение абсцисс, а вторая и следующие – значения функций в этих точках. Каждая функция записывается в одну строку.

Можно изобразить несколько графиков на одном рисунке. Для этого нужно заранее определить эти графики, например, $P1 = \backslash plot(x^2)$; $P2 = \backslash plot(x^3 + 1)$. Тогда
$\backslash showPlots([P1, P2])$ – дает изображение нескольких графиков на одном рисунке. Параметры, которые могут стоять в конце всех параметров showPlots: 'noAxes' — вывод графика без осей и 'lattice' — вывод графика с сеткой.

7.2. Построение 3D графиков функций

7.3. 3D графики функций, которые заданы явно

Для построения 3D графика функции f=f(x,y) используется команда
$\backslash plot3d(f, [x0, x1, y0, y1])$, где [x0,x1] — интервал по оси OX, [y0,y1] — интервал по оси OY.

Перемещение мыши с нажатой левой кнопкой приводит к вращению системы координат графика. Перемещение мыши с нажатой левой кнопкой и нажатой клавишей Shift приводит к изменению масштаба изображения.

7.4. 3D графики функций, которые заданы неявно

Для построения графика функции f(x,y,z)=0 используется команда
$\backslash implicitPlot3d(f, xMin, xMax, yMin, yMax, zMin, zMax)$, где числа xMin, xMax, yMin, yMax, zMin, zMax задают область в пространстве, имеющую форму параллелепипеда, в которой изображается неявная функция.

Можно дополнительно указывать координаты источника света (lightX, lightY, lightZ), цвет (color) и число точек на сетке (gridSize). По умолчанию принимается на сетке 50 точек на

каждом ребре параллелепипеда. Цвет в формате RGB (красный, зеленый, голубой) задается числом color=R*256*256+G*256+B, где каждая буква обозначает неотрицательное целое число не превосходящее 255. Например, 255*256*256 — красный цвет, а 255*256*256+255*256 — желтый (красный+зеленый). Допустимые наборы аргументов команды implicitPlot3d:

(f,xMin,xMax,yMin,yMax,zMin,zMax,gridSize),

(f,xMin,xMax,yMin,yMax,zMin,zMax,lightX,lightY,lightZ,gridSize),

(f,xMin,xMax,yMin,yMax,zMin,zMax,lightX,lightY,lightZ,color,gridSize).

8. Сервис пользователя

8.1. Загрузка и исполнение операторов

Основной способ ввода текста – это ввод текста пользователем в поле рабочей тетради в окне браузера. Можно вводить и сохраненный раннее текст, если имеется соответствующий текстовый файл. Такой текст сразу появится на экране.

Имеется возможность ввода больших математических объектов, которые не надо выводить на экран. Например, матриц или векторов большого размера. Для этого предусмотрен ввод математического объекта из файла пользователя. Предварительно этот математический объект должен быть записан в языке Mathpar и сохранен в виде текстового файла.

Оператор **a=\ fromFile(fileName)** присвоит переменной **a** объект из файла пользователя с именем fileName. При этом вывод объекта на экран не происходит.

8.2. Вывод и сохранение результатов

Поле, в котором появляются результаты вычислений, находится в рабочей тетради в окне браузера сразу под окном ввода. Сервис рабочей тетради обеспечивает добавление или удаление дополнительных окон для ввода. В каждом таком отдельном окне можно исполнять команды и получать результаты вычислений независимо от остальных. При этом все ранее введенные объекты сохраняются в памяти и могут быть использованы в любом окне.

Имеется возможность сохранить в одном текстовом файле все поля ввода и вывода, которые имеются в рабочей тетради. При загрузке такого файла обратно на сервер произойдет восстановление всех окон сразу. Пользователь может сохранить окна в языке Mathpar, в языке LaTeX или в pdf-файле.

Кроме этого, есть возможность сохранить математический объект в текстовом файле пользователя. При выполнении оператора **\ toFile(** f **, fileName)** происходит запись объекта f в языке Mathpar в текстовый файла и сохранение этого файла на компьютере пользователя. При этом сам объект на экран не выводится. Тем самым достигается сохранение больших объектов, запись которых в языке Mathpar превышает размеры экрана. Такие объекты могут быть затем введены прямо из полученного файла, как было сказано выше.

8.3. Очистка памяти

В памяти сохраняются все объекты, которые ввел пользователь. Для освобождения всех переменных в сервисе предусмотрена кнопка очистки памяти. Она расположена в правом верхнем углу и отмечена символом "c". Кроме того, очистку памяти от всех переменных можно выполнить и при помощи оператора $\backslash clean()$ языка Mathpar. В таком операторе можно указать имена переменных, которые нужно очистить в качестве аргументов. В таком случае сохранятся все те объекты, у которых не отмечены имена.

9. Сервис для вычислений на кластере

10. Настройка параметров задачи для вычислений на кластере

MathPartner обеспечивает связь с вычислительным кластером и предоставляет возможность для выполнения параллельных вычислений на кластере для отдельных операторов. Работы в области параллельной компьютерной алгебры ведутся уже более десяти лет см. [15, 16].

Для вычислений на кластере нужно задать постоянные настройки задачи на кластере:
TOTALNODES — общее количество узлов кластера, которые выделяются для вычислений,
PROCPERNODE — количество MPI-процессов, запускаемых на одном узле,
CLUSTERTIME — максимальное время (в минутах) выполнения программы, после истечения которого программа принудительно завершится.
MAXCLUSTERMEMORY — объем памяти, выделяемый для JVM для одного MPI-процесса (опция -Xmx). Так как общая оперативная память на узле делится поровну между всеми его ядрами, то за счет изменения количество ядер, которые выделяются на одном узле, можно менять объем памяти, который приходится на одно ядро.

11. Операторы для параллельных вычислений

В настоящий момент подключены следующие параллельные операторы: Для классических пространств (Z[x]) – это умножение матриц и вычисление обратной матрицы (присоединенной матрицы и определителя) – $\backslash adjointDetPar(A)$ Для тропических пространств (R64MaxPlus[x]): $\backslash BellmanEquationPar(A)$, $\backslash BellmanEquationPar(A, b)$, $\backslash BellmanInequalityPar(A)$,
$\backslash BellmanInequalityPar(A, b)$.

12. Обзор известных систем "Облачной Математики"

12.1. Пакеты компьютерной алгебры

Уже много лет известен математический сервис Вольфрам-Альфа (wolframalpha.com). Он предоставляет возможность выполнить один оператор языка Mathematica.

Осенью 2015 г. появились сообщения, что можно покупать доступ на полнофункциональную β-версию Mathematica Online – облачный вариант системы Mathematica.

Лицензии на пакет Mathematica есть у таких вузов, как Оксфордский университет, Кэмбриджский университет, Гарвардский университет. Среди российских университетов пакеты Mathematica приобрели Санкт-Петербургский государственный университет экономики, Томский политехнический университет, Дальневосточный федеральный университет, Уральский федеральный университет, Российская академия народного хозяйства и государственной службы, физико-математическая школа 239 в Санкт-Петербурге.

Компания Maplesoft не объявляла о планах создания облачного сервиса. Под названием Maple Cloud сегодня понимается сервис, который дает возможность пользователям обмениваться файлами с программами на языке Maple. Компания Maplesoft в 2009 г. была приобретена японской компанией Cybernet Systems Co. Ltd. для разработки программного обеспечения для автомобилей Тойота и с этого времени является ее филиалом.

Из свободно распространяемых систем компьютерной алгебры можно выделить систему SAGE. Она предоставляет графический интерфейс ко многим открытым пакетам компьютерной алгебры и существует как в десктопном варианте, так и в облачном варианте (cloud.sagemath.com).

На прошедшей в сентябре 2015 г. в Аахене (Германия) конференции "Computer Algebra in Scientific Computing"в двух докладах были сообщения о новых разработках в области Cloud Mathematics (см. сборник трудов конференции в [11]).

Один из них был посвящен облачному сервису *Solve Polynomial Systems by PHCpack(beta)*, который размещен на сайте kepler.math.uic.edu. Он предназначен для решения полиномиальных систем.

Другой доклад был посвящен облачному сервису *CloudMath*. Этот сервис разработан компанией Dr. Hornecker Softwareentwicklung (Freiburg/Germany) в сотрудничестве с профессором Wolfram Koepf из Института Математики университета Касселя (Германия). В настоящее время этот сервис проходит β-тестирование. Сервис CloudMath дает on-line доступ к SAGE и к другим открытым системам компьютерной алгебры. Информацию о нем можно найти на сайте http://www.hornesker.eu/cloudmatheng.aspx

12.2. Пакеты вычислительной математики

Среди пакетов вычислительной математики лидирует MATLAB. Он предоставляет пользователю интерактивную среду для разработки алгоритмов и анализа данных. Здесь можно строить 2D и 3D графики для визуализации данных, решать вычислительные задачи в таких областях, как обработка изображений, обработка сигналов, анализ систем управления, статистика, оптимизация и другие.

С 2012 г. он обеспечивает Интернет-доступ к рабочему столу MATLAB с полной поддержкой языка MATLAB из любого стандартного веб-браузера.

Он поддерживает кластерные вычисления на своем облаке (до 90 сек.) и на Amazon EC2 (до 15 мин.) (www.mathworks.com/products/parallel-computing/parallel-computing-on-the-cloud/)

СПИСОК ЛИТЕРАТУРЫ

1. *Malaschonok G.I.* Way to Parallel Symbolic Computations.«Облачные вычисления. Образование. Исследования. Разработка». М., 2011. URL://www.unicluster.ru/conf/2011/docs/TSU.Malaschonok G.I.pdf

2. *Малашонок Г.И.* Руководство по языку «MATHPAR»: учебное пособие. Тамбов: Издат. дом ТГУ им. Г.Р. Державина, 2013. 133 с.

3. *Малашонок Г.И.* О проекте параллельной компьютерной алгебры // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2009. Т. 14. Вып. 4. С. 744–748.

4. *Malaschonok G.I.* Project of Parallel Computer Algebra // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2010. Т. 15. Вып. 6. С. 1724–1729.

5. *Малашонок Г.И., Переславцева О.Н., Ивашов Д.С.* Параллельные символьные вычисления // Суперкомпьютерные технологии в науке, образовании и промышленности / под ред. В.А. Садовничего, Г.И. Савина, Вл.В. Воеводина. М.: Изд-во Московского университета, 2013. 208 с.

6. *Киреев С.А. , Малашонок Г.И.* Тропические вычисления в веб-сервисе MathPartner // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2014. Т. 19. Вып. 2. С. 539–550.

7. *Малашонок Г.И., Валеев Ю.Д., Лапаев А.О.* О выборе алгоритма умножения для полиномов и полиномиальных матриц. Записки научных семинаров ПОМИ. 2009. Т. 372. С. 50–82.

8. *Malaschonok, G.I.* Effective matrix methods in commutative domains. In: D. Krob, A.A. Mikhalev, A.V. Mikhalev (eds.) Formal Power Series and Algebraic Combinatorics. Springer, Berlin. 2000. P. 506–517.

9. *Akritas A.G., Malaschonok G.I.* Computations in Modules over Commutative Domain. Computer Algebra in Scientific Computing. Springer, Berlin. 2007. P. 11–23.

10. *Malaschonok, G.I.* Generalized Bruhat decomposition in commutative domains. International Workshop on Computer Algebra in Scientific Computing, LNCS 8136. Springer, Berlin, Heidelberg. 2013. P. 231–242.

11. *Malaschonok G., A. Scherbinin A.* Triangular Decomposition of Matrices in a Domain. Computer Algebra in Scientific Computing. LNCS 9301, Springer, Switzerland. 2015. P. 290–304.

12. *Malaschonok N.A.* Parallel Laplace method with assured accuracy by symbolic computations. “Computer Algebra in Scientific Computing“. Springer-Verlag, Berlin Heidelberg. 2006. P. 251–260.

13. *Malaschonok N.* An algorithm for symbolic solving of differential equations and estimation of accuracy. Computer Algebra in Scientific Computing. Springer-Verlag Berlin Heidelberg, 2009. P. 213–225.

14. *Malashonok N.A., Rybakov M.A.* Symbolic-Numerical Solution of Systems of Linear Ordinary Differential Equations with Required Accuracy. Programming and Computer Software, 2013. Vol. 39. № 3. P. 150–157.

15. *Малашонок Г.И., Валеев Ю.Д.* Рекурсивное распараллеливание символьно-численных алгоритмов // Вестник тамбовского университета. Серия Естественные и технические науки. Тамбов, 2006. Т. 11. Вып. 4. С. 536–549.

16.*Малашонок Г.И., Аветисян А.И., Валеев Ю.Д., Зуев М.С.* Параллельные алгоритмы компьютерной алгебры // Труды института системного программирования / под ред. В.П. Иванникова. М.: ИСП РАН, 2004. Т. 8. Ч. 2. С. 169–180.

БЛАГОДАРНОСТИ: Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 16-07-00420)

Поступила в редакцию 19 октября 2016 г.

Малашонок Геннадий Иванович, Тамбовский государственный университет им. Г.Р. Державина, г. Тамбов, Российская Федерация, доктор физико-математических наук, профессор кафедры функционального анализа, e-mail: malaschonok@ya.ru

UDC 519.6
DOI: 10.20310/1810-0198-2016-21-6-2026-2041

NEW GENERATION OF SYMBOLIC COMPUTATION SYSTEMS

© **G. I. Malaschonok**

Tambov State University named after G.R. Derzhavin
33 Internatsionalnaya St., Tambov, Russian Federation, 392000
E-mail: malaschonok@ya.ru

We define a new generation of symbolic computation systems - mathematical cloud services that have emerged in the last 10 years. The main part of the article is devoted to describing features of one of these systems, which is called MathPartner. The effect of such systems on the development of many modern technologies, primarily educational technologies. The final section gives an overview of other well-known cloud systems of computer algebra and computational mathematics.

Key words: parallel computing; Cloud math, Math Partner, computer algebra, symbolic computations

REFERENCES

1. *Malaschonok G.I.* Way to Parallel Symbolic Computations.«Oblachnye vychisleniya. Obrazovanie. Issledovaniya. Razrabotka». Moskva, 2011. Available at://www.unicluster.ru/conf/2011/docs/TSU.Malaschonok G.I.pdf

2. *Malashonok G.I.* Rukovodstvo po yazyku «MATHPAR»: uchebnoe posobie. Tambov: Izdatel'skij dom TGU im. G.R. Derzhavina, 2013. 133 c.

3. *Malashonok G.I.* O proekte parallel'noj komp'yuternoj algebry // Vestnik Tambovskogo universiteta. Seriya Estestvennye i tekhnicheskie nauki – Tambov University Review. Series: Natural and Technical Sciencesv, 2009. T. 14. Vyp. 4. S. 744–748.

4. *Malaschonok G.I.* Project of Parallel Computer Algebra // Vestnik Tambovskogo universiteta. Seriya Estestvennye i tekhnicheskie nauki – Tambov University Review. Series: Natural and Technical Sciences, 2010. T. 15. Vyp. 6. S. 1724–1729.

5. *Malashonok G.I., Pereslavtseva O.N., Ivashov D.C.* Parallel'nye simvol'nye vychisleniya // Superkomp'yuternye tekhnologii v nauke, obrazovanii i promyshlennosti / pod redaktsiej V.A. Sadovnichego, G.I. Savina, Vl.V. Voevodina. M.: Izdatel'stvo Moskovskogo universiteta, 2013. 208 s.

6. *Kireev C.A. , Malashonok G.I.* Tropicheskie vychisleniya v veb-servise MathPartner // Vestnik Tambovskogo universiteta. Seriya Estestvennye i tekhnicheskie nauki – Tambov University Review. Series: Natural and Technical Sciences, 2014. T. 19. Vyp. 2. S. 539–550.

7. *Malashonok G.I., Valeev YU.D., Lapaev A.O.* O vybore algoritma umnozheniya dlya polinomov i polinomial'nyh matrits. Zapiski nauchnyh seminarov POMI. 2009. T. 372. S. 50–82.

8. *Malaschonok, G.I.* Effective matrix methods in commutative domains. In: D. Krob, A.A. Mikhalev, A.V. Mikhalev (eds.) Formal Power Series and Algebraic Combinatorics. Springer, Berlin. 2000. P. 506–517.

9. *Akritas A.G., Malaschonok G.I.* Computations in Modules over Commutative Domain. Computer Algebra in Scientific Computing. Springer, Berlin. 2007. P. 11–23.

10. *Malaschonok, G.I.* Generalized Bruhat decomposition in commutative domains. International Workshop on Computer Algebra in Scientific Computing, LNCS 8136. Springer, Berlin, Heidelberg. 2013. P. 231–242.

11. *Malaschonok G., A. Scherbinin A.* Triangular Decomposition of Matrices in a Domain. Computer Algebra in Scientific Computing. LNCS 9301, Springer, Switzerland. 2015. P. 290–304.

12. *Malaschonok N.A.* Parallel Laplace method with assured accuracy by symbolic computations. "Computer Algebra in Scientific Computing". Springer-Verlag, Berlin Heidelberg. 2006. P. 251–260.

13. *Malaschonok N.* An algorithm for symbolic solving of differential equations and estimation of accuracy. Computer Algebra in Scientific Computing. Springer-Verlag Berlin Heidelberg, 2009. P. 213–225.

14. *Malashonok N.A., Rybakov M.A.* Symbolic-Numerical Solution of Systems of Linear Ordinary Differential Equations with Required Accuracy. Programming and Computer Software, 2013. Vol. 39. № 3. P. 150–157.

15. *Malashonok G.I., Valeev YU.D.* Rekursivnoe rasparallelivanie simvol'no-chislennyh algoritmov // Vestnik Tambovskogo universiteta. Seriya Estestvennye i tekhnicheskie nauki – Tambov University Review. Series: Natural and Technical Sciences, 2006. T. 11. Vyp. 4. S. 536–549.

16.*Malashonok G.I., Avetisyan A.I., Valeev YU.D., Zuev M.S.* Parallel'nye algoritmy komp'yuternoj algebry // Trudy instituta sistemnogo programmirovaniya. / Pod red. V.P. Ivannikova. M.: ISP RAN, 2004. T. 8. CH. 2. S. 169–180.

ACKNOWLEDGEMENTS: The work is partially supported by the Russian Fund for Basic Research (project № 16-07-00420).

Received 19 October 2016

Malaschonok Gennadi Ivanovich, Tambov State University named after G.R. Derzhavin, Tambov, the Russian Federation, Doctor of Physics and Mathematics, Professor of the Functional Analysis Department, e-mail: malaschonok@ya.ru

Информация для цитирования:

Малашонок Г.И. Новое поколение систем символьных вычислений // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2016. Т. 21. Вып. 6. С. 2026-2041. DOI: 10.20310/1810-0198-2016-21-6-2026-2041

Malaschonok G.I. Novoe pokolenie sistem simvol'nyh vychislenij [New generation of symbolic computation systems]. *Vestnik Tambovskogo universiteta. Seriya Estestvennye i tekhnicheskie nauki – Tambov University Review. Series: Natural and Technical Sciences*, 2016, vol. 21, no. 6, pp. 2026-2041. DOI: 10.20310/1810-0198-2016-21-6-2026-2041 (In Russian)