

Тамбовский государственный университет
имени Г.Р.Державина

Г.И. Малашонок, А.А. Бетин, М.А. Рыбаков, Р.А. Смирнов

ПАРАЛЛЕЛЬНАЯ КОМПЬЮТЕРНАЯ АЛГЕБРА
Часть 3

Учебное пособие

Тамбов 2012

УДК 512.715 + 519.61 (075.8)

ББК 22.143 я 73

М 18

Рецензенты:

доктор физико-математических наук В.А. Федоров;

доктор физико-математических наук А.А. Шибков

**Малашонок Г.И., Бетин А.А., Рыбаков М.А.,
Смирнов Р.А.**

М 18 Параллельная компьютерная алгебра. Учеб. пособие.
Ч.3./ Г.И. Малашонок, А.А. Бетин, Рыбаков М.А., Смирнов Р.А.; ФГБОУ ВПО «Тамбовский государственный университет имени Г.Р.Державина». Тамбов: Изд. дом ТГУ им. Г.Р. Державина, 2012. 91 с.

Учебное пособие предназначено для освоения web-системы компьютерной математики Mathpar. Эта система позволяет проводить символично-численные расчеты в режиме онлайн и дает возможность использовать суперкомпьютер для выполнения масштабных вычислений.

Излагаемый материал дополняет существующую учебную литературу для пользователей системам компьютерной алгебры.

Пособие предназначено для математиков, специалистов в естественных и гуманитарных науках, экономистов, инженеров, аспирантов, студентов университетов и школьников старших классов.

А.А. Бетин, Г.И. Малашонок,
М.А. Рыбаков, Р.А. Смирнов

© ФГБОУВПО «Тамбовский государственный университет имени Г.Р.Державина», 2012

ОГЛАВЛЕНИЕ

Предисловие	13
1 Знакомство с системой	17
1.1. Ввод данных, запуск задачи	18
1.2. Математические функции	19
1.2.1. Константы	19
1.2.2. Функции одного аргумента	19
1.2.3. Функции двух аргументов	20
1.3. Действия с функциями	21
1.4. Решение алгебраических уравнений	23
1.5. Векторы и матрицы	24
1.6. Генерация случайных элементов	27
1.6.1. Генерация чисел	27
1.6.2. Генерация полиномов	28
1.6.3. Генерация матриц	28
1.7. Контрольные задания	29
2 Построение 2D и 3D графиков	31
2.1. Построение графиков функций	31
2.1.1. Явное задание функции	31
2.1.2. Функции, заданные параметрически	32
2.1.3. Функции, которые заданы таблицей значений	33
2.1.4. Построение разных графиков функций в одной системе координат	34
2.2. Построение 3D графиков функций	35
2.3. Контрольные задания	37

3	Выбор окружения для математических объектов	39
3.1.	Окружение	39
3.2.	Числовые множества	40
3.3.	Определение нескольких числовых множеств	41
3.4.	Групповые алгебры	41
3.5.	Идемпотентные алгебры. Тропическая математика	42
3.6.	Константы	42
3.7.	Контрольные задания	43
4	Функции одной и нескольких переменных	45
4.1.	Математические функции	45
4.1.1.	Константы	45
4.1.2.	Функции одного аргумента	45
4.1.3.	Функции двух аргументов	46
4.2.	Вычисление значений функции в точке	47
4.3.	Подстановка выражений в функции	49
4.4.	Вычисление предела функции в точке	49
4.5.	Дифференцирование функций	51
4.6.	Интегрирование композиций элементарных функций	52
4.7.	Контрольные задания	53
5	Ряды	55
5.1.	Контрольные задания	58
6	Решение дифференциальных уравнений и их систем	59
6.1.	Решение дифференциальных уравнений	59
6.2.	Решение систем дифференциальных уравнений	62
6.3.	Контрольные задания	66
7	Полиномиальные вычисления	67
7.1.	Вычисление значения полинома в точке	67
7.2.	Суммирование полинома по переменным. Геометрические прогрессии	68
7.3.	Вычисление базисов Гребнера	69
7.4.	Контрольные задания	70

8	Матричные функции	71
8.1.	Вычисление транспонированной матрицы	71
8.2.	Вычисление обратной и присоединенной матрицы . .	72
	8.2.1. Вычисление обратной матрицы	72
	8.2.2. Вычисление присоединенной матрицы	72
8.3.	Вычисление определителя матрицы	73
8.4.	Вычисление сопряженной матрицы	75
8.5.	Вычисление обобщенной обратной матрицы	75
8.6.	Вычисление ядра оператора и эшелонной формы . .	76
	8.6.1. Вычисление эшелонной формы матрицы	76
	8.6.2. Вычисление ядра оператора	77
8.7.	Вычисление характеристического полинома матрицы	78
8.8.	LDU-разложение	79
8.9.	Разложение Брюа	81
8.10.	Контрольные задания	82
9	Функции теории вероятностей и математической статистики	83
9.1.	Функции для работы с дискретными случайными величинами	83
9.2.	Функции для выборок	86
9.3.	Контрольные задания	87
10	Операторы управления. Процедурное программирование	89
10.1.	Процедуры и функции	89
10.2.	Операторы ветвления и циклов	90
10.3.	Контрольные задания	90
11	Вычисления на суперкомпьютере	91
11.1.	Параллельные полиномиальные вычисления	92
11.2.	Параллельные матричные вычисления	93
11.3.	Контрольные задания	93
12	Список операторов	95
	Литература	99

Предисловие

В третьей части учебного пособия «Параллельная компьютерная алгебра» излагаются основы программирования с использованием веб-системы параллельной компьютерной алгебры Mathpar.

Система Mathpar позволяет оперировать с функциями и функциональными матрицами, получать как точные численно-аналитические решения, так и решения, у которых числовые коэффициенты в аналитических выражениях имеют требуемую степень точности.

Для первого знакомства с системой, освоения простых функциональных и символьно-численных операций достаточно изучить первую главу. В первой главе описываются ввод данных и запуск вычислений. Даются обозначения для элементарных функций, таких как логарифм, синус, косинус и т.д., и констант — π , e , i , а также констант, которые необходимы для задания числовых множеств. Описываются способы задания вектора и матрицы, арифметические операции с векторами, команды генерации случайных чисел, полиномов и матриц, команды для решения алгебраических уравнений. Для всех команд приведены примеры и продемонстрированы результаты вычислений.

Вторая глава посвящена построению графиков функций. Система Mathpar позволяет строить графики функций, заданных явно, параметрически или по точкам. Причем, можно выполнить построение нескольких графиков в одной системе координат. В этой главе приводятся команды для построения графиков и примеры использования команд.

В третьей главе описываются способы задания окружения в системе Mathpar, т.е. того пространства, в котором будут определяться математические объекты. В любой момент пользователь может сменить окружение, задав новое алгебраическое пространство. Перемещение из некоторого окружения в текущее, как правило, должно выполняться явно. В некоторых случаях такое преобразование к текущему окружению происходит автоматически.

В четвертой главе описаны команды для задания математических функций одной или нескольких переменных, их композиций, вычисления значений функции в точке, подстановки выражений в функцию, вычисления предела функции в точке, символьного ин-

тегрирования композиций элементарных функций. Приводятся примеры выполнения рассмотренных команд. К каждому примеру даются результаты вычислений в системе Mathpar.

Пятая глава посвящена заданию и действиям с рядами в системе Mathpar. Рассматриваются способы задания ряда. Даются команды для сложения, вычитания, умножения двух рядов и для разложения функции в ряд Тейлора с определенным количеством членов ряда. Рассматриваются примеры.

В шестой главе описаны команды для решения обыкновенных дифференциальных уравнений и систем, а также систем дифференциальных уравнений с частными производными.

Седьмая глава посвящена полиномиальным вычислениям. Рассматриваются команды для вычисления значения полинома в точке, суммирования полинома по переменным, вычисления базиса Гребнера полиномиального идеала над рациональными числами. Для каждой команды приводится пример в системе Mathpar.

В восьмой главе описываются матричные функции — вычисление транспонированной матрицы, определителя матрицы, присоединенной и обратной матриц, эшелонной формы матрицы, ядра оператора, характеристического полинома матрицы и другие.

Девятая глава посвящена функциям теории вероятностей и математической статистики. Описывается задание дискретной случайной величины, команды для вычисления математического ожидания дискретной случайной величины, дисперсии, среднего квадратичного отклонения, суммы, произведения двух дискретных случайных величин, коэффициента ковариации дискретных случайных величин, коэффициента корреляции, построения многоугольника распределения и функции распределения дискретной случайной величины. В этой главе рассматриваются команды для задания выборок и для вычисления функций для них: выборочное среднее, выборочная дисперсия, коэффициент ковариации и коэффициент корреляции для двух выборок.

Система Mathpar позволяет создавать свои процедуры и функции. Десятая глава посвящена описанию способов создания процедур в системе. Приведены примеры написания процедур с использованием операторов ветвления и циклов.

В одиннадцатой главе описываются команды, которые управляют вычислениями на суперкомпьютере. Для решения вычисли-

тельных задач, которые требуют большого времени вычислений или больших объемов памяти, разработаны специальные функции, которые предоставляют пользователю ресурсы суперкомпьютера. При использовании этих функций вычисления производятся не на одном процессоре, а на выделенном множестве ядер суперкомпьютера, количество которых заказывает пользователь. Это такие операции, как вычисление базиса Гребнера, присоединенной матрицы, ступенчатого вида матрицы, обратной матрицы, определителя, ядра линейного оператора, характеристического полинома и др.

В двенадцатой главе приведен список основных операторов системы Mathpar.

Отметим, что главное отличие от других систем символьных вычислений — это высокая производительность Mathpar при решении задач, в которых проводятся функциональные и символьно-численные вычисления с математическими объектами большого размера. Это достигается за счет использования уникальных алгоритмов для параллелизации процесса вычисления и выполнения его на суперкомпьютере.

Система может оказаться полезной во многих областях науки и техники, где требуется высокая точность при решении матричных задач большого размера.

Авторы выражают благодарность всем, кто оказал помощь в создании данного учебного пособия. Большую работу по редактированию и разработке заданий выполнила ассистент кафедры математического анализа Оксана Переславцева. В создании пособия участвовали аспиранты кафедры математического анализа Иван Борисов, Евгений Дубовицкий, Светлана Тарарова, Дмитрий Шляпин, Дмитрий Ивашов, Юрий Юрин, а также студенты 45 группы Олег Бобков, Сергей Киреев, Владислав Лукашин, Юлия Матвеева, Лиляна Раецкая, Александр Почтарьков, Антон Щербинин.

Излагаемый материал дополняет существующую учебную литературу по применениям систем компьютерной алгебры и параллельным вычислениям.

Учебное пособие разрабатывалось при частичной поддержке программы ФЦП, проект 2.1.1/10437.

Глава 1

Знакомство с системой

Система `Mathpar` предназначена для выполнения численно-аналитических вычислений, построения 2D и 3D графиков. Система позволяет оперировать с функциями и функциональными матрицами, получать как точные численно-аналитические решения, так и решения, у которых числовые коэффициенты в аналитических выражениях имеют требуемую степень точности. Имеется возможность создания некоммутативных и идемпотентных объектов.

Входной язык системы, который описывается ниже, носит название языка `ATeX`. Он представляет собой некоторый активный диалект языка `TeX`, который позволяет выполнять вычисления. Задание и результат вычисления, представлены на языке `ATeX`. Однако сразу после вычислений пользователь видит весь текст в pdf-изображении, как принято в научных и технических публикациях.

Этот результат может быть использован дальше несколькими способами:

- 1) Кликнуть по нему мышкой. Он вернется к виду языка `ATeX`. Затем продолжить редактирование и решение задач.

- 2) Кликнуть по нему правой кнопкой мышки. Появится выпадающее меню. Поле «`Show Source`» позволяет получить текст в виде стандартной записи в языке `TeX` или в языке `MathML`. Для перехода от одного языка к другому имеется настройка в поле «`Format`». Полученный текст на языке `TeX` или `MathML` можно скопировать и поместить в `TeX`-файл или `html`-файл для публикации.

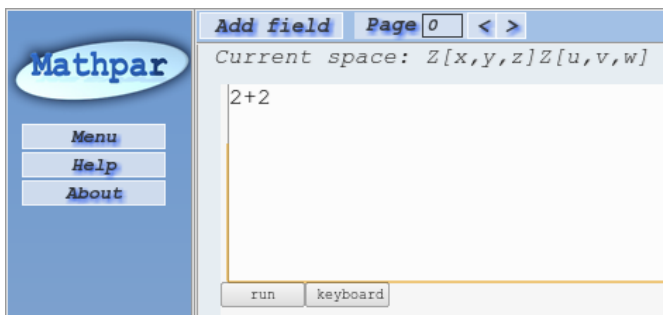


Рис. 1.1. Ввод данных в рабочее поле

1.1. Ввод данных, запуск задачи

В верхней части экрана находится поле ввода, куда нужно вводить математические выражения. Для запуска введенной задачи надо нажать на кнопку `run`.

Пример показан на рис. 1.1.

В левой части экрана находится кнопка `help` — это переход к разделам помощи. На всех страницах помощи все поля активные, содержащиеся там примеры можно запускать. Можно копировать тексты из примеров и переносить их в поле ввода пользователя.

При вводе задачи необходимо разделять вводимые выражения и операторы используя точку с запятой (;). Не ставить точку с запятой можно только один раз — после последнего оператора.

Для вывода результатов необходимо использовать команду `\print()`, где в качестве аргументов, разделенных запятыми, следует указывать имена тех выражений, которые требуется вывести. Если среди команд не встретился оператор печати `\print()` или какой-нибудь другой оператор вывода (`\plot()`, `\prints()` и т. д.), то будет выводиться результат, полученный в последнем операторе.

Все команды или операторы должны начинаться с символа «back slash» (`\`).

Для использования комментариев необходимо помещать их между символами процент (`%`).

Для запуска задачи нужно нажать кнопку `run` или в поле ввода нажать сочетание клавиш `Ctrl+Enter`.

Кнопка `Add Field` предназначена для добавления полей ввода. Для удаления поля ввода нужно нажать сочетание клавиш *Ctrl+Del* в этом поле.

В правом нижнем углу под полем ввода находятся две надписи «Substitution» и «Expand» с квадратными окошками активизации. Они предназначены для управления режимами автоматической подстановки и упрощения. По умолчанию эти режимы включены. При необходимости пользователь может выключить каждый этих режимов по своему усмотрению. Выключение первого из них приведет к тому, что объекты, получившие имена, не будут подставляться в последующих операторах, а будут сохраняться в виде абстрактных имен. Выключение второго — к тому, что раскрытие скобок и приведение подобных не будет происходить автоматически.

Кнопка `Keyboard` предназначена для вывода на экран поля клавиатуры. Это нужно тем устройствам, которые не имеют собственную клавиатуру. Под клавиатурой кнопка `Mobile` включает красный курсор в поле ввода, который очень полезен при наборе с мобильных устройств.

1.2. Математические функции

Приняты следующие обозначения для элементарных функций и констант.

1.2.1. Константы

`\i` — мнимая единица;
`\e` — основание натурального логарифма;
`\pi` — число π , то есть отношение длины окружности к диаметру;
`\infty` — знак бесконечности.

1.2.2. Функции одного аргумента

`\ln` — натуральный логарифм;
`\lg` — десятичный логарифм;
`\sin` — синус;
`\cos` — косинус;
`\tg` — тангенс;

`\ctg` — котангенс;
`\arcsin` — арксинус;
`\arccos` — арккосинус;
`\arctg` — арктангенс;
`\arcctg` — арккотангенс;
`\sh` — синус гиперболический;
`\ch` — косинус гиперболический;
`\th` — тангенс гиперболический;
`\cth` — котангенс гиперболический;
`\arcsh` — арксинус гиперболический;
`\arcch` — арккосинус гиперболический;
`\arth` — арктангенс гиперболический;
`\arccth` — арккотангенс гиперболический;
`\exp` — экспонента;
`\sqrt` — корень квадратный;
`\abs` — абсолютное значение для действительных чисел, модуль для комплексного числа;
`\sign` — знак числа. Возвращает 1, 0, -1, когда число положительное, ноль или отрицательное, соответственно;
`\unitStep(x)` — это функция, которая при $x \geq 0$ принимает значение 1, а при $x < 0$ принимает значение 0;
`\fact` — факториал. Определен для целых положительных чисел. Равносильная запись — «n!».

1.2.3. Функции двух аргументов

`^` — степень;
`\log` — логарифм от функции по указанному основанию;
`\root(n, x)` — корень степени n из x;
`\Gamma` — функция Гамма;
`\Gamma2` — функция Гамма 2;
`\binomial` — число сочетаний.

Примеры:

```

SPACE=R64[x, y];
f1=\sin(x);
f2=\sin(\cos(x+\tg(y)));
f3=\sin(x^2)+y;
\print(f1, f2, f3);

```

Результат выполнения:

```
SPACE = R64[x, y];  
f1 = sin(x);  
f2 = sin(cos(x + tg(y)));  
f3 = sin(x^2) + y.
```

1.3. Действия с функциями

Для перечисленных выше функций и их композиций в системе Mathpar можно вычислить значение функции в точке, подставить выражения в функцию вместо аргументов, вычислить предел функции, ее производную. Для этого определены следующие команды:

- Для вычисления значения функции в точке необходимо выполнить команду `\value(f, [var1, var2, ..., varn])`, где f — функция, а $var1, var2, \dots, varn$ — значения соответствующих переменных кольца.
- Для подстановки выражений в функцию необходимо выполнить команду `\value(f, [func1, func2, ..., funcn])`, где f — это функция, $func1, func2, \dots, funcn$ — функции, которые подставляются вместо соответствующих переменных.
- Для вычисления предела функции в точке необходимо выполнить команду `\lim(f, var)`, где f — это функция, а var — точка, в которой требуется найти предел.
- Для вычисления производной функции f по переменной u из кольца $\mathbb{Z}[x, y, z]$ необходимо выполнить команду `\D(f, y)`. Для нахождения смешанной производной первого порядка от функции f существует команда `\D(f, [x, y])`, для нахождения производной высших порядков нужно использовать команду `\D(f, [x^k, z^m, y^n])`, где k, m, n указывают, какого порядка по соответствующей переменной вычисляется производная.

Примеры:

```
1. SPACE=R[x, y];  
f=\sin(x^2+tg(y^3+x));  
g=\value(f, [1, 2]);  
\print(g);
```

Результат выполнения:

```
in: SPACE = R[x, y];  
    f = sin(x^2 + tg(y^3 + x));  
    g = value(f, [1, 2]);  
    print(g);  
out: g = 0.52.
```

```
2. SPACE=Z[x, y];  
f=x+y;  
g=f^2;  
r=\value(f, [x^2, y^2]);  
\print(g, f, r);
```

Результат выполнения:

```
in: SPACE = Z[x, y];  
    f = x + y;  
    g = f^2;  
    r = value(f, [x^2, y^2]);  
    print(g, f, r);  
out: g = y^2 + 2yx + x^2;  
    f = y + x;  
    r = x^2 + y^2.
```

```
3. SPACE=R64[x];  
f=\sin(x)/x;  
g=\lim(f, 0);  
\print(g);
```

Результат выполнения:

```
in: SPACE = R64[x];  
    f = sin(x)/x;  
    g = lim(f, 0);  
    print(g);  
out: g = 1.00.
```

```
4. SPACE=Z[x, y];  
f=\sin(x^2+ \tg(y^3+x));  
h= \D(f, y);  
\print(h);
```

Результат выполнения:

```
in: SPACE = Z[x, y];  
    f = sin(x^2 + tg(y^3 + x));  
    h = D(f, y);  
    print(h);  
out: h = 3y^2 cos(x^2 + tg(y^3 + x)) / (cos(y^3 + x))^2.
```

1.4. Решение алгебраических уравнений

Для решения алгебраических уравнений нужно выполнить команду `\solve`. Ниже используется команда настройки окружения «`FLOATPOS=N`». Она устанавливает число десятичных знаков после запятой (N), которые должны появиться при выводе числового результата приближенных вычислений. Она связана не с процессом вычислений, а только с выводом. По умолчанию $FLOATPOS = 2$.

Примеры:

```
1. SPACE=R64[x];  
b=\solve(x^2-5x+6=0);
```

Результат выполнения:

```
in: SPACE = R64[x];  
    b = solve(x^2 - 5x + 6 = 0);  
out: [2.00, 3.00].
```

```
2. SPACE=R64[x];  
FLOATPOS=6;  
b=\solve(x^4 + 2x + 1=0);
```

Результат выполнения:

```
in: SPACE = R64[x];  
    FLOATPOS = 6;  
    b = solve(x^4 + 2x + 1 = 0);  
out: [(0.771845 + 1.115143i),  
      (0.771845 - 1.115143i), -0.543689, -1.000000].
```

```
3. SPACE=R64[x];  
FLOATPOS=0;  
b=\solve(x^3+3x^2+3x+1=0);
```

Результат выполнения:

in: $SPACE = R64[x]$;

$FLOATPOS = 0$;

$b = solve(x^3 + 3x^2 + 3x + 1 = 0)$;

out: $[-1, -1, -1]$.

1.5. Векторы и матрицы

Для задания вектора нужно перечислить его элементы в квадратных скобках. Так задаются вектор-строки. Для задания матрицы нужно заключить в квадратные скобки список ее вектор-строк, например, $A = [[1, 2], [3, 4]]$.

Элемент матрицы можно получить, указав номер строки и столбца в двух нижних индексах у матрицы, а элемент вектора можно получить, указав его номер нижним индексом у вектора: $A_{\{i, j\}}$ для матриц и $A_{\{i\}}$ для вектора.

Можно получить строку матрицы в виде вектор-строки и столбец матрицы в виде вектор-столбца. Вектор-строка получается указанием номера строки в первом индексе и знака «вопрос» во втором индексе, например, $A_{\{i, ?\}}$. Вектор-столбец получается указанием номера столбца во втором индексе и знака «вопрос» в первом индексе, например, $A_{\{?, j\}}$.

Имена некоммутативных объектов, например, матриц и векторов, надо писать с заглавной латинской буквы.

Для обозначения нулевой и единичной матрицы используются заглавные буквы O и I , у которых указаны два индекса, обозначающих число строк и столбцов. С помощью символа I можно создавать прямоугольные матрицы любого размера, у которых элементы на главной диагонали равны 1, а остальные элементы нулевые. Например, $I_{\{2, 3\}}$ и $O_{\{2, 2\}}$ обозначают матрицы $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ и $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$. Можно задавать нулевые векторы, указывая в индексе число элементов: $O_{\{3\}}$ обозначает вектор $[0, 0, 0]$, а $I_{\{3\}}$ обозначает вектор $[1, 0, 0]$.

Вектор-столбец может быть образован транспонированием вектор-строки, например, $D = [7, 2, 3]^T$ — это вектор-столбец из трех элементов.

Арифметические операции обозначаются обычными знаками «+», «-», «*».

Примеры:

```
1. SPACE=Z[x];
A=[[x, 4], [y, 5]];
V=[x, y, 1, 2, x^6];
\print(A, V);
```

Результат выполнения:

in: $SPACE = Z[x]$;

$A = [[x, 4], [y, 5]]$;

$V = [x, y, 1, 2, x^6]$;

$print(A, V)$;

out: $A = \begin{pmatrix} x & 4 \\ y & 5 \end{pmatrix}$;

$V = [x, y, 1, 2, x^6]$.

```
2. A=[[3, 4], [3, 1]];
B=[[2, 5], [4, 7]];
C=A+B;
```

```
G=A-B;
```

```
T=A*B;
```

```
\print(C, G, T);
```

Результат выполнения:

in: $SPACE = Z[x]$;

$A = \begin{pmatrix} 3 & 4 \\ 3 & 1 \end{pmatrix}$;

$B = \begin{pmatrix} 2 & 5 \\ 4 & 7 \end{pmatrix}$;

$C = A + B$;

$G = A - B$;

$T = A * B$;

$print(C, G, T)$;

out: $C = \begin{pmatrix} 5 & 9 \\ 7 & 8 \end{pmatrix}$;

$G = \begin{pmatrix} 1 & -1 \\ -1 & -6 \end{pmatrix}$;

$T = \begin{pmatrix} 22 & 43 \\ 10 & 22 \end{pmatrix}$.

```

3. SPACE=Z[x];
A=[[1, 4], [-4, 5]];
det=A_{1, 1}*A_{2, 2}-A_{1, 2}*A_{2, 1};
\print(det);

```

Результат выполнения:

```

in: SPACE = Z[x];
    A =  $\begin{pmatrix} 1 & 4 \\ -4 & 5 \end{pmatrix}$ ;
    det = A_{1,1} * A_{2,2} - A_{1,2} * A_{2,1};
    print(det);
out: det = 21.

```

```

4. SPACE=R64[x];
A=[[3, 4], [3, 1]];
A_{1, 2}=-7x;
A_{2, 2}=x;
A_{1, 1}=7x;
A_{2, 1}=A_{2, 1}*x;
\print(A);

```

Результат выполнения:

```

in: SPACE = R64[x];
    A =  $\begin{pmatrix} 3 & 4 \\ 3 & 1 \end{pmatrix}$ ;
    A_{1,2} = -7x;
    A_{2,2} = x;
    A_{1,1} = 7x;
    A_{2,1} = A_{2,1} * x;
    print(A);
out: A =  $\begin{pmatrix} 7.00x & -7.00x \\ 3.00x & x \end{pmatrix}$ .

```

```

5. SPACE=Z[x, y];
A=[[x^2, y], [4, x+y]];
B=A_{1, ?};
C=A_{?, 2}; h=B_{2, 1}*C_{1, 2};
\print(B, C, h);

```

Результат выполнения:

```

in: SPACE = Z[x, y];

```

$$A = \begin{pmatrix} x^2 & y \\ 4 & x + y \end{pmatrix};$$

$$B = A_{1,?};$$

$$C = A_{?,2};$$

$$h = B_{2,1} * C_{1,2};$$

$$\text{print}(B, C, h);$$

out: $B = \begin{pmatrix} x^2 \\ 4 \end{pmatrix};$

$$C = \begin{pmatrix} 4 & y + x \end{pmatrix};$$

$$h = (4y + 4x).$$

```
6. SPACE=Z[x, y];
A=3x*I_{2, 2};
B=O_{3, 3};
\print(A, B);
```

Результат выполнения:

```
in: SPACE = Z[x, y];
A = 3x * I_{2,2};
B = O_{3,3};
print(A, B);
```

out: $A = \begin{pmatrix} 3x & 0 \\ 0 & 3x \end{pmatrix};$

$$B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

1.6. Генерация случайных элементов

В системе Mathmag есть возможность генерации случайных чисел, полиномов и матриц.

1.6.1. Генерация чисел

Для того чтобы получить случайное число, необходимо выполнить команду:

$\backslash\mathbf{randomNumber}(k)$, где в аргументе k указывается количество бит случайного числа.

Пример:

```
a=\randomNumber(10);  
b=\randomNumber(100);  
\print(a, b);
```

Результат выполнения:

```
a = 944;  
b = 850800798881527094755736477974.
```

1.6.2. Генерация полиномов

Для того чтобы создать случайный полином от s переменных, необходимо выполнить команду:

`\randomPolynom($d1, d2, \dots, ds, dens, bits$)`, где $dens$ — плотность полинома, а $bits$ — количество бит коэффициентов, $d1, d2, \dots, ds$ означают старшие степени переменных.

Пример:

```
f=\randomPolynom(4, 4, 10, 5);  
g=\randomPolynom(4, 4, 10, 5);  
h=f+g;  
e=\expand(h);  
\print(f, g, h, e);
```

Результат выполнения:

```
f = y3x3;  
g = 10yx3 + 2y;  
h = y3x3 + 10yx3 + 2y;  
e = y3x3 + 10yx3 + 2y.
```

1.6.3. Генерация матриц

Для того чтобы получить случайную числовую матрицу, необходимо выполнить команду:

`\randomMatrix($m, n, dens, bits$)`, где m — количество строк в матрице, n — количество столбцов матрицы, $dens$ — это плотность матрицы, $bits$ — количество бит ее коэффициентов.

Для того чтобы получить случайную полиномиальную матрицу, необходимо выполнить команду:

`\randomMatrix($m, n, dens, d1, d2, \dots, ds, pol_dens, pol_bits$)`, где

m — количество строк в матрице, n — количество столбцов матрицы, $dens$ — это плотность матрицы, $d1, d2, \dots, ds$ — наибольшие степени переменных полиномов, pol_dens — плотность полиномов, pol_bits — количество бит коэффициентов полиномов.

Пример:

```
matr_n=\randomMatrix(4, 4, 100, 5);
matr_p=\randomMatrix(2, 2, 100, 1, 1, 4, 25, 3);
\print(matr_n, matr_p);
```

Результат выполнения:

$$matr_n = \begin{pmatrix} 22 & 2 & 10 & 28 \\ 23 & 28 & 1 & 19 \\ 30 & 24 & 19 & 12 \\ 27 & 22 & 22 & 17 \end{pmatrix};$$

$$matr_p = \begin{pmatrix} 6z^3x + 7z^3 + 5z^2 + 3y & 7z^4x + 2z^4 + 7zyx + 5x \\ z^4yx + 2zy + 7y + 7x + 4 & 7z^2x + 7zx + z + 6x \end{pmatrix}.$$

1.7. Контрольные задания

1. В системе Mathpraг вычислите:

– $\ln 5, \sin 5, \cos 3, \operatorname{ctg} 7, \operatorname{arctg} 1, \operatorname{sh} 0, \operatorname{arcch} 0.5, \exp 8, 12!, \sqrt{100};$

– $\sqrt{\sin^2(5x - 1) + \exp x / \cos(2x)}$ при $x = 0.1, x = 1;$

– $\log_3 8, \sqrt[3]{50};$

– значение функции $f = \sin(\cos(x + \tan(y)))$ при $x = 0.2$ и $y = 1.$

2. Создайте два случайных вектора равной длины. Найдите их сумму и произведение.

Глава 2

Построение 2D и 3D графиков

2.1. Построение графиков функций

Система Mathpraq позволяет строить графики функций, заданных явно, параметрически или по точкам. Можно выполнить построение нескольких графиков в одной системе координат.

Три последних строчных аргумента можно использовать для задания наименований осей координат и названия всего графика. Ниже приведен пример, в котором введены такие строки: « x », « y », «Functions f1, f2, f3, f4».

2.1.1. Явное задание функции

Для построения графика функции $f = f(x)$ используется команда:

`\plot(f, [x0, x1, y0, y1])`, где $[x0, x1]$ — интервал по оси OX , $[y0, y1]$ — интервал по оси OY . Сокращенный вариант: `\plot(f, [x0, x1])`.

Полученный график можно загрузить с сайта. Для этого под полем графика нужно кликнуть на кнопку [Download](#). Файл с графиком будет загружен на компьютер пользователя.

Примеры:

```
1. f=x^2+\tg(x^2-1);  
p1=\plot(f, [-10, 10, -10, 10]);
```

Результат выполнения:

```
in: f = x2 + tg(x2 - 1);
```

```
p1 = plot(f, [-10, 10, -10, 10]);
```

out: рис. 2.1.

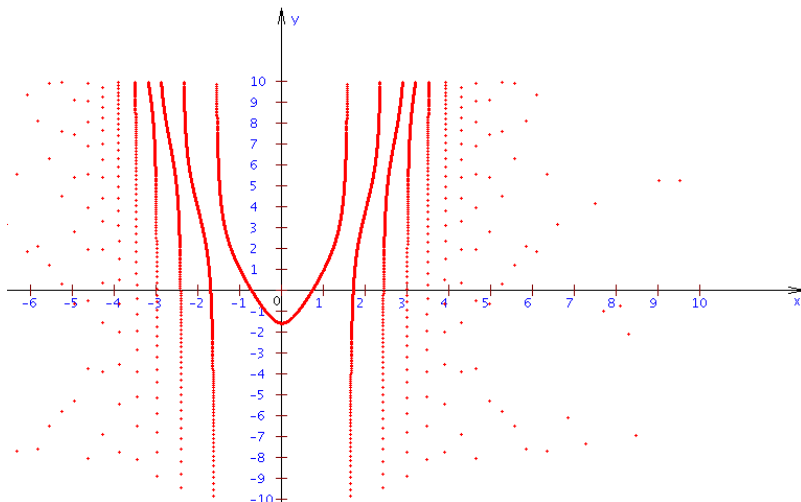


Рис. 2.1. График функции $f = x^2 + \text{tg}(x^2 - 1)$

```
2. f = \sin(x);
```

```
p3 = \plot([f, \tg(x)], [-10, 10, -10, 10]);
```

Результат выполнения:

```
in: f = sin(x);
```

```
p3 = plot([f, tg(x)], [-10, 10, -10, 10]);
```

out: рис. 2.2.

2.1.2. Функции, заданные параметрически

Для построения графиков функций, которые заданы параметрически, необходимо выполнить команду:

`\paramPlot([f, g], [t0, t1])`, где $f = x(t)$, $g = y(t)$ — функция, заданная параметрически, $[t0, t1]$ — интервал значений для изменения параметра.

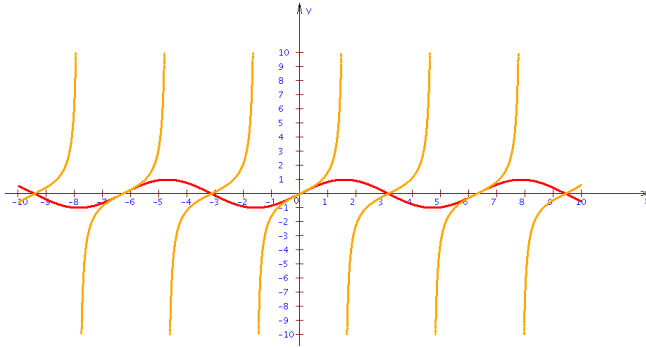


Рис. 2.2. Графики функций $f = \sin(x)$ и $g = \operatorname{tg}(x)$

Пример:

```
g=\sin(x);
k=\cos(x);
f=\paramPlot([g, k], [-10, 10]);
```

Результат выполнения:

```
in: g = sin(x);
    k = cos(x);
    f = paramPlot([g, k], [-10, 10]);
```

out: рис. 2.3.

2.1.3. Функции, которые заданы таблицей значений

Для построения графика функции, заданной таблицей пар координат точек на графике $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, необходимо выполнить команду:

```
\tablePlot([x1, x2, x3, ..., xn], [y1, y2, y3, ..., yn], [xmin, xmax,
ymín, ymax]),
```

где $[xmin, xmax]$ — интервал по оси OX , $[ymín, ymax]$ — интервал по оси OY .

Пример:

```
p=\tablePlot([[0, 1, 4, 9, 16, 25], [0, 1, 2, 3, 4, 5]],
              [-10, 10, -10, 10]);
```

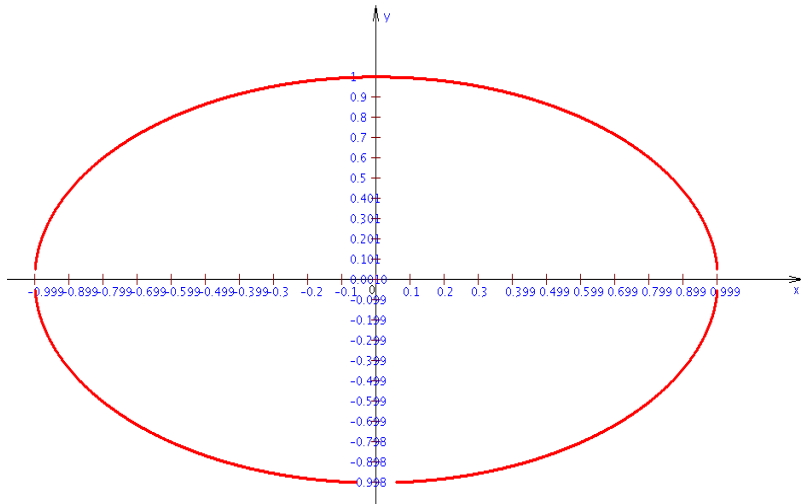


Рис. 2.3. График функции, заданной параметрически

Результат выполнения:

in:

```
p = tablePlot([0, 1, 4, 9, 16, 25], [0, 1, 2, 3, 4, 5], [-10, 10, -10, 10]);
```

out: рис. 2.4.

2.1.4. Построение разных графиков функций в одной системе координат

Для построения графиков функций, заданных разными способами, необходимо сначала построить график каждой функции, а затем выполнить команду:

```
\Showgraf([f1, f2, ..., fn]).
```

Можно указывать подписи по осям графика и его заголовок. Для этого достаточно выполнить команду:

```
\Showgraf([f1, f2, f3, f4], ["x", "y", "title"]),
```

указав вместо "x" — подпись по оси OX, вместо "y" — подпись по оси OY, вместо "title" — заголовок графика. По умолчанию используется ["x", "y", " "].

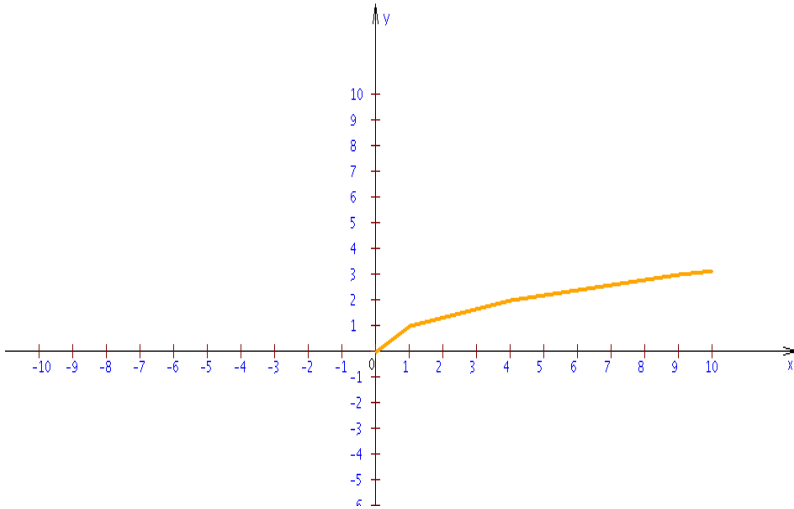


Рис. 2.4. График функции, заданной таблицей значений

Пример:

```
f1=\plot(\tg(x), [-20, 20, -20, 20]);
f2=\tablePlot([[0, 1, 4, 9, 16, 25], [0, 1, 2, 3, 4, 5]],
               [-10, 10, -10, 10]);
f3=\paramPlot([\sin(x), \cos(x)], [-10, 10]);
f4=\tablePlot([[0, 1, 4, 9, 16, 25],
               [0, -1, -2, -3, -4, -5]], [-10, 10, -10, 10]);
\Showgraf([f1, f2, f3, f4],
           ["x", "y", "The functions f1, f2, f3, f4, f5"]);
```

Результат запуска примера — рис. 2.5.

2.2. Построение 3D графиков функций

Система Mathrag позволяет строить 3D графики функций, которые заданы явно.

Для построения 3D графика функции $f = f(x, y)$ используется команда:

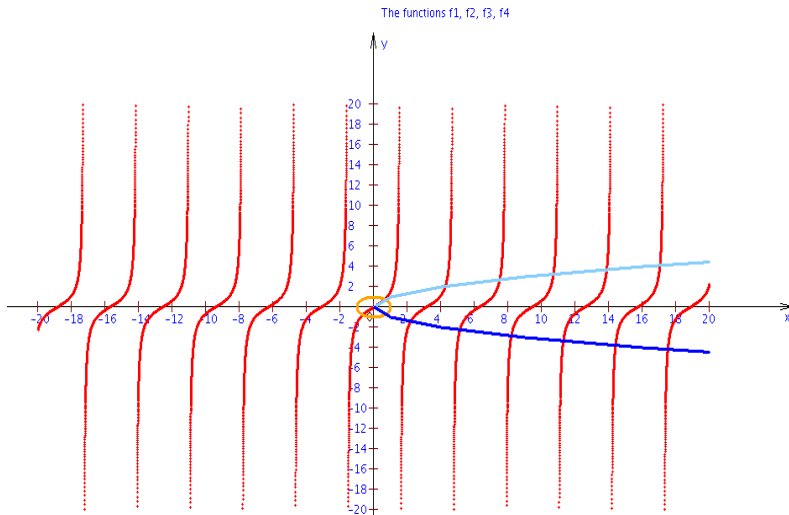


Рис. 2.5. Графики функций, заданных разными способами

`\plot3d(f, [x0, x1, y0, y1])`, где $[x0, x1]$ — интервал по оси OX , $[y0, y1]$ — интервал по оси OY .

Кроме того, полученные графики можно вращать и увеличивать либо уменьшать.

Перемещение мыши с нажатой левой кнопкой приводит к вращению системы координат графика. После остановки происходит перерисовка графика в новой повернутой системе координат.

Перемещение мыши с нажатой левой кнопкой и нажатой клавишей *Shift* приводит к изменению масштаба изображения. После остановки перемещения происходит перерисовка графика в новом масштабе.

Примеры:

1. `f=x^2/20+y^2/20;`
`\plot3d(f, [-20, 20, -20, 20]);`
2. `\plot3d([x/20+y^2/20, x^2/20+y/20], [-20, 20, -20, 20]);`

Результат выполнения:

in: $f = x^2/20 + y^2/20;$

```
plot3d(f, [-20, 20, -20, 20]);  
plot3d([x/20 + y^2/20, x^2/20 + y/20], [-20, 20, -20, 20]);  
out: рис. 2.6.
```

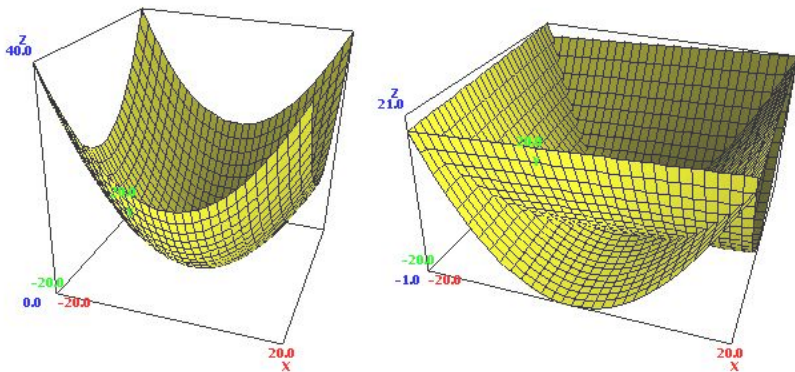


Рис. 2.6. Построение 3D графиков функций

2.3. Контрольные задания

В системе Mathpar постройте графики функций:

- $f(x) = x^2 + 2y;$
- $f(x) = \sqrt{\sin^2(5x - 1) + \exp x};$
- $f(x, y) = \sin(\cos(x + \tan(y))).$

Глава 3

Выбор окружения для математических объектов

3.1. Окружение

Прежде чем будет задан любой математический объект, число, функция или символ, должно быть ясно определено окружение — пространство, в котором будут определяться объекты. В этой главе описываются способы задания окружения в системе `Mathpar`. Перемещение из некоторого окружения в текущее, как правило, должно выполняться явно, с помощью функции `toNewRing`. В некоторых случаях такое преобразование к текущему окружению происходит автоматически.

Для выбора окружения задается *алгебраическое пространство переменных*. Оно определяется именами переменных и числовыми пространствами, в которых эти переменные принимают значения. Порядок переменных в списке переменных задает линейный порядок на этих переменных. Слева направо располагаются переменные, упорядоченные по старшинству от младших к старшим.

По умолчанию определено пространство шести целочисленных переменных $\mathbb{Z}[x, y, z]\mathbb{Z}[u, v, w]$, которые разделены на две группы по три переменные: самая младшая — x , самая старшая — w .

В любой момент пользователь может сменить окружение, задав новое алгебраическое пространство переменных с помощью команды установки «SPACE=». Например, для задач вычислительной ма-

тематики может быть достаточно пространства типа $\mathbb{R}64[x, y, z]$. Команда установки: «SPACE=R64[x, y, z];».

Все остальные имена, которые не указаны в качестве переменных, могут выбираться пользователем произвольно для обозначения математических объектов. Это свободные переменные. Необходимо придерживаться одного важного правила. Если имя свободной переменной начинается с заглавной буквы (верхний регистр), то такая переменная обозначает элемент алгебры, у которой *операция умножения некоммутативная*, а если со строчной буквы (нижний регистр) — *операция умножения коммутативная*.

3.2. Числовые множества

Определены следующие числовые множества:

\mathbb{Z} — множество целых чисел \mathbb{Z} ;

\mathbb{Z}_p — конечное поле из $p = \text{MOD}$ элементов $\mathbb{Z}/p\mathbb{Z}$, MOD — постоянная (см. п. 3.6);

\mathbb{Z}_{p32} — конечное поле из $p = \text{MOD}32$ элементов $\mathbb{Z}/p\mathbb{Z}$, MOD32 меньше 2^{31} (см. п. 3.6);

$\mathbb{Z}64$ — кольцо целых чисел z таких, что $-2^{63} \leq z < 2^{63}$;

\mathbb{Q} — множество рациональных чисел;

\mathbb{R} — множество чисел с плавающей точкой для хранения приближенных действительных чисел с произвольной мантиссой;

$\mathbb{R}64$ — множество чисел с плавающей точкой для хранения приближенных действительных чисел с двойной точностью (со стандартной 52-разрядной мантиссой и отдельным 11-разрядным полем для хранения порядка);

$\mathbb{R}128$ — стандартные 64-битные числа с плавающей точкой для хранения приближенных действительных чисел со стандартной 52-разрядной мантиссой и отдельным 64-разрядным полем для хранения порядка;

\mathbb{C} — комплексный класс, образованный из класса \mathbb{R} ;

$\mathbb{C}64$ — комплексный класс, образованный из класса $\mathbb{R}64$;

$\mathbb{C}128$ — комплексный класс, образованный из класса $\mathbb{R}128$;

$\mathbb{C}\mathbb{Z}$ — комплексный класс, образованный из класса \mathbb{Z} ;

$\mathbb{C}\mathbb{Z}_p$ — комплексный класс, образованный из класса \mathbb{Z}_p ;

$\mathbb{C}\mathbb{Z}_{p32}$ — комплексный класс, образованный из класса \mathbb{Z}_{p32} ;

$\mathbb{C}\mathbb{Z}64$ — комплексный класс, образованный из класса $\mathbb{Z}64$;

$\mathbb{C}Q$ — комплексный класс, образованный из класса Q .

Примеры простых полиномиальных колец:

$$\text{SPACE} = \mathbb{Z}[x, y, z];$$

$$\text{SPACE} = \text{R64}[u, v];$$

$$\text{SPACE} = \mathbb{C}[x].$$

3.3. Определение нескольких числовых множеств

Разрешается устанавливать алгебраические пространства из нескольких числовых множеств, например, пространство « $\mathbb{C}[z]\mathbb{R}[x, y]\mathbb{Z}[n, m]$ » позволяет работать с пятью именами переменных, определенных в множествах \mathbb{C} , \mathbb{R} и \mathbb{Z} соответственно. Первое множество считается основным. При необходимости к нему будут приводиться все остальные переменные. В данном случае это \mathbb{C} .

Его можно рассматривать как кольцо полиномов пяти переменных над \mathbb{C} , при этом оно обладает дополнительными свойствами. Если полином не содержит переменной z , то это полином с коэффициентами из \mathbb{R} . Если полином не содержит переменных z, x, y , то это полином с коэффициентами из \mathbb{Z} .

Примеры:

$$\text{SPACE} = \mathbb{Z}[x, y]\mathbb{Z}[u];$$

$$\text{SPACE} = \text{R64}[u, v]\mathbb{Z}[a, b];$$

$$\text{SPACE} = \mathbb{C}[x]\mathbb{R}[y, z];$$

Кольцо « $\mathbb{Z}[x, y, z]\mathbb{Z}[u, v, w]$ », в котором шесть переменных разделены на две группы, можно использовать для задач, в которых строятся полиномы, у которых коэффициенты являются полиномами или функциями других переменных. Например, характеристический полином для матрицы над кольцом $\mathbb{Z}[x, y, z]$ будет получен как полином с неизвестной u , коэффициенты которого лежат в кольце $\mathbb{Z}[x, y, z]$.

3.4. Групповые алгебры

Некоммутативные образующие группы должны начинаться с прописных латинских букв (верхний регистр). Символ G обозначает групповую алгебру. После него стоит список образующих, а перед ним — пространство, в котором действует группа.

Примеры свободных групповых алгебр:

SPACE=Z[x, y]G[U, V]; (образующие U, V);

SPACE=R64[u, v]G[A, B]; (образующие A, B);

SPACE=C[G[X, Y, Z, T]]; (образующие X, Y, Z, T).

Каждый элемент алгебры является суммой термов с коэффициентами, которые являются функциями. Например, «R64[x, y]G[X, Y, Z]» — это свободная групповая алгебра с тремя некоммутативными образующими X, Y, Z над функциями в $\mathbb{R}64[x, y]$. Тогда, например, $A = (t^2 + 1)X + \sin(t)Y + 3X^2y^3 + (t^2 + 1)XY^3X^2Y^{-2}x^2$ — элемент такой алгебры.

3.5. Идемпотентные алгебры. Тропическая математика

Кроме классических числовых алгебр с операциями «+, -, *» и операцией «/» для полей, пользователю будут доступны и идемпотентные алгебры. Для числового множества $\mathbb{R}64$ можно будет использовать алгебры *R64MaxPlus*, *R64MinPlus*, *R64MaxMin*, *R64MinMax*, *R64MaxMult*, *R64MinMult*. Для числового множества \mathbb{R} можно будет использовать алгебры *RMaxPlus*, *RMinPlus*, *RMaxMin*, *R64MinMax*, *RMaxMult*, *RMinMult*. Для числового множества \mathbb{Z} можно будет использовать алгебры *ZMaxPlus*, *ZMinPlus*, *ZMaxMin*, *ZMinMax*, *ZMaxMult*, *ZMinMult*.

В текущей версии можно использовать алгебру *ZMaxPlus*.

Пример:

```
SPACE=ZMaxPlus[x, y];
```

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

Результат выполнения:

```
c = 9;
```

```
d = 11.
```

3.6. Константы

Можно установить или заменить следующие постоянные:

ACCURACY — число точных десятичных позиций после запятой для чисел типа R в операциях умножения и деления.

FLOATPOS — число десятичных знаков после запятой, которые выводятся на печать.

ZERO_R — машинный ноль для чисел типа R.

ZERO_R64 — машинный ноль для чисел типа R64.

MOD32 — модуль для простого поля, не превосходящий 2^{31} .

MOD — модуль типа Z для простого поля.

Для установки машинного ноля $1/10^9$ (т. е. $1E-9$) нужно ввести команду «ZERO_R=9».

Пример:

```
SPACE=Zp32[x, y];
```

```
MOD32=7;
```

```
f1=37x+42y+55;
```

```
f2=2f1;
```

```
\print(f1, f2 );
```

Результат выполнения:

$$f1 = 2x - 1;$$

$$f2 = 4x + 5.$$

3.7. Контрольные задания

В системе Mathpar вычислите:

- значение функции $f(x) = \sqrt{\sin^2(5x - 1) + e^x}$ в \mathbb{R} при $x = 7$;
- значение функции $f(x) = x^3 + 10x$ в $\mathbb{Z}/(11)\mathbb{Z}$ при $x = 7$.

Глава 4

Функции одной и нескольких переменных

4.1. Математические функции

Приняты следующие обозначения для элементарных функций и констант.

4.1.1. Константы

$\sqrt{-1}$ — мнимая единица;

e — основание натурального логарифма;

π — число π , т. е. отношение длины окружности к диаметру;

∞ — знак бесконечности.

4.1.2. Функции одного аргумента

\ln — натуральный логарифм;

\lg — десятичный логарифм;

\sin — синус;

\cos — косинус;

\tan — тангенс;

\cot — котангенс;

\arcsin — арксинус;

\arccos — арккосинус;

`\arctg` — арктангенс;
`\arcctg` — арккотангенс;
`\sh` — синус гиперболический;
`\ch` — косинус гиперболический;
`\th` — тангенс гиперболический;
`\cth` — котангенс гиперболический;
`\arcsh` — арксинус гиперболический;
`\arcch` — арккосинус гиперболический;
`\arcth` — арктангенс гиперболический;
`\arccth` — арккотангенс гиперболический;
`\exp` — экспонента;
`\sqrt` — корень квадратный;
`\abs` — абсолютное значение для действительных чисел, модуль для комплексного числа;
`\sign` — знак числа. Возвращает 1, 0, -1, когда число положительное, ноль или отрицательное соответственно;
`\unitStep(x, a)` — это функция, которая при $x \geq 0$ принимает значение 1, а при $x < 0$ принимает значение 0;
`\fact` — факториал. Определен для целых положительных чисел. Можно писать в привычном виде, например, «5!».

4.1.3. Функции двух аргументов

`^` — степень;
`\log` — логарифм от функции по указанному основанию;
`\root(n, x)` — корень степени n из x ;
`\Gamma` — функция Гамма;
`\Gamma2` — функция Гамма 2;
`\binomial` — число сочетаний.

Примеры:

```
SPACE=R64[x, y];
f1=\sin(x);
f2=\sin(\cos(x+\tg(y)));
f3=\sin(x^2)+y;
\print(f1, f2, f3);
```

Результат выполнения:

```
f1 = sin(x);
f2 = sin(cos(x + tg(y)));
f3 = sin(x^2) + y.
```

4.2. Вычисление значений функции в точке

Для вычисления значения функции в точке необходимо выполнить команду:

`\value(f, [var1, var2, ..., varn])`, где f — функция, а $var1, var2, \dots, varn$ — значения соответствующих переменных.

Для тригонометрических функций мерой угла считается радиан или градус. Указание меры угла определяется константой `RADIAN`. Если не указывать угловую меру, то угловой мерой выбирается радиан. Чтобы поменять угловую меру с радиан на градусы, нужно выполнить команду «`RADIAN=0`;». Если же нужно поменять угловую меру с градусов на радианы, то нужно выполнить команду «`RADIAN=1`;».

Если аргументами являются целые числа $15k$ и $18k$ градусов или $\pi k/12$ и $\pi k/10$ радиан ($k \in \mathbb{Z}$), то значениями тригонометрических функций являются алгебраические числа.

Примеры:

```
1. SPACE=R[x, y];
f=\sin(x^2+\tg(y^3+x));
g=\value(f, [1, 2]);
\print(g);
```

Результат выполнения:

```
in: SPACE = R[x, y];
    f = sin(x^2 + tg(y^3 + x));
    g = value(f, [1, 2]);    print(g);
out: g = 0.52.
```

```
2. SPACE=Z[x];
RADIAN=0;
f=\sin(x);
g=\value(f, 15);
\print(g);
```

Результат выполнения:

```
in: SPACE = Z[x];
    RADIAN = 0;
    f = sin(x);
```

```
g = value(f, 15);
print(g);
out: g = ( $\sqrt{6} - (\sqrt{2})$ )/(4).
```

```
3. SPACE= Z[x];
RADIAN=0;
f=\sin(x);
g=\value(f, 225);
\print(g);
```

Результат выполнения:

$$g = (-1 \cdot \sqrt{2})/(2).$$

```
4. SPACE= Z[x];
RADIAN=0;
f=\cos(x);
g=\value(f, 54);
\print(g);
```

Результат выполнения:

$$g = \sqrt{(5 - \sqrt{5})}/(8)$$

```
5. SPACE= Z[x];
RADIAN=0;
f=\tg(x);
g=\value(f, 126);
\print(g);
```

Результат выполнения:

$$g = (-1 \cdot \sqrt{(1 + 2 \cdot \sqrt{5})/(5)}).$$

```
6. SPACE= Z[x];
RADIAN=0;
f=\sin(x);
g=\value(f, 216);
\print(g);
```

Результат выполнения:

$$g = (-1 \cdot \sqrt{(5 - \sqrt{5})}/(8)).$$


```

7. SPACE= Z[x];
RADIAN=0;
f=\cos(x);
g=\value(f, 108);
\print(g);

```

Результат выполнения:

$$g = (1 - \sqrt{5})/4.$$

4.3. Подстановка выражений в функции

Для вычисления композиции функций нужно подставлять в функцию вместо ее аргументов другие функции. Для этого необходимо выполнить команду:

`\value(f, [func1, func2, ..., funcn])`, где f — данная функция, $func1, func2, \dots, funcn$ — функции, которые подставляются вместо соответствующих переменных.

Пример:

```

SPACE=Z[x, y];
f=x+y; g=f^2;
r=\value(g, [x^2, y^2]);
\print(r);

```

Результат выполнения:

$$\text{in: } g = y^2 + 2yx + x^2;$$

$$f = y + x;$$

$$\text{out: } r = y^4 + 2y^2x^2 + x^4.$$

4.4. Вычисление предела функции в точке

Для вычисления предела функции в точке необходимо выполнить команду: `\lim(f, var)`, где f — это функция, а var — точка, возможно бесконечная, в которой требуется найти предел, конечный или бесконечный.

Примеры:

```

1. SPACE=R64[x];
f=\sin(x)/x;
g=\lim(f, 0);
\print(g);

```

Результат выполнения:

```
in: SPACE = R64[x];
```

```
    f = sin(x)/x;
```

```
    g = lim(f, 0);
```

```
    print(g);
```

```
out: g = 1.00.
```

```
2. SPACE=R64[x];
```

```
f=(x^2-2x+2)/(x^2+x-2);
```

```
g=\lim(f, 1);
```

```
\print(g);
```

Результат выполнения:

```
in: SPACE = R64[x];
```

```
    f = (x2 - 2x + 2)/(x2 + x - 2);
```

```
    g = lim(f, 1);
```

```
    print(g);
```

```
out: g = ∞.
```

```
3. SPACE=R64[x];
```

```
f=\sin(x+3)/(x^2+6x+9);
```

```
g=\lim(f, -3);
```

```
\print(g);
```

Результат выполнения:

```
in: SPACE = R64[x];
```

```
    f = sin(x + 3)/(x2 + 6x + 9);
```

```
    g = lim(f, -3);
```

```
    print(g);
```

```
out: g = ∞.
```

```
4. SPACE=R64[x];
```

```
f=(1+1/x)^x;
```

```
g=\lim(f, \infty);
```

```
\print(g);
```

Результат выполнения:

```
in: SPACE = R64[x];
```

```
    f = (1 + 1/x)x;
```

```
    g = lim(f, ∞);
```

```
    print(g);
```

```
out: g = 2.72.
```

4.5. Дифференцирование функций

Для вычисления производной функции f по переменной y из кольца $\mathbb{Z}[x, y, z]$ необходимо выполнить команду $\backslash\mathbf{D}(f, y)$. Вычисление третьей производной по y можно выполнить $\backslash\mathbf{D}(f, y^3)$. Если необходимо найти производную функции f один раз по первой переменной из текущего кольца (в данном случае x) можно записать $\backslash\mathbf{D}(f)$.

Для нахождения смешанной производной первого порядка от функции f существует команда $\backslash\mathbf{D}(f, [x, y])$, для нахождения производной высших порядков нужно использовать команду $\backslash\mathbf{D}(f, [x^k, z^m, y^n])$, где k, m, n указывают, какого порядка по соответствующей переменной вычисляется производная.

Примеры:

```
1. SPACE=Z[x, y];
f=\sin(x^2+ \tg(y^3+x));
h= \D(f, y);
\print(h);
```

Результат выполнения:

```
in: SPACE = Z[x, y];
    f = sin(x^2 + tg(y^3 + x));
    h = D(f, y);
    print(h);
```

out: $h = 3y^2 \cos(x^2 + \operatorname{tg}(y^3 + x)) / (\cos(y^3 + x))^2$.

```
2. SPACE=Z[x, y];
f=\sin(x^2+ \tg(y^3+x));
h= \D(f);
\print(h);
```

Результат выполнения:

```
in: SPACE = Z[x, y];
    f = sin(x^2 + tg(y^3 + x));
    h = D(f);
    print(h);
```

out: $h = (2x \cos(x^2 + \operatorname{tg}(y^3 + x))(\cos(y^3 + x))^2 + \cos(x^2 + \operatorname{tg}(y^3 + x))) / (\cos(y^3 + x))^2$.

```
3. SPACE=Z[x, y, z];
f=x^8y^4z^9;
```

```
g=\D(f, [x^2, y^2, z^2]);  
\print(g);
```

Результат выполнения:

```
in: SPACE = Z[x, y, z];  
    f = x^8*y^4*z^9;  
    g = D(f, [x^2, y^2, z^2]);  
    print(g);  
out: g = 48384z^7y^2x^6.
```

4.6. Интегрирование композиций элементарных функций

Символьное интегрирование композиций элементарных функций выполняется командой:

```
\int(f(x))dx.
```

Примеры:

```
1. SPACE=Z[x, y, z]Z[u, v, w];  
n=\int((x+2)/(x^2+4x+2))d x;  
dn=\D(n);  
\print(n, dn);
```

Результат выполнения:

```
in: SPACE = Z[x, y, z]Z[u, v, w];  
    n = \int((x + 2)/(x^2 + 4x + 2))dx;  
    dn = D(n);  
    print(n, dn);  
out: n = 1/2 * ln(x^2 + 4x + 2);  
    dn = (x + 2)/(x^2 + 4x + 2).
```

```
2. SPACE=Z[x, y, z]Z[u, v, w];  
l=\int(x^6+x-2)d x;  
dl=\D(l);  
\print(l, dl);
```

Результат выполнения:

```
in: SPACE = Z[x, y, z]Z[u, v, w];  
    l = \int(x^6 + x - 2)dx;  
    dl = D(l);
```

```
print(l, dl);  
out: l = 1/7x7 + 1/2x2 - 2x;  
dl = x6 + x - 2.
```

4.7. Контрольные задания

В системе Mathpar:

- подставьте в функцию $f(x) = \sqrt{\sin^2(5x - 1) + e^x}$ вместо x выражение $x + y$, а вместо $y - 5$;
- найдите $\lim(x^3 + 10x)/x^2$ при $x \rightarrow 0$;
- найдите производную функции $f(x) = \sqrt{\sin^2(5x - 1) + e^x}$;
- найдите $\int_0^2 (x^3 + 10x)dx$.

Глава 5

Ряды

Ряд задается в виде $f = \sum_{i=k}^{\infty} F(i, x, y, \dots, z)$, где i — индекс суммирования, k — начальное значение i , $F(i, x, y, \dots, z)$ — функция многих переменных, т.е. F может зависеть и от i .

Над рядами определены арифметические операции: сложение, вычитание и умножение.

Пусть f и g — ряды.

Для сложения двух рядов необходимо выполнить команду: `\seriesAdd(f, g)`.

Для разности двух рядов необходимо выполнить команду: `\seriesSubtract(f, g)`.

Для умножения двух рядов необходимо выполнить команду: `\seriesMultiply(f, g)`.

Для разложения функции в ряд Тейлора с определенным количеством членов ряда необходимо выполнить команду `\teilor(f, point, num)`, где f — функция, $point$ — точка, num — общее количество членов ряда.

Примеры:

```
1. SPACE=R[x, y];
f=\sum_{a=2}^{\infty} (2xyab);
g=\sum_{b=4}^{\infty} (ya\sin(abx));
h=\seriesAdd(f, g);
\print(f, g, h);
```

Результат выполнения:

$$f = \sum_{a=2}^{\infty} 2.00yxab;$$

$$g = \sum_{b=4}^{\infty} ya \sin(abx);$$

$$h = \left(\sum_{a_0=4}^{\infty} (2.00yxa_0b + ya \sin(aa_0x)) + \sum_{a=2}^3 (2.00yxab) \right).$$

```
2. SPACE=R[x, y];
f=\sum_{a=1}^{\infty} (xya\cos(b));
g=\sum_{b=2}^{\infty} (56x^4a\cos(abx));
h=\seriesSubtract(f, g);
\print(f, g, h);
```

Результат выполнения:

$$f = \sum_{a=1}^{\infty} yxa \cos(b);$$

$$g = \sum_{b=2}^{\infty} 56.00x^4a \cos(abx);$$

$$h = \left(\sum_{a_0=2}^{\infty} (yxa_0 \cos(b) - 56.00x^4a \cos(aa_0x)) + yx \cos(b) \right).$$

```
3. SPACE=R[x, y];
f=\sum_{a=5}^{\infty} (2x^2ydab);
g=\sum_{b=1}^{\infty} (5y^8x^2bl\cos(a_1x));
h=\seriesSubtract(f, g);
\print(f, g, h);
```

Результат выполнения:

$$f = \sum_{a=5}^{\infty} 2.00yx^2dab;$$

$$g = \sum_{b=1}^{\infty} 5.00y^8x^2bl \cos(a_1x);$$

$$h = \left(\sum_{a=5}^{\infty} (2.00y^2 dab - 5.00y^8 x^2 al \cos(a_1x)) - \sum_{b=1}^{4.0} (5.00y^8 x^2 bl \cos(a_1x)) \right).$$

```
4. SPACE=R[x, y];
f=\sum_{a=6}^{\infty} (xaa_0);
g=\sum_{a=9}^{\infty} (56x^4a\cos(a_1x));
h=\seriesMultiply(f, g);
\print(f, g, h);
```

Результат выполнения:

$$f = \sum_{a=6}^{\infty} xaa_0;$$

$$g = \sum_{a=9}^{\infty} 56.00x^4a \cos(a_1x);$$

$$h = \sum_{a_2=6}^{\infty} \sum_{a=9}^{\infty} xa_2a_0 \cdot 56.00x^4a \cos(a_1x).$$

```
5. SPACE=R[x, y];
f=\sum_{a=6}^{\infty} (\sin(xa)\cos(gy)a_0);
g=\sum_{a=9}^{\infty} (6x^2a\sin(axy^2));
h=\seriesMultiply(f, g);
\print(f, g, h);
```

Результат выполнения:

$$f = \sum_{a=6}^{\infty} \sin(xa) \cos\left(\sum_{a=9}^{\infty} 56.00x^4a \cos(a_1x)y\right)a_0;$$

$$g = \sum_{a=9}^{\infty} 6.00x^2a \sin(axy^2);$$

$$h = \sum_{a_2=6}^{\infty} \sum_{a=9}^{\infty} \sin(xa_2) \cos\left(\sum_{a_2=9}^{\infty} 56.00x^4a_2 \cos(a_1x)y\right)a_0 6.00x^2a \sin(axy^2).$$

```

6. SPACE=R[x];
FLOATPOS=15;
a=\teilor(\sin(x), 0, 7);
c=\value(a);
\print(a, c);

```

Результат выполнения:

```

a = ((-x^7)/(7!) + x^5/(5!) + (-x^3)/(3!) + x/(1!));
c   =      (-0.000198412698412x^7 + 0.008333333333333x^5 -
0.166666666666666x^3 + x).

```

5.1. Контрольные задания

В системе Mathpar :

- разложите функцию $f(x) = \sin^2(5x - 1)$ в ряд Тейлора;
- создайте два случайных ряда с полиномиальными членами, используя генерацию полиномов. Найдите сумму, разность и произведение полученных рядов.

Глава 6

Решение дифференциальных уравнений и их систем

6.1. Решение дифференциальных уравнений

Для решения дифференциального уравнения с постоянными коэффициентами необходимо выполнить следующие шаги:

1. Задать пространство переменных (*SPACE*).
2. Задать уравнение (`\systLDE`).
3. Задать начальные условия (`\initCond`).
4. Получить решение (`\solveLDE`).

Примеры:

```
1. SPACE=R64[t];
ZERO_R64=8;
g=\systLDE(\d(y, t, 3)+3\d(y, t, 2)+3\d(y, t)+y=1);
f=\initCond(\d(y, t, 0, 0)=0, \d(y, t, 0, 1)=0,
             \d(y, t, 0, 2)=0);
h=\solveLDE(g, f);
\print(h);
```

Результат выполнения:

```
in: SPACE = R64[t];
    ZERO_R64 = 8;
    g = y_t''' + 3y_t'' + 3y_t' + y = 1;
    f = { y_t=0 = 0,
          y_t=0 = 0,
          y_t=0 = 0,
    };
    h = solveLDE(g, f);
    print(h);
```

out: $h = (1.00 + (-t^2)e^{-t}/2.00) - (te^{-t} + e^{-t})$.

2. SPACE=R64[t];

```
ZERO_R64=8;
g=\systLDE(\d(y, t, 2)-2\d(y, t)+y=\exp(t));
f=\initCond(\d(y, t, 0, 0)=1, \d(y, t, 0, 1)=2);
h=\solveLDE(g, f);
\print(h);
```

Результат выполнения:

```
in: SPACE = R64[t];
    ZERO_R64 = 8;
    g = y_t'' - 2y_t' + y = e^t;
    f = { y_t=0 = 1,
          y_t=0 = 2,
    };
    h = solveLDE(g, f);
    print(h);
```

out: $h = e^t t^2 / 2.00 + te^t + e^t$.

3. SPACE=R64[t];

```
ZERO_R64=8;
g=\systLDE(\d(y, t, 2)+\d(y, t)-12y=3);
f=\initCond(\d(y, t, 0, 0)=1, \d(y, t, 0, 1)=0);
h=\solveLDE(g, f);
\print(h);
```

Результат выполнения:

```
in: SPACE = R64[t];
    ZERO_R64 = 8;
    g = y_t'' + y_t' - 12y = 3;
    f = { y_t=0 = 1,
          y_t=0 = 0,
    };
```

```

    h = solveLDE(g, f);
    print(h);
out: h = 1.11e-1.62t + 2.89e0.62t - 3.00.

```

```

4. SPACE=R64[t];
ZERO_R64=8;
g=\systLDE(\d(y, t)-2y=0);
f=\initCond(\d(y, t, 0, 0)=1);
h=\solveLDE(g, f);
\print(h);

```

Результат выполнения:

```

in: SPACE = R64[t];
    ZERO_R64 = 8;
    g = y'_t - 2y = 0;
    f = y_{t=0} = 1;
    h = solveLDE(g, f);
    print(h).
out: h = et;

```

```

5. SPACE=R64[t];
ZERO_R64=8;
g=\systLDE(\d(y, t, 2)-4y=4t);
f=\initCond(\d(y, t, 0, 0)=a, \d(y, t, 0, 1)=b);
h=\solveLDE(g, f);
\print(h);

```

Результат выполнения:

```

in: SPACE = R64[t];
    ZERO_R64 = 8;
    g = y''_t - 4y = 4t;
    f = {
        y_{t=0} = a,
        y'_{t=0} = b,
    };
    h = solveLDE(g, f);
    print(h);
out: h = (-8.00 + (-2.00b) + 2.00a)/4.00e-t + (8.00 + 2.00b +
2.00a)/4.00et - 4.00t.

```

6.2. Решение систем дифференциальных уравнений

Для решения системы дифференциальных уравнений с постоянными коэффициентами необходимо выполнить следующие шаги:

1. Задать пространство переменных (*SPACE*).
2. Задать систему уравнений (`\systLDE`).
3. Задать начальные условия (`\initCond`).
4. Получить решение (`\solveLDE`).

Примеры:

```
1. SPACE=R64[t];
g=\systLDE(\d(x, t)-y+z=0, -x-y+\d(y, t)=0, -x-z+\d(z, t)=0);
f= \initCond(\d(x, t, 0, 0)=1, \d(y, t, 0, 0)=2,
             \d(z, t, 0, 0)=3);
h= \solveLDE(g, f);
\print(h);
```

Результат выполнения:

in: *SPACE* = *R64*[*t*];

$$g = \begin{cases} x'_t - y + z = 0, \\ -x - y + y'_t = 0, \\ -x - z + z'_t = 0, \end{cases}$$

$$f = \begin{cases} x_{t=0} = 1, \\ y_{t=0} = 2, \\ z_{t=0} = 3, \end{cases}$$

$$h = \text{solveLDE}(g, f);$$

print(*h*);

out: $h = [(-2.00) + 5.00e^t + (-1.00e^t)t, 2.00 + (-1.00e^t), (-2.00) + 4.00e^t + (-1.00e^t)t].$

```
2. SPACE=R64[t];
```

```
g=\systLDE(\d(x, t, 2)+\d(x, t)-\d(y, t)=1,
           \d(x, t)+x-\d(y, t, 2)=1+4\exp(t));
```

```
f=\initCond(\d(x, t, 0, 0)=1, \d(x, t, 0, 1)=2,
            \d(y, t, 0, 0)=0, \d(y, t, 0, 1)=1);
```

```
h= \solveLDE(g, f);
```

```
\print(h);
```

Результат выполнения:

in: $SPACE = R64[t]$;

$$g = \begin{cases} x_t'' + x_t' - y_t'' & = 1, \\ x_t' + x - y_t'' & = 1 + 4e^t, \end{cases}$$

$$f = \begin{cases} x_{t=0} & = 1, \\ x_{t=0}' & = 2, \\ y_{t=0} & = 0, \\ y_{t=0}' & = 1, \end{cases}$$

$h = \text{solveLDE}(g, f)$;

$\text{print}(h)$;

out: $h = [1.00 + 2.00e^t + (-1.00e^t)t + (-2.00e^{-t}) + (-1.00e^{-t})t, (-2.00) + (-1.00t) + 3.00e^t + (-2.00e^t)t + (-1.00e^{-t})]$.

3. $SPACE=R64[t]$;

$g = \backslash\text{systLDE}(3 \backslash d(x, t) + 2x + \backslash d(y, t) = 1, \backslash d(x, t) + 4 \backslash d(y, t) + 3y = 0)$;

$f = \backslash\text{initCond}(\backslash d(x, t, 0, 0) = 0, \backslash d(x, t, 0, 1) = 0,$

$\backslash d(y, t, 0, 0) = 0, \backslash d(y, t, 0, 1) = 0)$;

$h = \backslash\text{solveLDE}(g, f)$;

$\backslash\text{print}(h)$;

Результат выполнения:

in: $SPACE = R64[t]$;

$$g = \begin{cases} 3x_t' + 2x + y_t' & = 1, \\ x_t' + 4y_t' + 3y & = 0, \end{cases}$$

$$f = \begin{cases} x_{t=0}' & = 0, \\ x_{t=0} & = 0, \\ y_{t=0}' & = 0, \\ y_{t=0} & = 0, \end{cases}$$

$h = \text{solveLDE}(g, f)$;

$\text{print}(h)$;

out: $h = [0.50 + (-0.30e^{-0.55t}) + (-0.20e^{-t}), (-0.20e^{-0.55t}) + 0.20e^{-t}]$.

4. $SPACE=R64[t]$;

$g = \backslash\text{systLDE}(\backslash d(x, t) + 3x - 4y = 9 \backslash \exp(2t),$

$2x + \backslash d(y, t) - 3y = 3 \backslash \exp(2t))$;

$f = \backslash\text{initCond}(\backslash d(x, t, 0, 0) = 2, \backslash d(y, t, 0, 0) = 0)$;

$h = \backslash\text{solveLDE}(g, f)$;

$\backslash\text{print}(h)$;

Результат выполнения:

in: $SPACE = R64[t]$;

$$g = \begin{cases} x'_t + 3x - 4y = 9e^{2t}, \\ 2x + y'_t - 3y = 3e^{2t}, \end{cases}$$

$$f = \begin{cases} x_{t=0} = 2, \\ y_{t=0} = 0, \end{cases}$$

$h = solveLDE(g, f)$;

$print(h)$;

out: $h = [e^t + e^{2.00t}, e^t + (-1.00e^{2.00t})]$.

5. $SPACE=R64[t]$;

$g = \backslash systLDE(\backslash d(x, t) - x - 2y = 0, \backslash d(y, t) - 2x - y = 1)$;

$f = \backslash \initCond(\backslash d(x, t, 0, 0) = 0, \backslash d(y, t, 0, 0) = 5)$;

$h = \backslash solveLDE(g, f)$;

$\backslash print(h)$;

Результат выполнения:

in: $SPACE = R64[t]$;

$$g = \begin{cases} x'_t - x - 2y = 0, \\ y'_t - 2x - y = 1, \end{cases}$$

$$f = \begin{cases} x_{t=0} = 0, \\ y_{t=0} = 5, \end{cases}$$

$h = solveLDE(g, f)$;

$print(h)$;

out: $h = [(-0.67) + 0.17e^{3.00t} + 0.50e^{-t}, 0.33 + (-0.50e^{-t}) + 0.17e^{3.00t}]$.

6. $SPACE=R64[t]$;

$g = \backslash systLDE(\backslash d(x, t, 2) + \backslash d(y, t) + y = \backslash exp(t) - t,$

$\backslash d(x, t) - x + 2\backslash d(y, t, 2) - y = -\backslash exp(-t))$;

$f = \backslash \initCond(\backslash d(x, t, 0, 0) = 1, \backslash d(x, t, 0, 1) = 2,$

$\backslash d(y, t, 0, 0) = 0, \backslash d(y, t, 0, 1) = 0)$;

$h = \backslash solveLDE(g, f)$;

$\backslash print(h)$;

Результат выполнения:

in: $SPACE = R64[t]$;

$$g = \begin{cases} x''_t + y'_t + y = e^t - t, \\ x'_t - x + 2y''_t - y = -e^{-t}, \end{cases}$$

$$f = \begin{cases} x_{t=0} = 1, \\ x_{t=0} = 2, \\ y_{t=0} = 0, \\ y_{t=0} = 0, \end{cases}$$

$h = \text{solveLDE}(g, f);$
 $\text{print}(h);$

out: $h = [1.00t + e^t, 1.00 + (-1.00t) + (-1.00e^{-t})].$

7. $\text{SPACE}=\text{R64}[t];$

$g = \text{systLDE}(\backslash d(x, t, 2) + \backslash d(y, t) = \backslash sh(t) - \backslash sin(t) - t,$
 $\backslash d(y, t, 2) + \backslash d(x, t) = \backslash ch(t) - \backslash cos(t));$
 $f = \text{initCond}(\backslash d(x, t, 0, 0) = 0, \backslash d(x, t, 0, 1) = 2,$
 $\backslash d(y, t, 0, 0) = 1, \backslash d(y, t, 0, 1) = 0);$
 $h = \text{solveLDE}(g, f);$
 $\backslash \text{print}(h);$

Результат выполнения:

in: $\text{SPACE} = \text{R64}[t];$

$$g = \begin{cases} x_t'' + y_t' = sh(t) - sin(t) - t, \\ y_t'' + x_t' = ch(t) - cos(t), \end{cases}$$

$$f = \begin{cases} x_{t=0} = 0, \\ x_{t=0} = 2, \\ y_{t=0} = 1, \\ y_{t=0} = 0, \end{cases}$$

$h = \text{solveLDE}(g, f);$
 $\text{print}(h);$

out: $h = [1.00t + 0.50e^t + (-0.50e^{-t}), (-1.00t^2)/2.00 + 0.50e^{1.00it} + (0.50e^{-1.00it})].$

8. $\text{SPACE}=\text{R64}[t];$

$g = \text{systLDE}(\backslash d(x, t, 2) - \backslash d(x, t) + \backslash d(y, t) = \backslash exp(-t) + \backslash cos(t),$
 $\backslash d(x, t) - \backslash d(y, t, 2) - \backslash d(y, t) = 2 \backslash exp(t) + \backslash sin(t));$
 $f = \text{initCond}(\backslash d(x, t, 0, 0) = 2, \backslash d(x, t, 0, 1) = 1,$
 $\backslash d(y, t, 0, 0) = 0, \backslash d(y, t, 0, 1) = 1);$
 $h = \text{solveLDE}(g, f);$
 $\backslash \text{print}(h);$

Результат выполнения:

in: $\text{SPACE} = \text{R64}[t];$

$$g = \begin{cases} x_t'' - x_t' + y_t' = e^{-t} + cos(t), \\ x_t' - y_t'' - y_t' = 2e^t + sin(t), \end{cases}$$

$$f = \begin{cases} x_{t=0} & = 2, \\ x_{t=0} & = 1, \\ y_{t=0} & = 0, \\ y_{t=0} & = 1, \end{cases}$$

$h = \text{solveLDE}(g, f);$
 $\text{print}(h);$
 out: $h = [0.50e^{1.00it} + 0.50e^{-1.00it} + (-1.00e^{-t}), 0.50i(e^{1.00it}) + (-0.50i(e^{-1.00it})) + (2.00e^t)].$

6.3. Контрольные задания

В системе Mathprag решите:

- дифференциальное уравнение $y'' + 3y' - y = e^t$ с начальными условиями $y(0) = 1, y'(0) = 1$;
- систему дифференциальных уравнений

$$\begin{cases} x'' + y' = \cos t + t^2, \\ y'' + x' = \sin t, \end{cases}$$

с начальными условиями $x(0) = y(0) = 0, x(1) = 2, y(1) = 1$.

Глава 7

Полиномиальные вычисления

7.1. Вычисление значения полинома в точке

Для вычисления значения функции в точке необходимо выполнить :

`\value(f, [var1, var2, ..., varn])`, где f — это полином, в который на позиции переменных кольца подставляем соответствующие значения $var1, var2, \dots, varn$.

Пример:

```
SPACE=R[x, y];  
f=x^2+5x(y^3+x);  
g=\value(f, [1, 2]);  
\print(g);
```

Результат выполнения:

```
in: SPACE = R[x, y];  
    f = x^2 + 5x(y^3 + x);  
    g = value(f, [1, 2]);  
    print(g);  
out: g = 46.00.
```

7.2. Суммирование полинома по переменным. Геометрические прогрессии

Для суммирования полинома по переменным необходимо выполнить команду:

`\SumOfPol(f, [x, y], [x1, x2, y1, y2])`, где f — полином, x, y — переменные по которым ведется суммирование, $x1, x2$ — интервал суммирования по x , $y1, y2$ — интервал суммирования по y .

Если интервалы суммирования для всех переменных совпадают, то можно записать:

`\SumOfPol(f, [x, y], [x1, x2])`, где $x1, x2$ — интервал суммирования по x и y .

Пример:

```
SPACE=R[x, y, z];
f=x^2z+xy+y^3xz;
res=\SumOfPol(f, [x, y], [2, 4, -2, 3]);
\print(res);
```

Результат выполнения:

```
in: SPACE = R[x, y, z];
    f = x^2z + xy + y^3xz;
    res = SumOfPol(f, [x, y], [2, 4, -2, 3]);
    print(res);
out: res = 417.00z + 27.00.
```

Для преобразования полинома с помощью формулы суммы геометрической прогрессии необходимо выполнить команду:

`\SearchOfProgression(f)`. Данная команда ищет геометрическую прогрессию с наибольшим числом членов среди мономов полинома, затем делает это еще раз для оставшихся членов и так далее. Найденные прогрессии записываются в виде $S_n = b_1(q^n - 1)/(q - 1)$, где S_n — сумма первых n членов, b_1 — первый член геометрической прогрессии, q — знаменатель прогрессии.

Пример:

```
SPACE=R[x, y, z];
f=x^3+x^4+x^5+x^6+x^7+x^8+x^9+x^10+x^11+x^12+x^13;
g=x+x^5+x^9+x^13+xyz+7x^2y^2z^2+7x^3y^3z^3+100xy+x^2+
                                     x^3+x^4;
f1=\SearchOfProgression(f);
```

```
g1=\SearchOfProgression(g);
\print(f1, g1);
```

Результат выполнения:

```
in: SPACE = R[x, y, z];
    f = x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 + x12 + x13;
    g = x + x5 + x9 + x13 + xyz + 7x2y2z2 + 7x3y3z3 + 100xy + x + x2 + x3 + x4;
    f1 = SearchOfProgression(f);
    g1 = SearchOfProgression(g);
    print(f1, g1);
out: f1 = (x14 - x3)/(x - 1);
     g1 = ((100.00yx + x13 + x9 + x) + (z4y4x4 - zyx))/(zyx - 1) + (x6 -
x)/(x - 1) + 6.00z3y3x3 + 6.00z2y2x2).
```

7.3. Вычисление базисов Гребнера

Для вычисления базиса Гребнера полиномиального идеала над рациональными числами можно воспользоваться командой `\gbasisB()` или командой `\gbasis()`. Команда `\gbasisB()` вычисляет базиса Гребнера, используя алгоритм Бухбергера, а команда `\gbasis()` использует его матричный вариант, предложенный Фужером.

Примеры:

```
1. SPACE=Q[x, y, z];
b=\gbasisB(x4y3+2xy2+3x+1, x3y2+x2, x4y+z2+xy4+3);
\print(b);
```

Результат выполнения:

```
in: SPACE = Q[x, y, z];
    b = gbasisB(x4y3 + 2xy2 + 3x + 1, x3y2 + x2, x4y + z2 + xy4 + 3);
    print(b);
out: b = [z2 - x4 + 3x2 + (-10)x + 9, y + (-9)x4 + (-3)x3 - x2 + (-81)x +
27, x5 + 9x2 + (-6)x + 1].
```

```
2. SPACE=Z[x, y, z];
b=\gbasis(x4y3+2xy2+3x+1, x3y2+x2, x4y+z2+x y4+3);
\print(b);
```

Результат выполнения:

```
in: SPACE = Q[x, y, z];
```

```
b = gbasis(x^4y^3 + 2xy^2 + 3x + 1, x^3y^2 + x^2, x^4y + z^2 + xy^4 + 3);
print(b);
out: b = [z^2 - x^4 + 3x^2 + (-10)x + 9, x^5 + 9x^2 + (-6)x + 1, y + (-9)x^4 +
(-3)x^3 - x^2 + (-81)x + 27].
```

7.4. Контрольные задания

В системе Mathpar вычислите:

- $f(1) + f(2) + f(3) + f(4) + f(5)$ для $f = -3x^3 - x^2 + x + 2$;
- базис Гребнера полиномиального идеала для полиномов $x^2 + xy$, $4xy^3 - 2xy - 4$, $y^2 - x$, $x^2y^2 + x + y - 6$.

Глава 8

Матричные функции

8.1. Вычисление транспонированной матрицы

Для вычисления транспонированной матрицы для матрицы A необходимо выполнить команду:

`\transpose(A)` или $A^{\mathbf{T}}$.

Пример:

```
SPACE=Z[x];  
A=[[1, 2], [4, 5]];  
B=A^T;  
\print(B);
```

Результат выполнения:

in: $SPACE = Z[x]$;

$$A = \begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix};$$

$$B = A^T;$$

$print(B)$;

out: $B = \begin{pmatrix} 1 & 4 \\ 2 & 5 \end{pmatrix}$.

8.2. Вычисление обратной и присоединенной матрицы

8.2.1. Вычисление обратной матрицы

Для вычисления обратной матрицы для матрицы A необходимо выполнить команду:

```
\inverse(A) или A^{-1}.
```

Примеры:

```
1. SPACE=Z[x];  
A=[[1, 4], [4, 5]];  
B=\inverse(A);  
\print(B);
```

Результат выполнения:

```
in: SPACE = Z[x];
```

$$A = \begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix};$$

$$B = \text{inverse}(A);$$

```
print(B);
```

```
out: B = \left( \begin{array}{cc} (-5)/11 & 4/11 \\ 4/11 & (-1)/11 \end{array} \right).
```

```
2. SPACE=Z[x, y];  
A=[[x+y, x], [y, \cos(x)]];  
B=\inverse(A);  
\print(B);
```

Результат выполнения:

```
in: SPACE = Z[x, y];
```

$$A = \begin{pmatrix} x + y & x \\ y & \cos(x) \end{pmatrix};$$

$$B = \text{inverse}(A);$$

```
print(B);
```

```
out: B = \left( \begin{array}{cc} \frac{\cos(x)}{y \cos(x) + x \cos(x) + (-yx)} & \frac{-x}{y \cos(x) + x \cos(x) + (-yx)} \\ \frac{-y}{(y \cos(x) + x \cos(x) + (-yx))} & \frac{x}{(y \cos(x) + x \cos(x) + (-yx))} \end{array} \right).
```

8.2.2. Вычисление присоединенной матрицы

Для вычисления присоединенной матрицы для заданной матрицы необходимо выполнить команду:

`\adjoint(A)` или \mathbf{A}^{\star} .

Примеры:

```
1. SPACE=Z[x];
A=[[1, 4], [4, 5]];
B=\adjoint(A);
\print(B);
```

Результат выполнения:

in: $SPACE = Z[x]$;

$$A = \begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix};$$

$$B = \text{adjoint}(A);$$

print(B);

out: $B = \begin{pmatrix} 5 & -4 \\ -4 & 1 \end{pmatrix}.$

```
2. SPACE=Z[x, y];
A=[[ \cos(y), \sin(x)], [\sin(y), \cos(x)]];
B=\adjoint(A);
\print(B);
```

Результат выполнения:

in: $SPACE = Z[x, y]$;

$$A = \begin{pmatrix} \cos(y) & \sin(x) \\ \sin(y) & \cos(x) \end{pmatrix};$$

$$B = \text{adjoint}(A);$$

print(B);

out: $B = \begin{pmatrix} \cos(x) & -\sin(x) \\ -\sin(y) & \cos(y) \end{pmatrix}.$

8.3. Вычисление определителя матрицы

Для вычисления определителя матрицы A необходимо выполнить команду:

`\det(A)`.

Примеры:

```
1. SPACE=Z[x];
A=[[1, 4], [4, 5]];
B=\det(A);
\print(B);
```

Результат выполнения:

in: $SPACE = Z[x]$;

$$A = \begin{pmatrix} 1 & 4 \\ 4 & 5 \end{pmatrix};$$

$$B = \det(A);$$

$print(B)$;

out: $B = -11$.

2. $SPACE=R[x]$;

$A=[[3, 4], [3, 1]]$;

$B=\det(A)$;

$\print(B)$;

Результат выполнения:

in: $SPACE = R[x]$;

$$A = \begin{pmatrix} 3 & 4 \\ 3 & 1 \end{pmatrix};$$

$$B = \det(A);$$

$print(B)$;

out: $B = -9$.

3. $SPACE=Z[x, y]$;

$A=[[x^2, y], [4, x+y]]$;

$B=\det(A)$;

$\print(B)$;

Результат выполнения:

in: $SPACE = Z[x, y]$;

$$A = \begin{pmatrix} x^2 & y \\ 4 & x + y \end{pmatrix};$$

$$B = \det(A);$$

$print(B)$;

out: $B = yx^2 - 4y + x^3$.

4. $SPACE=Z[x, y]$;

$A=[[x+y, \sin(x)], [y, \cos(x)]]$;

$B=\det(A)$;

$\print(B)$;

Результат выполнения:

in: $SPACE = Z[x, y]$;

$$A = \begin{pmatrix} x + y & \sin(x) \\ y & \cos(x) \end{pmatrix};$$

$B = \det(A);$
 $\text{print}(B);$
 out: $B = y \cdot \cos(x) + x \cdot \cos(x) - y \cdot \sin(x).$

8.4. Вычисление сопряженной матрицы

Для вычисления сопряженной матрицы необходимо выполнить команду:

`\conjugate(A)` или `A^{\ast}`.

Пример:

```
SPACE=C[x];
A=[[1+\i, 2-\i], [-3, -2\i]];
B=A^{\ast};
\print(B);
```

Результат выполнения:

in: $SPACE = C[x];$

$$A = \begin{pmatrix} 1 + i & 2 - i \\ -3 & -2i \end{pmatrix};$$

$$B = A^*;$$

$\text{print}(B);$

out: $B = \begin{pmatrix} 1 - 1.0i & -3 \\ 2 + 1.0i & 2.0i \end{pmatrix}.$

8.5. Вычисление обобщенной обратной матрицы

Для вычисления обобщенной обратной матрицы Мура-Пенроуза необходимо выполнить команду:

`\genInverse(A)` или `A^{+}`.

Пример:

```
SPACE=Z[x];
A=[[1, 4, 5], [2, 4, 5]];
B=A^{+};
\print(B);
```

Результат выполнения:

in: $SPACE = Z[x];$

$$A = \begin{pmatrix} 1 & 4 & 5 \\ 2 & 4 & 5 \end{pmatrix};$$

$$B = A^+;$$

$print(B);$

$$\text{out: } B = \begin{pmatrix} -1 & 1 & 0 \\ 8/41 & (-4)/41 & 0 \\ 10/41 & (-5)/41 & 0 \end{pmatrix}.$$

8.6. Вычисление ядра оператора и эшелонной формы

8.6.1. Вычисление эшелонной формы матрицы

Для вычисления эшелонной формы матрицы A необходимо выполнить команду:

$\backslash\text{toEchelonForm}(A).$

Примеры:

```
1. SPACE=Z[x];
A=[[1, 4], [4, 5]];
B=\toEchelonForm(A);
\print(B);
```

Результат выполнения:

in: $SPACE = Z[x];$

$$A = \begin{pmatrix} 1 & 4 \\ 4 & 5 \end{pmatrix};$$

$$B = \text{toEchelonForm}(A);$$

$print(B);$

$$\text{out: } B = \begin{pmatrix} -11 & 0 \\ 0 & -11 \end{pmatrix}.$$

```
2. SPACE=Z[x, y];
A=[[ \cos(y), \sin(x)], [\sin(y), \cos(x)]];
B=\toEchelonForm(A);
\print(B);
```

Результат выполнения:

in: $SPACE = Z[x, y];$

$$A = \begin{pmatrix} \cos(y) & \sin(x) \\ \sin(y) & \cos(x) \end{pmatrix};$$

$B = toEchelonForm(A);$

$print(B);$

out: $B = \begin{pmatrix} \cos(y)\cos(x) - \sin(x)\sin(y) & 0 \\ 0 & \cos(y)\cos(x) - \sin(x)\sin(y) \end{pmatrix}.$

8.6.2. Вычисление ядра оператора

Для вычисления ядра оператора матрицы A необходимо выполнить команду:

$\backslash kernel(A).$

Примеры:

1. $SPACE = Z[x];$

$A = [[1, 4], [4, 16]];$

$B = \backslash kernel(A);$

$\backslash print(B);$

Результат выполнения:

in: $SPACE = Z[x];$

$$A = \begin{pmatrix} 1 & 4 \\ 4 & 16 \end{pmatrix};$$

$B = kernel(A);$

$print(B);$

out: $B = \begin{pmatrix} 0 & 4 \\ 0 & -1 \end{pmatrix}.$

2. $SPACE = Z[x, y];$

$A = [[x+y, x], [(x+y)x, x^2]];$

$B = \backslash kernel(A);$

$\backslash print(B);$

Результат выполнения:

in: $SPACE = Z[x, y];$

$$A = \begin{pmatrix} x+y & x \\ (x+y)x & x^2 \end{pmatrix};$$

$B = kernel(A);$

`print(B);`
 out: $B = \begin{pmatrix} 0 & x \\ 0 & -y - x \end{pmatrix}$.

8.7. Вычисление характеристического полинома матрицы

Для вычисления характеристического полинома матрицы A необходимо выполнить команду:

`\charPolynom(A)`.

Примеры:

1. `SPACE=Z[x];`
`A=[[1, 4], [4, 5]];`
`B=\charPolynom(A);`
`\print(B);`

Результат выполнения:

in: $SPACE = Z[x];$
 $A = \begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix};$
 $B = charPolynom(A);$
`print(B);`
 out: $B = x^2 + (-6)x + (-11).$

2. `SPACE=Z[t, x, y];`
`A=[[\cos(y), \sin(x)], [\sin(y), \cos(x)]];`
`B=\charPolynom(A);`
`\print(B);`

Результат выполнения:

in: $SPACE = Z[x, y];$
 $A = \begin{pmatrix} \cos(y) & \sin(x) \\ \sin(y) & \cos(x) \end{pmatrix};$
 $B = adjoint(A);$
`print(B);`
 out: $B = t^2 + (-\cos(x) - \cos(y))t + \cos(y)\cos(x) - \sin(x)\sin(y).$

8.8. LDU-разложение

Для вычисления LDU-разложения матрицы A , нужно выполнить команду:

```
\LDU(A).
```

Результат — вектор из трёх матриц $[L, D, U]$. Здесь L — нижняя треугольная матрица, U — верхняя треугольная матрица, D — матрица перестановок, умноженная на матрицу, которая является обратной к диагональной матрице. Если элементы матрицы A из коммутативной области R , то и элементы матриц L, D^{-1}, U также принадлежат области R .

Примеры:

```
1. SPACE=Z[x];
A=[[0, 1, 0], [4, 5, 1],[1, 1, 1]];
B=\LDU(A);
\print(B);
```

Результат выполнения:

```
in: SPACE = Z[x];
```

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 4 & 5 & 1 \\ 1 & 1 & 1 \end{pmatrix};$$

$$B = LDU(A)$$

```
print(B);
```

```
out: B =
```

$$\left[\begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ -1 & 1 & 3 \end{pmatrix}, \begin{pmatrix} 0 & 1/16 & 0 \\ 1/4 & 0 & 0 \\ 0 & 0 & 1/12 \end{pmatrix}, \begin{pmatrix} 4 & 5 & 1 \\ 0 & 4 & 0 \\ 0 & 0 & 3 \end{pmatrix} \right].$$

```
2. SPACE=Z[x];
A=[[1, 4,0,1], [4, 5,5,3],[1,2,2,2],[3,0,0,1]];
B=\LDU(A);
\print(B);
```

Результат выполнения:

```
in: SPACE = Z[x];
```

$$A = \begin{pmatrix} 1 & -4 & 0 & 1 \\ 4 & 5 & 5 & 3 \\ 1 & 2 & 2 & 2 \\ 3 & 0 & 0 & 1 \end{pmatrix};$$

$$\begin{aligned}
& B = LDU(A) \\
& \text{print}(B); \\
\text{out: } B = & \left[\begin{array}{c} \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 4 & -11 & 0 & 0 \\ 1 & -2 & -12 & 0 \\ 3 & -12 & 60 & -60 \end{array} \right) \\ \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1/(-11) & 0 & 0 \\ 0 & 0 & 1/132 & 0 \\ 0 & 0 & 0 & 1/720 \end{array} \right) \\ \left(\begin{array}{cccc} 1 & 4 & 0 & 1 \\ 0 & -11 & 5 & -1 \\ 0 & 0 & -12 & -13 \\ 0 & 0 & 0 & -60 \end{array} \right) \end{array} \right].
\end{aligned}$$

```

3. SPACE=Z[x,y];
A=[[cos(y),sin(x)], [sin(y),cos(x)]];
B=\LDU(A);
\print(B);

```

Результат выполнения:

in: $SPACE = Z[x, y];$

$$A = \begin{pmatrix} \cos(y) & \sin(x) \\ \sin(y) & \cos(x) \end{pmatrix}$$

$$B = LDU(A)$$

$\text{print}(B);$

$$\text{out: } B = \left[\begin{array}{c} \left(\begin{array}{cc} \cos(y) & 0 \\ \sin(y) & \cos(y)\cos(x) + (-\sin(x)\sin(y)) \end{array} \right) \\ \left(\begin{array}{cc} 1/\cos(y) & 0 \\ 0 & 1/((\cos(y))^2\cos(x) + (-1\cos(y)\sin(x)\sin(y))) \end{array} \right) \\ \left(\begin{array}{cc} \cos(y) & \sin(x) \\ 0 & \cos(y)\cos(x) + (-\sin(x)\sin(y)) \end{array} \right) \end{array} \right].$$

8.9. Разложение Брюа

Для вычисления разложения Брюа матрицы A , нужно выполнить команду:

`\BruhatDecomposition(A)`.

Результат — вектор из трёх матриц $[V, D, U]$. Здесь V и U — верхние треугольные матрицы, D — матрица перестановок, умноженная на матрицу, которая является обратной к диагональной матрице. Если элементы матрицы A из коммутативной области R , то и элементы матриц V, D^{-1}, U также принадлежат области R .

Примеры:

1. `SPACE=Z[x];`

`A=[[1, 4,0,1], [4, 5,5,3],[1,2,2,2],[3,0,0,1]];`

`B=\BruhatDecomposition(A);`

`\print(B);`

Результат выполнения:

in: `SPACE = Z[x];`

$$A = \begin{pmatrix} 1 & -4 & 0 & 1 \\ 4 & 5 & 5 & 3 \\ 1 & 2 & 2 & 2 \\ 3 & 0 & 0 & 1 \end{pmatrix};$$

`B = BruhatDecomposition(A)`

`print(B);`

$$\text{out: } B = \left[\begin{array}{c} \begin{pmatrix} -24 & 0 & 12 & 1 \\ 0 & 60 & 15 & 4 \\ 0 & 0 & 6 & 1 \\ 0 & 0 & 0 & 3 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 1/(-144) & 0 \\ 0 & 0 & 0 & 1/(-1440) \\ 0 & 1/18 & 0 & 0 \\ 1/3 & 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 3 & 0 & 0 & 1 \\ 0 & 6 & 6 & 5 \\ 0 & 0 & -24 & -16 \\ 0 & 0 & 0 & 60 \end{pmatrix} \end{array} \right].$$

2. `SPACE=Z[x,y];`

```
A=[[cos(y), sin(x)], [sin(y), cos(x)]];
B=\BruhatDecomposition(A);
\print(B);
```

Результат выполнения:

in: $SPACE = Z[x, y];$

$$A = \begin{pmatrix} \cos(y) & \sin(x) \\ \sin(y) & \cos(x) \end{pmatrix}$$

$$B = \text{BruhatDecomposition}(A)$$

$\text{print}(B);$

out: $B =$

$$\begin{bmatrix} \begin{pmatrix} -\cos(y)\cos(x) + \sin(x)\sin(y) & \cos(y) \\ 0 & \sin(y) \end{pmatrix} \\ \begin{pmatrix} 0 & 1/(-\cos(y)\sin(y)\cos(x) + \sin(x)(\sin(y))^2) \\ 1/\sin(y) & 0 \end{pmatrix} \\ \begin{pmatrix} \sin(y) & \cos(x) \\ 0 & -\cos(y)\cos(x) + \sin(x)\sin(y) \end{pmatrix} \end{bmatrix}.$$

8.10. Контрольные задания

В системе Mathpar для матриц $A \in \mathbb{Q}$, $B \in \mathbb{Z}/(17)\mathbb{Z}[x, y]$

$$A = \begin{pmatrix} 3 & 0 & 7 \\ 0 & -5 & 5 \\ 4 & 17 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & x & y+1 \\ x^2 & 0 & -y-x \\ x^2+y & -x & -3yx \end{pmatrix}$$

вычислите:

- транспонированную матрицу;
- определитель;
- обратную матрицу;
- присоединенную матрицу;
- характеристический полином;
- разложение Брюа;
- LDU разложение;
- ядро оператора и эшелонную форму.

Глава 9

Функции теории вероятностей и математической статистики

9.1. Функции для работы с дискретными случайными величинами

Дискретная случайная величина задается как матрица, имеющая две строки. В первой строке записаны значения случайной величины, во второй — соответствующие им вероятности. То есть каждый элемент второй строки является числом от 0 до 1, и сумма элементов второй строки должна быть равна 1, например, $M = ([1, 2, 3, 4, 5], [0.4, 0.1, 0.1, 0.2, 0.2])$.

Для работы с дискретными случайными величинами определены следующие функции:

`\mathExpectation`(M) вычисляет математическое ожидание дискретной случайной величины M .

`\dispersion`(M) вычисляет дисперсию дискретной случайной величины M .

`\meanSquareDeviation`(M) вычисляет среднее квадратичное

отклонение дискретной случайной величины M .

`\addQU(M1, M2)` складывает две дискретные случайные величины $M1$ и $M2$.

`\multiplyQU(M1, M2)` умножает две дискретные случайные величины $M1$ и $M2$.

`\covariance(M1, M2)` вычисляет коэффициент ковариации двух дискретных случайных величин $M1$ и $M2$.

`\correlation(M1, M2)` вычисляет коэффициент корреляции двух дискретных случайных величин $M1$ и $M2$.

`\plotPolygonDistribution(M, V)` строит многоугольник распределения дискретной случайной величины M .

`\plotDistributionFunction(M, V)` строит функцию распределения дискретной случайной величины M , где V — это матрица из одной строки, 4 элемента которой задают границы графика: $[x1, x2, y1, y2]$.

`\simplifyQU(M)` упрощает дискретную случайную величину M .

Примеры:

```
1. SPACE=R[x, y];
M=[[1, 2], [0.2, 0.8]];
g=\mathExpectation(M);
g1=\dispersion(M);
g2=\meanSquareDeviation(M);
\print(g, g1, g2);
```

Результат выполнения:

```
in: SPACE = R[x, y];
    M =  $\begin{pmatrix} 1 & 2 \\ 0.2 & 0.8 \end{pmatrix}$ ;
    g = mathExpectation(M);
    g1 = dispersion(M);
    g2 = meanSquareDeviation(M);
    print(g, g1, g2);
out: g = 1.8;
     g1 = 0.16;
     g2 = 0.39.
```

```
2. SPACE=R[x, y];
M=[[7, 5, 3, 5, 1], [0.2, 0.1, 0.3, 0.1, 0.3]];
g=\simplifyQU(M);
\print(g);
```

Результат выполнения:

```
in: SPACE = R[x, y];  
    M =  $\begin{pmatrix} 7 & 5 & 3 & 5 & 1 \\ 0.2 & 0.1 & 0.3 & 0.1 & 0.3 \end{pmatrix}$ ;  
    g = simplifyQU(M);  
    print(g);  
out: g =  $\begin{pmatrix} 1 & 3 & 5 & 7 \\ 0.3 & 0.3 & 0.2 & 0.2 \end{pmatrix}$ .
```

```
3. SPACE=R[x, y];  
M1=[[0, 1], [0.33333, 0.66666]];  
M2=[[1, 2], [0.25, 0.75]];  
g=\addQU(M1, M2);  
g1= \multiplyQU(M1, M2);  
\print(g, g1);
```

Результат выполнения:

```
in: SPACE = R[x, y];  
    M1 =  $\begin{pmatrix} 0 & 1 \\ 0.33333 & 0.66666 \end{pmatrix}$ ;  
    M2 =  $\begin{pmatrix} 1 & 2 \\ 0.25 & 0.75 \end{pmatrix}$ ;  
    g = addQU(M1, M2);  
    g1 = multiplyQU(M1, M2);  
    print(g, g1);  
out: g =  $\begin{pmatrix} 1 & 2 & 3 \\ 0.08 & 0.41 & 0.49 \end{pmatrix}$ ;  
    g1 =  $\begin{pmatrix} 0 & 1 & 2 \\ 0.33 & 0.16 & 0.49 \end{pmatrix}$ .
```

```
4. SPACE=R[x, y];  
M=[[-7, -2, 0, 3, 5, 7, 9],  
 [0.3, 0.05, 0.2, 0.1, 0.1, 0.2, 0.05]];  
V=[-10, 10, 0, 1];  
\plotPolygonDistribution(M, V);
```

Результат выполнения:

```
in: SPACE = R[x, y];  
    M =  $\begin{pmatrix} -7 & -2 & 0 & 3 & 5 & 7 & 9 \\ 0.3 & 0.05 & 0.2 & 0.1 & 0.1 & 0.2 & 0.05 \end{pmatrix}$ ;  
    V = [-10, 10, 0, 1];
```

`plotPolygonDistribution(M, V);`
out: рис. 9.1.

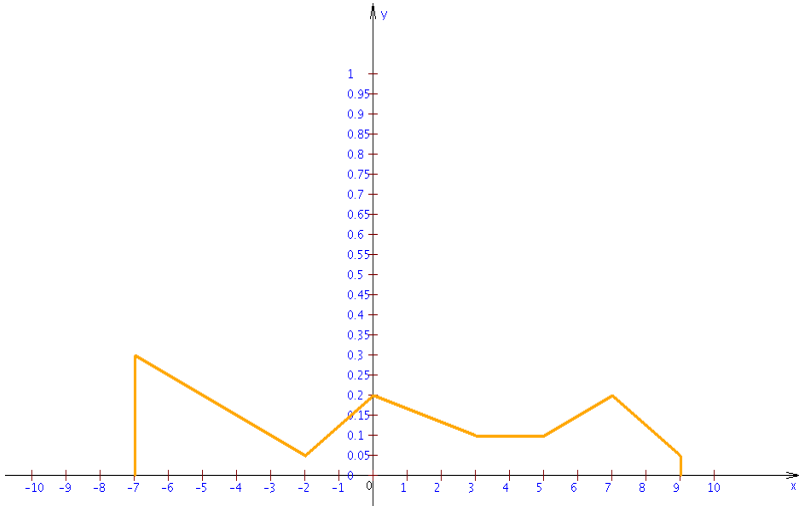


Рис. 9.1. Многоугольник распределения дискретной случайной величины из примера 4

9.2. Функции для выборок

Выборка задается как матрица из одной строки, например, $[1, 7, 10, 15]$.

`\sampleMean(W)` вычисляет выборочное среднее выборки W .

`\sampleDispersion(W)` вычисляет выборочную дисперсию выборки W .

`\covarianceCoefficient(W1, W2)` вычисляет коэффициент ковариации для 2 выборок $W1$ и $W2$.

`\correlationCoefficient(W1, W2)` вычисляет коэффициент корреляции для 2 выборок $W1$ и $W2$.

Пример:

`SPACE=R[x, y];`

`W1=[0, 1];`

```

W2=[1, 2];
g=\sampleMean(W1);
g1=\sampleDispersion(W1);
g2=\covarianceCoefficient(W1, W2);
g3=\correlationCoefficient(W1, W2);
\print(g, g1, g2, g3);

```

Результат выполнения:

```

in: SPACE = R[x, y];
    W1 = [0, 1];
    W2 = [1, 2];
    g = sampleMean(W1);
    g1 = sampleDispersion(W1);
    g2 = covarianceCoefficient(W1, W2);
    g3 = correlationCoefficient(W1, W2);
    print(g, g1, g2, g3);
out: g = 0.5;
     g1 = 0.25; g2 = 0.25; g3 = 1.00.

```

9.3. Контрольные задания

Заданы дискретные случайные величины M и K :

M					K				
1	1.2	1.4	1.6	1.8	0.9	1.0	1.1	1.2	1.3
0.1	0.1	0.3	0.4	0.1	0.2	0.3	0.2	0.2	0.1

В системе Mathprog найдите:

- математическое ожидание случайных величин M и K ;
- дисперсию случайных величин M и K ;
- среднее квадратичное отклонение случайных величин M и K ;
- сумму, произведение случайных величин M и K ;
- коэффициент ковариации случайных величин M и K ;
- коэффициент корреляции случайных величин M и K ;
- постройте многоугольник распределения дискретной случайной величины M и ее функции распределения.

Глава 10

Операторы управления. Процедурное программирование

10.1. Процедуры и функции

Система Mathprog позволяет создавать свои процедуры и функции. Для этого используется команда `\procedure`. После команды указывается имя процедуры, и в фигурных скобках описывается сама процедура.

Пример:

```
\procedure myProc2() {  
    d=4; \print(d); }  
\procedure myProc(c, d) {  
    if (c < d) {\return d;}  
    else{\return d+5;}}  
\myProc2();  
a=10;  
c=\myProc(5+a, a);  
\print(a, c);
```

Результат выполнения:

$d = 4; a = 10; c = 15.$

10.2. Операторы ветвления и циклов

Система Mathpraг дает возможность использовать операторы ветвления и циклов.

`\if() { } \else { }` — оператор ветвления;
`\while() { }` — оператор цикла с предусловием;
`\for(; ;) { }` — оператор цикла со счетчиком.

Примеры:

1. `a=5; b=1;`
`if(b < a) {b=b+a;}`
`else{\print(a, b);}`
`if(b < a) {b=b+a;}`
`else{\print(a, b);}`

Результат выполнения:

$a = 5; b = 6.$

2. `a=0; b=10;`
`while(a < b) { a=a+5; \print(a);}`

Результат выполнения:

$a = 5; a = 10.$

3. `for (i=3; i \le 11; i=i+5) {`
`\print(i);}`

Результат выполнения:

$i = 3; i = 8.$

10.3. Контрольные задания

В системе mathpraг напишите программу:

- для поиска наибольшего коэффициента матрицы;
- для вывода всех чисел от 1 до 3000, которые делятся на 252, а при делении на 101 дают в остатке 3.

Глава 11

Вычисления на суперкомпьютере

Для решения вычислительных задач; требующих большого времени вычислений или больших объемов памяти, разработаны специальные функции, которые предоставляют пользователю ресурсы суперкомпьютера. При использовании этих функций вычисления производятся не на одном процессоре, а на выделенном множестве ядер суперкомпьютера, количество которых заказывает пользователь. Имеются следующие функции (парфункции), которые использует суперкомпьютер:

- 1) `\gbasisPar` — вычисление базиса Гребнера;
- 2) `\adjointPar` — вычисление присоединенной матрицы;
- 3) `\adjointDetPar` — вычисление присоединенной матрицы и определителя матрицы;
- 4) `\echelonFormPar` — вычисление ступенчатого вида матрицы;
- 5) `\inversePar` — вычисление обратной матрицы;
- 6) `\detPar` — вычисление определителя матрицы;
- 7) `\kernelPar` — вычисление ядра линейного оператора;
- 8) `\charPolPar` — вычисление характеристического полинома;

- 9) `\multiplyPar` — вычисление произведения матриц;
 10) `\multiplyPar` — вычисление произведения полиномов.

До применения любой из этих функций пользователь должен указать параметры, определяющие параллельное окружение:

TOTALPROCNUMBER — общее количество процессоров (ядер), которые выделяются для вычислений;

NODEPROCNUMBER — количество ядер на одном узле;

CLUSTERTIME — максимальное время (в минутах) выполнения программы, после истечения которого программа принудительно завершится.

Для задания количества ядер на одном узле пользователь должен знать, какой кластер используется, сколько ядер на узле ему доступно. По умолчанию параметры *TOTALPROCNUMBER* и *NODEPROCNUMBER* устанавливаются так, чтобы использовались все ядра одного узла, а *CLUSTERTIME* = 1.

Возможность менять число ядер на одном узле является важной особенностью, так как оперативная память на одном узле используется всеми *NODEPROCNUMBER* ядрами. Следовательно, можно регулировать размер оперативной памяти, который доступен одному ядру.

11.1. Параллельные полиномиальные вычисления

Для параллельного вычисления произведения полиномов надо использовать команду:

`\multiplyPar(p1, p2)`, где $p1, p2$ — входные полиномы.

Пример:

```
TOTALPROCNUMBER=4;
NODEPROCNUMBER=2;
CLUSTERTIME=1;
\multiplyPar(p1, p2);
```

Для параллельного вычисления базиса Гребнера необходимо использовать команду:

`\gbasisPar([p1, p2, p3, ..., pn])`, где $p1, p2, p3, \dots, pn$ — полиномы.

Пример:

```
\gbasisPar([x^2+2x+1, x^4+x-2]);
```

11.2. Параллельные матричные вычисления

Для параллельного вычисления произведения матриц $m1$ и $m2$ необходимо использовать команду:

```
\multiplyPar(m1, m2).
```

Для параллельного вычисления присоединенной матрицы для матрицы m можно использовать команду:

```
\adjointPar(m).
```

Аналогично для матрицы m можно выполнить вычисление ступенчатого вида `\echelonFormPar(m)`, вычисление определителя `\detPar(m)`, вычисление ядра оператора `\kernelPar(m)`, вычисление характеристического полинома `\charPolPar(m)`. Команда `\adjointDetPar(m)` позволяет вычислить присоединенную матрицу и определитель одновременно.

11.3. Контрольные задания

В системе `Mathprg` задайте квадратную матрицу размера 500, используя генерацию матриц. На суперкомпьютере для этой матрицы вычислите:

- определитель;
- обратную матрицу;
- присоединенную матрицу;
- характеристический полином;
- ступенчатый вид;
- ядро оператора.

Глава 12

Список операторов

Правило образования наименований математических объектов:

Заглавные и строчные буквы всюду различаются. Пользователь может давать любые имена для математических объектов. Однако эти имена не должны совпадать с операторами и константами, которые определены в системе. Кроме того, имена объектов, умножение которых некоммукативно, например, векторов и матриц, должны начинаться с заглавных латинских букв, а все остальные имена объектов — со строчных букв. Это дает возможность сразу после ввода автоматически получать упрощенное выражение.

Приведем список основных операторов системы Mathpra:

`\clean` — удаление всех введенных имен объектов. Если в операторе перечислены имена объектов, то удаляются только объекты с этими именами.

Инфиксные арифметические операторы:

`+` — сложение;
`-` — вычитание;
`/` — деление;
`*` — умножение (можно использовать пробел вместо знака умножения);
`\times` — некоммукативное умножение.

Постфиксные арифметические операторы:

`!` — факториал;

$x^{\wedge}\{n\}$ — возведение в степень;

Инфиксные операторы сравнения:

`\le` — меньше или равно;

`>` — больше;

`\ge` — больше или равно;

`==` — равно;

`\ne` — неравно.

Инфиксные логические операторы:

`\lor` — дизъюнкция;

`&` — конъюнкция;

`\neg` — отрицание.

Основные префиксные операторы:

`\d` — символ дифференцирования при записи дифференциального уравнения;

`\D` — производная функции: `\D(f)` и `\D(f, x)` — первая производная по x ; `\D(f, y^3)` — третья производная по y и т. д.;

`\expand` — преобразование выражения в сумму с раскрытием всех скобок в выражении;

`\expandLn` — преобразование в сумму выражения, которое содержит логарифмические, показательные и тригонометрические функции;

`\extendedGCD` — расширенный алгоритм вычисления наибольшего общего делителя (НОД) полиномов. В результате получается вектор, содержащий НОД и дополнительные множители аргументов;

`\GCD` — вычисление НОД полиномов;

`\factor` — представление выражения в виде произведения;

`\factorLn` — представление выражения, содержащего логарифмические и показательные функции, в виде произведения;

`\initCond` — задание начальных условий для системы линейных дифференциальных уравнений;

`\LCM` — вычисление наименьшего общего кратного (НОК) полиномов;

`\lim` — предел выражения;

\print — печать выражений. Аргументами выступают имена выражений, разделенные запятыми. Каждое выражение будет печататься на новой строке;

\printS — печать выражений в одну строчку, для перехода на следующую строчку нужно использовать «\p»;

\plot — построение графика функции, которая задана явно;

\plot3D — построение графика функции двух переменных, которая задана явно;

\paramPlot — построение графика функции, которая задана параметрически;

\tablePlot — построение графика функции, заданной таблицей аргументов и значений;

\prod — произведение (символ \prod);

\randomPolynom — генерация случайного полинома;

\randomMatrix — генерация случайной матрицы;

\randomNumber — генерация случайного числа;

\sequence — задание последовательности;

\Showgraf — построение в одной системе координат графиков функций, которые должны быть определены раньше;

\solveLDE — решение систем линейных дифференциальных уравнений;

\systLAE — задание систем линейных алгебраических уравнений;

\systLDE — задание систем линейных дифференциальных уравнений;

\sum — сумма (символ \sum);

\time — определение процессорного времени в миллисекундах;

\value — вычисление значения выражения при подстановке заданных выражений или чисел вместо переменных кольца.

Операторы процедуры, ветвления и цикла:

\procedure — оператор объявления процедуры;

\if() { } \else { } — оператор ветвления;

\while() { } — оператор цикла с предусловием;

\for(; ;) { } — оператор цикла с счетчиком.

Матрицы, их элементы и матричные операторы:

[,] — задание вектора (строки);

$[[,], [,]]$ — задание матрицы;
 $A_{\{i,j\}}$ — (i,j) -элемент матрицы A ;
 $A_{\{i,?\}}$ — строка i матрицы A ;
 $A_{\{?,j\}}$ — столбец j матрицы A ;
 $O_{\{n,m\}}$ — нулевая матрица размера $n \times m$;
 $I_{\{n,m\}}$ — $n \times m$ матрица с единицами на главной диагонали;
 $+$, $-$, $*$ — сложение, вычитание, умножение;
`\charPolynom` — характеристический полином;
`\kernel` — ядро оператора (нуль-пространство);
`\transpose` или $A^{\mathbf{T}}$ — транспонированная матрица;
`\conjugate` или $A^{\mathbf{ast}}$ — сопряженная матрица;
`\toEchelonForm` — эшелонная (ступенчатая) форма;
`\det` — определитель;
`\inverse` или A^{-1} — обратная матрица;
`\adjoint` или $A^{\mathbf{star}}$ — присоединенная матрица;
`\genInverse` или $A^{\{+\}}$ — обобщенная обратная матрица
Мурра-Пенроуза;
`\closure` или $A^{\mathbf{times}}$ — замыкание, т.е. сумма $I + A + A^2 + A^3 + \dots$. Для классических алгебр это эквивалентно $(I - A)^{-1}$;
`\LDU` — LDU-представление матрицы. Результат — вектор из трёх матриц $[L, D, U]$. Здесь L — нижняя треугольная матрица, U — верхняя треугольная матрица, D — матрица перестановок, умноженная на обратную к диагональной матрицу;
`\BruhatDecomposition` — разложение Брюа матрицы. Результат — вектор из трёх матриц $[V, D, U]$. Здесь V, U — верхняя треугольная матрица, D — матрица перестановок, умноженная на обратную к диагональной матрицу.

Литература

1. Akritas A.G., Malaschonok G.I. Computation of Adjoint Matrix // Computational Science, ICCS 2006, LNCS 3992, Springer, Berlin, 2006. P. 486-489.
2. Akritas A.G., Malaschonok G.I., P.S.Vigklas. The SVD-Fundamental Theorem of Linear Algebra // Nonlinear Analysis: Modeling and Control, 2006. V. 11, No. 2. P. 123-136.
3. Akritas A.G., Malaschonok G.I. Computations in Modules over Commutative Domain // Computer Algebra in Scientific Computing, Springer, Berlin, 2007. P. 11-23.
4. Борисов И.А. Некоторые подходы к вычислению базисов Гребнера // Исследовательские проекты студентов. Приложение к журналу Вестник Тамбовского университета. Тамбов, 2011. С. 191-194.
5. Валеев Ю.Д. О сложности алгоритмов для разреженных полиномов // Проблемы теоретической кибернетики: тез. докл. XIV Междунар. конф. М.: Изд-во механико-математического факультета МГУ, 2005. С. 52.
6. Егоров А.Ю. Приложение базисов Гребнера в биологии // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2009. Т. 14, вып. 1. С. 290-291.
7. Зуев М.С. Блочные символьные матричные алгоритмы: автореф. дис. канд. физ.-мат. наук. Тамбов: Издательский дом ТГУ им. Г.Р. Державина, 2008.
8. Зуев М.С. Сложность алгоритмов для разреженных полиномиальных матриц // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2007. Т. 12, вып. 1. С. 131.
9. Зуев М.С. О сложности алгоритмов умножения матриц над полиномами // Проблемы теоретической кибернетики: тез. докл. XIV Междунар. конф. М.: Изд-во механико-математического факультета МГУ, 2005. С. 52.
10. Ивашов Д.С. Факторизация полиномов многих переменных // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2011. Т. 16, вып. 1. С. 133-137.
11. Лапаев А.О. Реализация алгоритма Карацубы и оценки сложности // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2008. Т. 13, вып. 1. С. 142.
12. Малашонок Г.И. Конструктивная математика и принцип близкодействия // Вестник Тамбовского университета. Сер.: Есте-

ственные и технические науки. Тамбов, 2011. Т. 16, вып. 1. С. 115-120.

13. Параллельная компьютерная алгебра: учебное пособие в 3 ч. Малашонок Г.И. [и др.]. / Тамбов: Издательский дом ТГУ им. Г.Р. Державина, 2011. Ч. 2.

14. Малашонок Г.И. Компьютерная математика для вычислительной сети // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2010. Т. 15, вып. 1. С. 322-327.

15. Малашонок Г.И., Смирнов Р.А. О композиции функций и машинном представлении // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2010. Т. 15, вып. 1. С. 327-331.

16. Малашонок Г.И., Ивашов Д.С. Об алгоритме факторизации полиномов многих переменных // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2010. Т. 15, вып. 1. С. 331-334.

17. Параллельная компьютерная алгебра: учебное пособие в 3 ч. Малашонок Г.И. [и др.]. / Тамбов: Издательский дом ТГУ им. Г.Р. Державина, 2010. Ч. 1.

18. 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.

19. Малашонок Г.И., Бетин А.А. Вычисление комплексных корней полиномов // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2008. Т. 13, вып. 1. С. 138-141.

20. Малашонок Г.И., Старов М.В. Вычисление матричной степени и матричных функций // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2008. Т. 13, вып. 1. С. 133-138.

21. Малашонок Г.И. О вычислении ядра оператора, действующего в модуле // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2008. Т. 13, вып. 1. С. 129-131.

22. Малашонок Г.И., Зуев М.С. О вычислении обратной матрицы // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2008. Т. 13, вып. 1. С. 115-121.

23. Алгоритмы компьютерной алгебры: учебное пособие в 2 ч.

Малашонок Г.И. [и др.]. / Тамбов: Издательский дом ТГУ им. Г.Р. Державина, 2008. Ч. 1.

24. Малашонок Г.И., Бетин А.А. О вычислении комплексных корней полиномов // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2007. Т. 12, вып. 1. С. 150-152.

25. Малашонок Г.И., Лапаев О.А., Пирютин И.А. Реализация алгоритма Карацубы для умножения целых чисел // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2007. Т. 12, вып. 1. С. 130.

26. Malashonok G.I., Valeev Yu.D. and Lapaev A.O. On the choice of a multiplication algorithm for polynomials and polynomial matrices // Journal of Mathematical Sciences. 2009. V. 168, No. 3. P. 398-416

27. Malaschonok G. I. Project of Parallel Computer Algebra // Tambov University Reports. Series: Natural and Technical Sciences. Tambov, 2010. V. 15. Issue 6. P. 1724-1729.

28. Malaschonok G. I. Fast Generalized Bruhat Decomposition // Computer Algebra in Scientific Computing. 12th International Workshop, CASC 2010. Tsakhkadzor, Armenia, September 2010. Proceedings, 2010. P. 194.

29. Malaschonok G.I. Architecture of ParCA3 project // International conference Polynomial Computer Algebra. St.Petersburg, PDMI RAS, 2010. P. 44-47.

30. Malaschonok G.I. Classes structure for ParCA-2 system // International conference Polynomial Computer Algebra. St.Petersburg, PDMI RAS, 2009. P. 66-68.

31. Малашонок Н.А. Один пример символьного решения системы дифференциальных уравнений в частных производных // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2010. Т. 15, вып. 6. С. 1761-1766.

32. Malaschonok N. An algorithm for symbolic solving of differential equations and estimation of accuracy // Computer Algebra in Scientific Computing. 11th International Workshop, CASC 2009, Kobe, Japan, September 2009. Springer-Verlag Berlin Heidelberg, 2009. P. 213-225.

33. Malaschonok N.A. An algorithm for symbolic solving systems of partial differential equations // International conference Polynomial Computer Algebra. St.Petersburg, PDMI RAS, 2009. P. 109-111.

34. Malaschonok N. An algorithm for symbolic solving of differential

equations and estimation of accuracy // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2009. Т. 14, вып. 4. С. 748-749.

35. Malaschonok N.A. Estimation of accuracy of polynomial roots in the algorithm for symbolic solving of differential equations // International conference Polynomial computer algebra. St.Petersburg, PDMI RAS, 2008. P. 53-56.

36. Malaschonok N.A. Solving differential equations by parallel Laplace method with assured accuracy // Serdica, Journal of Computing, Sofia, Bulgaria, 2007. V. 33, No. 4. P. 20-35.

37. Pereslavytseva O. N. Calculation of the characteristic polynomial of a matrix // Discrete Mathematics and Applications. 2011. V. 21, No. 1. P. 109-129.

38. Переславцева О.Н. О вычислении характеристического полинома матрицы // Дискретная математика. М., 2011. Т. 23, вып. 1. С. 28-45.

39. Переславцева О.Н. Вычисление характеристического полинома для полиномиальных матриц // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2009. Т. 14, вып. 1. С. 274-277.

40. Переславцева О.Н. О вычислении коэффициентов характеристического полинома // Вычислительные методы и программирование: новые вычислительные технологии. М., 2008. Т. 9, №. 1. С. 366-370. URL: <http://num-meth.sr.cc.msu.ru>

41. Переславцева О.Н. Метод вычисления характеристического полинома матрицы // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2008. Т. 13, вып. 1. С. 131-133.

42. Переславцева О.Н. Вычислительные эксперименты с алгоритмами вычисления характеристических полиномов матриц // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2007. Т. 12, вып. 1. С. 126-128.

43. Поздникин А.Г. Полиномиальные вычисления в системах компьютерной алгебры PARCA и FORM: сравнительный анализ // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2011. Т. 16, вып. 1. С. 128-132.

44. Поздникин А.Г. Об одном подходе к арифметике сверхбольших полиномов // Вестник Тамбовского университета. Сер.: Есте-

ственные и технические науки. Тамбов, 2010. Т. 15, вып. 6. С. 1767-1770.

45. Поздникин А.Г. Приложение базисов Гребнера в робототехнике // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2009. Т. 14, вып. 1. С. 288-289.

46. Рыбаков М.А. Решение систем линейных дифференциальных уравнений с кусочно-непрерывными правыми частями с помощью преобразования Лапласа // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2010. Т. 15, вып. 1. С. 339-341.

47. Рыбаков М.А. Решение систем линейных дифференциальных уравнений с постоянными коэффициентами с помощью преобразования Лапласа // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2009. Т. 14, вып. 4. С. 791-792.

48. Сажнева О.А. Решения систем линейных неоднородных и однородных уравнений над кольцом полиномов // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2009. Т. 14, вып. 1. С. 277-281.

49. Смирнов Р.А. Вычисление канонических форм и пределов композиций элементарных функций, заданных в символьном виде // Исследовательские проекты студентов. Приложение к журналу Вестник Тамбовского университета. Тамбов, 2008. С. 345-346.

50. Старов М.В. Реализация метода Фужера вычисления базиса полиномиальных идеалов // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2010. Т. 15, вып. 1. С. 345-346.

51. Старов М.В. Реализация метода Фужера // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2009. Т. 14, вып. 4. С. 802-803.

52. Тарарова С.М. Вычисление неопределённого интеграла дробно-рациональной функции // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2010. Т. 15, вып. 1. С. 353-354.

53. Тарарова С.М. Символьное интегрирование композиций элементарных функций // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2009. Т. 14, вып. 1. С. 283-286.

54. Юрин Ю.Ю. Визуализация функции плотности среды // Исследовательские проекты студентов. Приложение к журналу Вестник Тамбовского университета. Тамбов, 2008. С. 349-350.

55. Юрин Ю.Ю. Визуализация функции плотности среды // Вестник Тамбовского университета. Сер.: Естественные и технические науки. Тамбов, 2010. Т. 15, вып. 1. С. 355-357.

