Облачный математический сервис «Math Partner» http://mathpar.cloud.unihub.ru

G.I.Malaschonok

Tambov State University

Семинар «Компьютерная алгебра» факультета ВМК МГУ и ВЦ РАН

18 ноября 2015 г.

Введение

Система "Math Partner"доступна на платформе программы “Университетский Кластер“: **http://mathpar.cloud.unihub.ru**. Она:
– является свободной,
– доступна из любого устройства с выходом в Интернет,
– ее язык максимально приближен к естественной математической записи и является расширением распространенного языка LaTeX,
– позволяет хранить и выводить на экран тексты как в исходном виде (txt), так и в подготовленном для публикации виде (pdf)
– позволяет создавать и хранить 2D и 3D графики, анимировать их, визуализировать снятые экспериментально табличные функции и др.
– пользователь обеспечен облачной тетрадью, где может проводить вычисления, читать математические тексты.
– он может создавать и хранить учебные материалы, пользоваться готовыми формулами для проведения самостоятельных вычислений
– ввод и вывод может производить в файлы пользователя
– может проводить вычисления на удаленном суперкомпьютере
– язык может использоваться как простой процедурный язык для создания программ

О языке Mathpar

Языком сервиса является язык Mathpar. При его разработке был учтен опыт применения языков компьютерной алгебы в Mathematica и Maple. Выделим основные отличия языка Mathpar.

1.Язык Mathpar является расширением языка LaTeX, путем добавления операторов присвоения, вычисления, управления, операторов установки окружения и операторов вывода графиков.

2. Имеется группа операторов вычисления, которые предназначены для использования многопроцессорного кластера

3. Имеются операторы задания окружения. Окружение определяет основное числовое множество, типы операций в этом множестве, имена переменных и некоторые константы.

4. Добавлены в оператор присвоения правила, которые позволяют вести запись подобную естественной математической записи.

5. Является процедурным язык высокого уровня. Для знакомства с ним можно пользоваться страницами "Помощи" на сайте, или загрузить "Руководство по языку" .

6. 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.

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.

ПРИМЕР:

“В магазине продаются яблоки по цене ” a=5 “ рублей. В отряде ” n=15 “ учеников. Для похода каждому нужно по” m=1.5 ”кг. Сколько это будет стоить?.” РЕШЕНИЕ: “ c=a*n*m

Операторы присвоения

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

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 $proc1(arg1, arg2, ..)\{\text{оператор}_1; \text{оператор}_2; ..\}$

Операторы управления

Кроме операторов присвоения 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*.

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 результыт компилируются в PDF, как при выводе в языке LaTeX. В частности, все имена, которые начинаются с символа “backslash” пишутся жирным шрифтом “bold”, а сам знак (\) не ставится.

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 $\mathbf{A} * \mathbf{B} - \mathbf{B} * \mathbf{A}$.

Допустимые отступления в записи оператора присвоения

Отметим, что синтаксис допускает следующие отступления от правил в операторе присвоения. Слева от знака равно, в том месте, где должно быть имя переменной, может стоять выражение. В одном операторе может быть несколько знаков "равно". Может отсутствовать знак "равно"и вся левая часть.

Во всех этих некорректных случаях значение выражения, которое стоит справа, будет вычислено и доступно для вывода. Но оно не будет присвоено никакой переменной, если слева нет имени переменной. Все выражения стоящие до правого знака "равно"будут игнорироваться. Если такой оператор является последним, то вычисленное значение попадет в вывод, а если он не последний, то вычисленное значение будет потеряно.

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.

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: "$$*bf* жирный\шрифт$$". 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.

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 так далее.

Числовые множества

Пользователь может выбирать следующие числовые множества:
Z – множество целых чисел $\mathbb{Z}$,
Zp – конечное поле $\mathbb{Z}/p\mathbb{Z}$, характеристика p задается пользователем (в постоянной MOD);
Zp32 – конечное поле $\mathbb{Z}/p\mathbb{Z}$, характеристика p задается постоянной MOD32 которая должна быть меньше, чем 2^{31},
Z64 – подмножество целых чисел $\{z : -2^{63} \leq z < 2^{63}\}$,
Q – множество рациональных чисел $\mathbb{Q}$,
R – множество приближенных действительных чисел, у которых число цифр в мантиссе задается пользователем (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.

Переменные

Пользователь может 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.

Смена окружения

Пользователь может многократно изменять окружение по своему усмотрению. 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().

Постоянные окружения

MOD32 (268435399)– характеристика поля вычетов Zp32: до 2^{31}.
MOD (268435399)– характеристика поля вычетов Zp. Может быть произвольным простым числом.
RADIAN (1)– опеределяет меру углов: радианы (1) или градусы (0).
TIMEOUT (15) – максимальное время вычислений (в сек.)
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 всех известных переменных их значениями (1) или выключает его (0).
FLOATPOS (2) – число знаков после запятой, для печати.
MachineEpsilonR64 (36) – 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) – Число, абсолютное значение которого меньше, чем $10^{-MachineEpsilonR}$ считается машинным нулем.

Основные классы функций
Элементарные функции и математические символы

Определены следующие математические символы:

\i – мнимая единица i,

\e – число e,

\pi – число π,

\infty – знак бесконечности ∞,

\emptyset – пустое множество $\emptyset$

Определены следующие элементарные функии:

\exp() – экспоненциальная функция,

\ln() – натуральный логарифм,

\lg() – десятичный логарифм,

\log(a, b) или log_a(b) – логарифм числа b по основанию a,

\sgrt() – корень квадратный,

\cubrt() – корень кубический,

\rootof(a, n) – корень из числа a степени n,

\sin() – синус,

\cos() – косинус,

\tg() – тангенс,

Элементарные функции (продолжение)

\ctg() – котангенс,
\arcsin() – арксинус,
\arccos() – арккосинус,
\arctg() – арктангенс,
\arcctg() – арккотангенс,
\sh() – гиперболический синус,
\ch() – гиперболический косинус,
\th() – гиперболический тангенс,
\cth() – гиперболический котангенс,
\arcsh() – гиперболический арксинус,
\arcch() – гиперболический арккосинус,
\arctgh() – гиперболический арктангенс,
\arcctgh() – гиперболический арккотангенс,
\sc() – секанс,
\csc() – косеканс,
\arcsec() – арксеканс,
\arccsc() – арккосеканс.

Операции над числами

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

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.

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 драбь,
$\backslash quotientAndRemainder(fr)$ – 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.

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.
$\backslash solveNAE(f_1, f_2, .., f_c)$ – 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:
$\backslash quotientAndRemainder(a, b)$ – 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.

Функции нескольких аргументов

Некоторые функции от двух аргументов:
$\backslash log(a, b)$ – логарифм b по основанию ,
$\backslash rootOf(b, n)$ – 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

Векторы отмечаются квадратными скобками, например, V=[1,2,x,y]. Матрицы – двойными скобками: A=[[1,2],[f,g]]. Для матриц и векторов определены стандартные операции сложения, вычитания, умножения: +, -, *.

Чтобы адресовать элементы матрицы нужно дать им наименование a=\elementOf(A), теперь можно брать отдельные элементы, столбцы и строки матрицы:

a_{i,j} — (i,j)-элемент матрицы A, a_{1,1} – верхний левый угловой элемент;

a_{i,?} — строка i матрицы A;

a_{?,j} — столбец j матрицы A.

v=\elementOf(V). После этого можно обращаться к компонентам вектора-строки: v_{1}, v_{2} и так далее.

\O_{n,m} — нулевая матрица размера $n \times m$;

\I_{n,m} — $n \times m$ матрица с единицами на главной диагонали.

Аналогично создается нулевая вектор-строка:

\O_{n}. Это основной способ создания массива из n объектов в языке Mathpar.

Другие матричные функции в Mathpar:
\charPolynom() — характеристический полином;
\kernel() — ядро оператора (нуль-пространство);
\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 Муррa-Пенроуза;
\closure() или **A**^{**times**} — замыкание, т.е. сумма $I + A + A^2 + A^3 + \ldots$. Для классических алгебр это эквивалентно $(I - A)^{-1}$;
\BruhatDecomposition() — разложение Брюа матрицы.
\LDU() — 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.

Преобразование Лапласа и решение систем линейных дифференциальных уравнений с постоянными коэффициентами

$\laplaceTransform()$ – прямое преобразование Лапласа,
$\inverseLaplaceTransform()$ – обратное преобразование Лапласа,
$\solveLDE(S, J)$ — решение системы линейных дифференциальных уравнений с постоянными коэффициентами S при начальных условиях J. Вот примеры задания S и J:
$\systLDE(\d(x,t) + x - 4y = t\^2\exp(2t),\ 7\d(y,t) - 2x - 2y = 3\exp(3t) + t)$; – пример задания системы дифференциальных уравнений S:

$$\begin{cases} x'_t + x - 4y = t^2 e^{2t}, \\ 7y'_t - 2x - 2y = 3e^{3t} + t. \end{cases}$$

$\initCond(\d(x,t,0,0) = 2, \d(y,t,0,0) = 0)$; – пример задания начальных условий J для решения системы дифференциальных уравнений.

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 матриц A – это левая часть системы дифференциальных уравнений после преобразования Лапласа, вектор B – правая часть системы после преобразования Лапласа и C – это матрица начальных условий.

Функции теории вероятностей и математической статистики

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.

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]].

Функции дискретных случайных величин:

$\backslash mathExpectation()$ – математическое ожидание,

$\backslash dispersion()$ – дисперсия,

$\backslash meanSquareDeviation()$ – среднее квадратичное отклонение,

$\backslash addQU(a, b)$ – сумма ,

$\backslash multiplyQU(a, b)$ – произведение ,

$\backslash covariance(a, b)$ – коэффициент ковариации ,

$\backslash correlation(a, b)$ – коэффициент корреляции ,

$\backslash simplifyQU()$ упрощение записи дискретной случайной величины, если в ней присутствуют повторяющиеся значения.

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.

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.

Вычисления в тропической математике
Идемпотентные алгебры

Кроме классических числовых алгебр с операциями + , -, * и операцией "делить" для полей, можно производить вычисления в идемпотентных алгебрах (в тропической математике). 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:

$$x \leq y \Leftrightarrow x \oplus y = y.$$

Поэтому нейтральный элемент по сложению $\not\vdash$ 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.

Идемпотентные полукольца

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 $c = 9,\ d = 11$, так как произошло назначение операций:

$$+ \rightarrow \textit{максимум} \text{ и } * \rightarrow \textit{сложение}.$$

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 $1 + a + a^2 + a^3 + \ldots$

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 $Ax \leq x$;
$\backslash BellmanInequality(A, b)$ — решение неравенства $Ax \oplus b \leq x$.
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.

Построение 2D и 3D графиков

Построение графиков функций на плоскости

Графическое окружение задается командой set2D(). Если у команды set2D() нет параметров, то границы для графиков расчитываются автоматически, для явных функций выбирается интервал [0,1] по оси абсцисс, а наименования осей координат будет X и Y, соответственно.

Можно задавать такие параметры для команды set2D():
$\backslash set2D(x0, x1)$,
$\backslash set2D(x0, x1,' title')$,
$\backslash set2D(x0, x1, y0, y1)$,
$\backslash set2D(x0, x1, y0, y1,' title')$,
$\backslash set2D(x0, x1,' title',' nameOX',' nameOY')$,
$\backslash set2D(x0, x1, y0, y1,' title',' nameOX',' nameOY')$. 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.

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

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.

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.

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).

Сервис пользователя
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.

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 раннее 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.

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 паяти 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 сохраняться все те объекты, у которых не отмеченны имена.

Сервис для вычислений на кластере

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 меняеть объем памяти, который приходится на одно ядро.

Операторы для параллельных вычислений

В настоящий момент подключены следующие параллельные операторы:

Для классических пространств (Z[x]):

$\backslash adjointDetPar(A)$ – вычисление обратной матрицы (присоединенной матрицы и определителя).

Для тропических пространств (R64MaxPlus[x]):

$\backslash BellmanEquationPar(A)$,

$\backslash BellmanEquationPar(A, b)$,

$\backslash BellmanInequalityPar(A)$,

$\backslash BellmanInequalityPar(A, b)$.

Планируется, что в ближайшее время будут введены многие матричные операции, факторизация полиномов многих переменных и решение систем линейных дифференциальных уравнений с постоянными коэффициентами.

СПАСИБО ЗА ВНИМАНИЕ