МИНОБРНАУКИ РОССИИ

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

Институт прикладной математики и компьютерных наук
Кафедра «Прикладная математика и информатика»

Утверждено на заседании кафедры
«Прикладная математика и информатика»
24 января 2022 г., протокол № 5

Заведующий кафедрой
________________________М.В. Грязев

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению практических (семинарских) занятий
по дисциплине (модулю)
«Практикум на электронных вычислительных машинах»
Часть 1

основной профессиональной образовательной программы
высшего образования – программы бакалавриата

по направлению подготовки
01.03.02 Прикладная математика и информатика

с направленностью (профилем)
Прикладная математика и информатика

Форма обучения: очная

Идентификационный номер образовательной программы: 010302-01-22

Тула 2022 год

Разработчик методических указаний

Смирнов О.И., доцент каф. ПМиИ, к.ф.-м.н., доцент

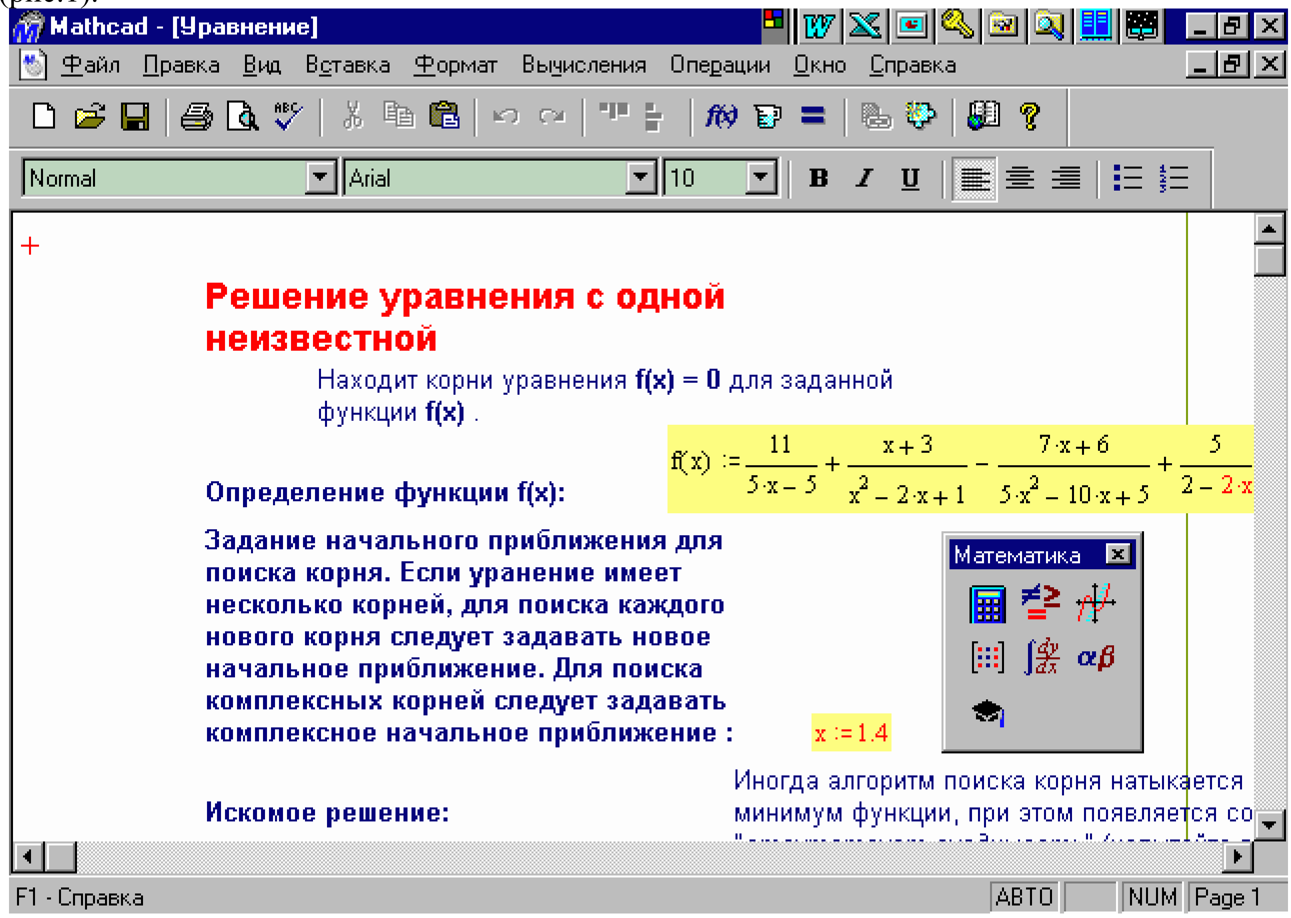
(ФИО, должность, ученая степень, ученое звание) *(подпись)*

1 семестр
Математический пакет Maxima

Занятие 1.
Средства стандартных вычислений в пакете Maxima.

Maxima является уникальной системой для научных и инженерных расчетов и позволяет работать с формулами, числами, текстом и графиками. С помощью Maxima можно решить почти любую математическую задачу символьно либо численно.

Рабочее окно Maxima является стандартным окном Windows – приложения. Сверху располагается строка заголовка, где приводится название приложения (Maxima) и имя рабочего листа (документа). Ниже располагается строка меню, где все команды пакета сгруппированы по функциональному назначению. Ниже строки меню располагаются панели инструментов – Стандартная и Форматирование. Панель Математика обычно находится в рабочей зоне документа. Для открытия других панелей необходимо выполнить последовательность – Вид/Панели. С помощью курсора можно менять положение и форму панелей. Самая нижняя строка – строка состояния, где приводится информация о текущих режимах. Для удобства работы с большими листами имеются линейки вертикальной и горизонтальной прокрутки (рис.1).



Основным документом Maxima является рабочий лист. Maxima допускает ввод формул и текста в любом месте рабочего документа. Каждое математическое выражение или фрагмент текста является областью. Рабочий документ Maxima есть совокупность таких областей. Maxima создает три типа областей – текстовое, математическое и графическое.

Определение значений переменных

Для того, чтобы можно было вычислить выражение, зависящее от каких-либо переменных, значения этих переменных должны быть определены. Для этого нужно:

1. Ввести имя переменной.

2. Ввести двоеточие [:], что приведёт к появлению знака присваивания [:=] и следующего за ним поля ввода.

3. Напечатать в поле ввода число или выражение. Maxima вычислит соответствующее значение и присвоит его имени переменной.

Вычисление выражения

Для того, чтобы получить числовой результат, нужно:

1. Ввести в рабочий документ выражение, значение которого нужно вычислить.

2. Ввести знак равенства [=], после чего Maxima вычисляет введенное ранее выражение и выводит в рабочий документ результат вычислений. Для вычисления выражения в ручном режиме необходимо нажать клавишу [F9].

Построение формул и редактирование документа

Редактор формул Mathcad строит математические выражения, собирая отдельные его части, используя правила старшинства операций и некоторые другие правила, которые упрощают ввод знаменателей, показателей степени и выражений в радикалах. Для этого в Maxima имеется особый маркер ввода. Рассмотрим, для примера, как набирается выражение:

Набираем:

2 z − 3 b^3 .

Пока мы печатали символы подряд, согласно правилам старшинства операций. Теперь нам надо имеющееся выражение сделать числителем. Для этого надо использовать выделяющую рамку. Чтобы она появилась, следует нажать клавишу [-](пробел). Последующие нажатия на [-] увеличивают рамку, и она охватит целиком все необходимое пространство.

Далее нажимаем клавишу [/] для появления на экране дробной черты. Теперь печатаем:

a + z^3 .

Нажимаем несколько раз [-] пока рамка не охватит z^3, после чего набираем:

-8.

Теперь все выражение набрано.

Основные потребности в редактировании формул можно удовлетворить, если уметь изменять буквы и числа, а также изменять или вставлять операторы.

Вставка оператора

Прежде чем вставлять оператор, вспомните, что всё заключенное выделяющей рамкой становится первым операндом следующего оператора:

1. Щёлкните на операнде, чтобы заключить его в выделяющую рамку. Используйте клавишу - [↑], если нужно увеличить выделяющую рамку.

2. Наберите комбинацию клавиш, задающую оператор.

3. Чтобы вставить оператор перед выделенным выражением, нажмите клавишу [Ins] прежде, чем начнёте печатать.

Замена оператора:

1. Щёлкните на операторе, чтобы заключить выражение в выделяющую рамку. Убедитесь, что в рамку заключён оператор вместе с его операндами.

2. Нажмите [BkSp], чтобы удалить оператор.

3. Напечатайте новый оператор.

Вставка и удаление скобок:

1. Заключите выражение в выделяющую рамку.

2. Нажмите апостроф ['] чтобы вставить скобки.

3. Нажмите [Del], чтобы удалить пару скобок

Вставка и удаление знака:

1. Щёлкните на выражении, чтобы заключить его в выделяющую рамку. Используйте - [↑], если нужно увеличить выделяющую рамку.

2. Для вставки знака нажмите [Ins]. Нажмите знак минус.

3. Для удаления знака заключите выражение целиком, включая знак минус, в выделяющую рамку. Нажмите [BkSp].

Изменение стиля переменных и констант

Чтобы изменить гарнитуру, размер, начертание, расположение или цвет шрифта, которым записываются переменные или константы:

1. Выберите Уравнение из меню Формат.

2. В диалоговом окне выберите Variables для переменных или Constants для констант и нажмите кнопку Изменить.

3. Выберите необходимые свойства шрифта в диалоговом окне и нажмите "OK" . Необходимые параметры можно также задать через панель

Форматирование.

Управление отображением чисел во всём документе:

1. Выберите Результат из меню Формат.

2. В диалоговом окне задайте значения отображением чисел.

Итерационные вычисления

В Maxima есть специальный тип переменных — дискретные аргументы. Такие переменные принимают целые значе ния. Maxima вычисляет заданное выражение столько раз, сколько значений содержит дискретный аргумент.

Рассмотрим пример. Пусть пользователю необходимо вычислить функцию y=10+x2 при аргументе, принимающем значения 0, 1, … ,10. Это задание выполняется так:

j:0;10 — задание дискретного аргумента.

y[j:10+j^2 — вычисление функции при фиксированном аргументе.

В результате набора “;” на экране появляется “..”. Для ввода нижнего индекса используется квадратная скобка — “[”. Если набрать: y= , то перед пользователем появится вектор значений y — вертикальная таблица значений y с одним столбцом.

Построение формул и редактирование документа

Редактор формул Mathcad строит математические выражения, собирая отдельные его части, используя правила старшинства операций и некоторые другие правила, которые упрощают ввод знаменателей, показателей степени и выражений в радикалах. Для этого в Maxima имеется особый маркер ввода. Рассмотрим, для примера, как набирается выражение:

Набираем:

2 z − 3 b^3 .

Пока мы печатали символы подряд, согласно правилам старшинства операций. Теперь нам надо имеющееся выражение сделать числителем. Для этого надо использовать выделяющую рамку. Чтобы она появилась, следует нажать клавишу [-](пробел). Последующие нажатия на [-] увеличивают рамку, и она охватит целиком все необходимое пространство.

Далее нажимаем клавишу [/] для появления на экране дробной черты. Теперь печатаем:

a + z^3 .

Нажимаем несколько раз [-] пока рамка не охватит z^3, после чего набираем:

-8.

Теперь все выражение набрано.

Занятие 2.
Средства обработки последовательностей, векторных и матричных величин.

Векторы и матрицы.

Maxima поддерживает два вида массивов – одномерные (векторы) и двумерные (матрицы). Элементами массива могут быть числа, строки, математические выражения и даже другие массивы. Основные операции для работы с векторами и матрицами собраны на панели математических инструментов Matrix. Учтите, что элементы матрицы по умолчанию нумеруются с 0, если хотите, чтобы элементы матрицы нумеровались с 1, нужно в начале документа ввести ORIGIN:=1.

Способы задания матрицы.

Матрицу можно целиком ввести с клавиатуры, либо с помощью функциональной зависимости элемента массива от его индексов.

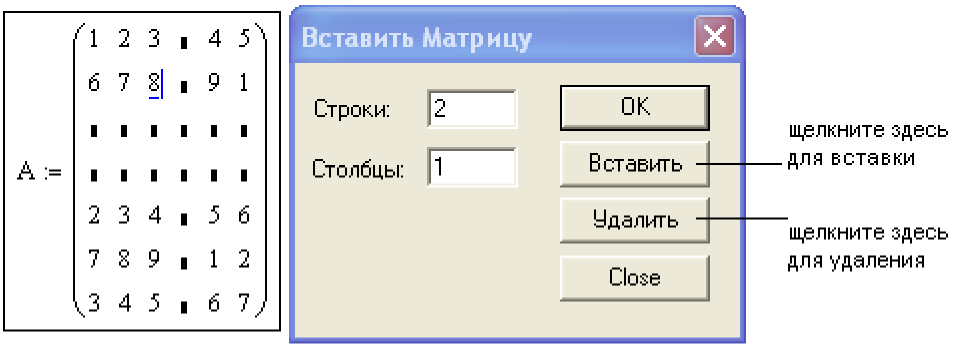
<u>Введение элементов матрицы с клавиатуры</u>

Для того чтобы ввести элементы матрицы с клавиатуры, выберите команду меню Insert →Matrix или на панели инструментов Matrix щелкните на кнопке Matrix or Vector. В открывшемся диалоговом окне введите количество строк (Rows) и столбцов (Columns). При нажатии клавиши OK появится шаблон матрицы, в который можно вводить ее элементы.

Это же диалоговое окно (Insert Matrix или Вставить Матрицу) позволяет добавлять и удалять несколько строк и столбцов в уже имеющейся матрице.

Для того, чтобы добавить строки и столбцы в матрицу, установите курсор на элемент матрицы, справа от которого вы хотите вставить столбцы и ниже которого вы хотите вставить строки. Введите количество вставляемых строк и столбцов и щелкните на кнопке Insert (Вставить).

Например:



Для того, чтобы удалить строки и столбцы из матрицы, установите курсор на на элемент матрицы, справа от которого вы хотите удалить столбцы и ниже которого вы хотите удалить строки. Введите количество удаляемых строк и столбцов и щелкните на кнопке Delete (Удалить). ПРИ ЭТОМ СТРОКА И СТОЛБЕЦ, НА ПЕРЕСЕЧЕНИИ КОТОРЫХ СТОЯЛ УСТАНОВЛЕННЫЙ КУРСОР ТОЖЕ БУДУТ УДАЛЕНЫ.

Для доступа к элементу матрицы нужно указать номер строки и столбца нужного элемента в виде индексов.

Если ввести элемент матрицы, которого не существует, то матрица автоматически будет увеличена до размера, вмещающего введенный элемент. Например:

$ORIGIN := 1$

$M_1 := 3 \qquad M = (3)$

$M_2 := 1 \qquad M = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$

$M_{4,3} := 4 \qquad M = \begin{pmatrix} 3 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 4 \end{pmatrix}$

Кроме доступа к отдельным элементам матрицы Maxima дает возможность выводить и изменять отдельный столбец или строку матрицы. Для того, чтобы обратиться к столбцу матрицы введите ее имя, щелкните на кнопке с изображением $M^{< >}$ на панели инструментов Matrix и в появившемся поле введите номер столбца.

Для выделения аналогичным образом строки, матрицу нужно предварительно транспонировать. Например:

$A := \begin{pmatrix} 1 & 8 & 4 \\ 6 & 3 & 7 \\ 9 & 1 & 2 \end{pmatrix} \qquad A^{\langle 1 \rangle} = \begin{pmatrix} 1 \\ 6 \\ 9 \end{pmatrix} \qquad \left(A^T\right)^{\langle 2 \rangle} = \begin{pmatrix} 6 \\ 3 \\ 7 \end{pmatrix}$

$A^{\langle 2 \rangle} := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \qquad A = \begin{pmatrix} 1 & 0 & 4 \\ 6 & 0 & 7 \\ 9 & 0 & 2 \end{pmatrix}$

Задание матрицы, элементы которой являются функциями индексов.

Например, для того, чтобы задать матрицу A размером 4x 5 , каждый элемент которой равен сумме номера строки и удвоенного номера столбца, то нужно

задать переменные диапазона i:=1..4 , j:=1..5 и задать формулу $A_{i,j}$:=i+2k.

Можно задать такую матрицу без использования переменных диапазона, с помощью специальной функции matrix(4,5,f), описав заранее функцию f. Например:

$f(i,k) := i + 2\cdot k$

$A := matrix(4,5,f)$

$A = \begin{pmatrix} 0 & 2 & 4 & 6 & 8 \\ 1 & 3 & 5 & 7 & 9 \\ 2 & 4 & 6 & 8 & 10 \\ 3 & 5 & 7 & 9 & 11 \end{pmatrix}$

Основные операторы и функции для работы с массивами.

- арифметические операции: поэлементное сложение (+), вычитание (-), матричное умножение (*), которые вводятся с помощью клавиш
- специфические матричные операции: транспонирование, вычисление обратной матрицы, определителя, векторного произведения (только для трехкомпонентных векторов), суммы элементов вектора, которые можно найти в виде кнопок на панели инструментов Matrix, например:

$A := \begin{pmatrix} 1 & 2 & 8 \\ 7 & 3 & 1 \\ 5 & 2 & 4 \end{pmatrix} \qquad A^T = \begin{pmatrix} 1 & 7 & 5 \\ 2 & 3 & 2 \\ 8 & 1 & 4 \end{pmatrix} \qquad A^{-1} = \begin{pmatrix} -0.227 & -0.182 & 0.5 \\ 0.523 & 0.818 & -1.25 \\ 0.023 & -0.182 & 0.25 \end{pmatrix} \qquad |A| = -44$

$U := \begin{pmatrix} 4 \\ 3 \\ 2 \end{pmatrix} \qquad V := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad A\cdot V = \begin{pmatrix} 29 \\ 16 \\ 21 \end{pmatrix} \qquad U\cdot V = 16 \qquad U \times V = \begin{pmatrix} 5 \\ -10 \\ 5 \end{pmatrix} \qquad \sum V = 6$

Некоторые матричные операции заданы в Maxima в виде функций (для их вставки следует выбрать команду меню Insert →Function→Vector and Matrix). Рассмотрим некоторые из них:

- identity(n) – возвращает единичную матрицу размера nxn;
- diag (v) – возвращает диагональную матрицу, у которой на диагонали расположены элементы вектора v;
- rank (M) – возвращает ранг матрицы M;
- tr (M) – возвращает след (сумму диагональных элементов) матрицы M;
- norme (M) – возвращает евклидову норму матрицы M (корень из суммы квадратов всех элементов).

Оператор векторизации.

В Maxima массивы используют для хранения различных наборов значений. Для таких массивов редко используются матричные операции, чаще нужно применить ту или иную скалярную операцию ко всем элементам массива. Для этого используется оператор векторизации (Vectorize). На экране этот оператор изображается в виде стрелки над выражением, к которому он применен. Например, чтобы перемножить поэлементно две матрицы, нужно поставить вектор над записью их произведения.

Объединение матриц и выделение подматрицы.

В Maxima можно присоединять матрицы СПРАВА с помощью функции augment (аргументами может быть любое количество матриц с одинаковым количеством строк) и СНИЗУ с помощью функции stack (аргументами должны быть матрицы с одинаковым количеством столбцов). Например:

$$A := \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \qquad B := \begin{pmatrix} 9 & 8 & 7 & 6 & 5 \\ 7 & 3 & 2 & 1 & 0 \\ 8 & 5 & 9 & 6 & 3 \end{pmatrix}$$

$$C := \mathrm{augment}(A, B)$$

$$C = \begin{pmatrix} 1 & 2 & 9 & 8 & 7 & 6 & 5 \\ 3 & 4 & 7 & 3 & 2 & 1 & 0 \\ 5 & 6 & 8 & 5 & 9 & 6 & 3 \end{pmatrix}$$

$$A := \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad B := \begin{pmatrix} 9 & 8 \\ 7 & 3 \\ 8 & 5 \end{pmatrix}$$

$$C := \mathrm{stack}(A, B) \qquad C = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 9 & 8 \\ 7 & 3 \\ 8 & 5 \end{pmatrix}$$

Выделение подматрицы осуществляется с помощью функции submatrix(*M, imin, imax, jmin, jmax*), где *M* – исходная матрица, *imin, imax* – номера первой и последней строк исходной матрицы, входящих в выделяемый блок, *jmin, jmax* – номера первого и последнего столбцов исходной матрицы, входящих в выделяемый блок.

Собственные вектора и собственные числа.

Для поиска собственных векторов и собственных чисел в Maxima предусмотрены следующие функции:

- *eigenvals(A)* – собственные числа матрицы A;
- *eigenvecs(A)* – собственные векторы матрицы A;

- *eigenvec (A,λ)* – собственный вектор, соответствующий собственному значению λ. Например:

$$A := \begin{pmatrix} 1 & 2 & 8 \\ 7 & 3 & 1 \\ 5 & 2 & 4 \end{pmatrix} \qquad \mathrm{eigenvals}(A) = \begin{pmatrix} -4 \\ 11 \\ 1 \end{pmatrix} \qquad \mathrm{eigenvec}(A,1) = \begin{pmatrix} 0.236 \\ -0.943 \\ 0.236 \end{pmatrix}$$

$$\mathrm{eigenvecs}(A) = \begin{pmatrix} 0.7 & 0.577 & -0.236 \\ -0.661 & 0.577 & 0.943 \\ -0.272 & 0.577 & -0.236 \end{pmatrix}$$

Занятие 3.
Приемы работы с экспериментальными данными.

При проведении научно-технических расчетов часто используются эмпирические зависимости, причем число точек этих зависимостей ограничено. Неизбежно возникает задача восполнения данных: имеется некоторое количество точек, через которые следует провести кривую. Это не что иное, как классическая задача интерполяции.

Интерполяция – частный случай более общей задачи аппроксимации (приближенного представления), возникающей при замене кривой, описываемой функцией сложной природы, другой кривой, в некотором смысле близкой заданной, имеющей более простые уравнения.

Задача сглаживания кривой возникает, когда данные, используемые для ее восстановления, определены в результате измерений или эмпирически с некоторой погрешностью либо представляет кривую, описываемую функцией, недостаточно гладкой (например, недифференцируемой или дифференцируемой всего несколько раз).

Обработка данных средствами Maxima: интерполяция.

Известно, что экспериментальные данные, как правило, задаются дискретно в виде массива данных из двух пар чисел (x_i, y_i). В связи с этим возникает задача аппроксимации дискретных данных непрерывной функцией $f(x)$. В Maxima для обработки экспериментальных данных существуют встроенные функции, которые позволяют выполнять интерполяцию.

Для построения линейной интерполяции служит встроенная функция *linterp*

linterp(x,y, t) – функция, которая аппроксимирует данные векторов *x* и *y* кусочно-линейной зависимостью;

– *x* – вектор действительных данных аргумента;

– *y* – вектор действительных данных значений того же размера;

– *t* – значение аргумента, при котором вычисляется интерполяционная функция.

Замечание: элементы вектора *x* должны быть определены в порядке возрастания.

Чтобы осуществить линейную интерполяцию, надо выполнить следующие действия:

1. Ввести векторы данных *x* и *y*.
2. Определить функцию *linterp (x, в, t)* .
3. Вычислить значение этой функции в необходимых точках, например, *linterp(x,y,2.4)* =3.52 или *linterp(x,в,6)* =5.9, или построить ее график.

Замечание: функция *A (t)* на графику имеет аргумент *t*, а не *x*. Это означает, что функция *A (t)* исчисляется не только при заданных значениях аргумента, а в намного большем количестве аргументов в интервале изменения переменной, что автоматически обеспечивает Mathcad. Mathcad, по умолчанию, соединяет точки графика прямыми линиями, осуществляет их линейную интерполяцию.

В большинстве практических приложений желательно соединить экспериментальные точки не ломанной линией, а гладкой кривой. Лучше всего для этих целей подходит интерполяция кубическими сплайнами, т.е. отрезками кубических парабол. Коэффициенты полиномов рассчитываются так, чтобы непрерывными были первая и вторая производные. Линия, которую описывает сплайн-функция, напоминает по форме гибкую линейку, закрепленную в узловых точках (отсюда и название интерполяции: spline — гибкая линейка).

Для осуществления сплайновой аппроксимации система Maxima предлагает четыре встроенные функции. Три из них служат для получения векторов вторых производных сплайн-функций при различном виде интерполяции:

cspline(VX,VY) — возвращает вектор VS вторых производных при приближении в опорных точках к кубическому полиному;

pspline(VX,VY) — возвращает вектор VS вторых производных при приближении к опорным точкам параболической кривой;

lspline(VX,VY) — возвращает вектор VS вторых производных при приближении к опорным точкам прямой.

Наконец, четвертая функция

interp(s,x,y,t) - функция, которая аппроксимирует данные векторов *x* и *y* кубическими сплайнами;

– s - вектор вторых производных, созданный одной из функций *cspline, pspline* или *lspline*;
– *x* - вектор действительных данных аргумента, элементы которого расположены в порядке возрастания;
– *y* - вектор действительных значений того же размера;
– *t* - значение аргумента, при котором исчисляется функция, которая интерполируется.

Перед применением функции *interp* необходимо предварительно определить первый из ее аргументов - векторную переменную *s*. Выполняется это с помощью одной из трех встроенных функций тех же аргументов *(x,y)*.

– *ispiine(x,y)* - вектор значений коэффициентов линейного сплайна;
– *pspiine(x,y)* - вектор значений коэффициентов квадратичного сплайна;
– *cspline(x,y)* - вектор значений коэффициентов кубического сплайна;
– *x, y* - векторы данных.

Более сложный тип интерполяции - так называемая интерполяция *В-сплайнами*. В отличие от обычной сплайн-интерполяции, сшивание элементарных В-сплайнов выполняется не в точках *x* и *y*, а в других точках, координаты которых предлагается ввести пользователю. Сплайны могут быть полиномами 1, 2 или 3 степени (линейные, квадратичные или кубические). Применяется интерполяция В-сплайнами точно так же, как и обычная сплайн-интерполяция, разница состоит только в определении вспомогательной функции коэффициентов сплайна.

Построение линейной регрессии

Другой широко распространенной задачей обработки данных является представление их совокупности некоторой функцией y(x). Задача регрессии заключается в получении параметров этой функции такими, чтобы функция приближала облако исходных точек (заданных векторами VX и VY*)* с наименьшей среднеквадратичной погрешностью. Чаще всего используется линейная регрессия, при которой функция y(x) имеет вид

$y(x) = a + b \cdot x$

и описывает отрезок прямой. К линейной регрессии можно свести многие виды нелинейной регрессии при двупараметрических зависимостях y(x).

Для проведения линейной регрессии в систему встроен ряд приведенных ниже функций:

corr(VX,VY) — возвращает скаляр — коэффициент корреляции Пирсона;

interp(VX,VY) — возвращает значение параметра а (смещение линии регрессии по вертикали);

s1ope(VX, VY) — возвращает значение параметра b (наклона линии регрессии).

В Maxima реализована возможность выполнения линейной регрессии общего вида. При ней заданная совокупность точек приближается функцией вида:

$F(x, K1, K2,\dots,Kn) = K1 \cdot F1(x) + K2 \cdot F2(x) + ... + Kn \cdot Fn(x)$.

Таким образом, функция регрессии является линейной комбинацией функций $K1 \cdot F1(x)$, $K2 \cdot F2(x),\dots$, $Kn \cdot Fn(x)$, причем сами эти функции могут быть нелинейными, что резко расширяет возможности такой аппроксимации и распространяет ее на нелинейные функции.

Для реализации линейной регрессии общего вида используется функция

1infit(VX,VY,F).

Эта функция возвращает вектор коэффициентов линейной регрессии общего вида К, при котором среднеквадратичная погрешность приближения облака исходных точек, если их координаты хранятся в векторах VX и VY, оказывается минимальной. Вектор F должен содержать функции F1(x), F2(x),…, Fn(x), записанные в символьном виде.

Введена в новую версию Maxima и функция для обеспечения полиномиальной регрессии при произвольной степени полинома регрессии:

Regress(VX,VY,n)

Она возвращает вектор VS, запрашиваемый функцией interp(VS,VX,VY,x), содержащий коэффициенты многочлена n-й степени, который наилучшим образом приближает «облако» точек с координатами, хранящимися в векторах VX и VY.

На практике не рекомендуется делать степень аппроксимирующего полинома выше четвертой — шестой, поскольку погрешности реализации регрессии сильно возрастают.

Функция regress создаст единственный приближающий полином, коэффициенты которого вычисляются по всей совокупности заданных точек, т. е. глобально. Иногда полезна другая функция полиномиальной регрессии, дающая локальные приближения отрезками полиномов второй степени, — 1oess(VX,VY,span). Эта функция возвращает используемый функцией interp(VS,VX,VY,x) вектор VS, дающий наилучшее приближение данных (с координатами точек в векторах VX и VY) отрезками полиномов второй степени. Аргумент span>0 указывает размер локальной области приближаемых данных (рекомендуемое начальное значение - 0,75). Чем больше span, тем сильнее сказывается сглаживание данных. При больших span эта функция приближается к regress(VX,VY,2).

Maxima позволяет выполнять также многомерную регрессию, самый типичный случай которой — приближение трехмерных поверхностей. Их можно характеризовать массивом значений высот z, соответствующих двумерному массиву Mxy координат точек (x,y) на горизонтальной плоскости. Новых функций для этого не задано. Используются уже описанные функции в несколько иной форме:

regress(Mxy,Vz,n) — возвращает вектор, запрашиваемый функцией

interp(VS,Mxy,Vz,V) для вычисления многочлена n-й степени, который наилучшим образом приближает точки множества Mxy и Vz. Mxy — матрица m·2, содержащая координаты x и y. Vz — m-мерный вектор, содержащий z-координат, соответствующих m точкам, указанным в Mxy;

1oes(Mxy,Vz,span) — аналогична loes(VX,VY,span), но в многомерном случае;

1nterp(VS, Mxy,Vz,V) — возвращает значение z по заданным векторам VS (создается функциями regress или 1oess) и Mxy, Vz и V (вектор координат x и y заданной точки, для которой находится z*).*

Функция для нелинейной регрессии общего вида. Под нелинейной регрессией общего вида подразумевается нахождение вектора K параметров произвольной фучнкции F(x,K1,K2,...,Kn), при котором обеспечивается минимальная среднеквадратичная погрешность приближения облака исходных точек. Для проведения'нелинейной регрессии общего вида используется функция genfit(VX,VY,VS,F). Эта функция возвращает вектор K параметров функции F, дающий минимальную среднеквадратичную погрешность приближения функцией F(x,K1,K2,...,Kn) исходных данных. F должен быть вектором с символьными элементами, содержащими уравнение исходной функции и ее производных по всем параметрам. Вектор VS должен содержать начальные значения элементов вектора K, необходимые для решения системы нелинейных уравнений регрессии итерационным методом.

Функции сглаживания данных.

Данные большинства экспериментов имеют случайные составляющие погрешности. Поэтому часто возникает необходимость статистического сглаживания данных. Ряд функций Maxima предназначен для выполнения операций сглаживания данных различными методами (в их названии имеется слово smooth - гладкий). Вот перечень этих функций:

Medsmooth(VY,n) - для вектора с m действительными числами возвращает m-мерный вектор сглаженных данных по методу скользящей медианы, параметр n задает ширину окна сглаживания (n должно быть нечетным числом, меньшим m);

ksmooth(VX,VY,b) - возвращает n-мерный вектор сглаженных VY, вычисленных на основе распределения Гаусса. VX и VY - n-мерные векторы действительных чисел. Параметр b (полоса пропускания) задает ширину окна сглаживания (b должно в несколько раз превышать интервал между точками по оси x);

supsmooth(VX,VY) - возвращает n-мерный вектор сглаженных VY, вычисленных на основе использования процедуры линейного сглаживания методом наименьших квадратов по правилу k-ближайших соседей с адаптивным выбором k. VX и VY - n-мерные векторы действительных чисел. Элементы вектора VX должны идти в порядке возрастания.

Функция предсказания.

Весьма интересной является функция предсказания predikt(data,k,N), где data - вектор данных, k - степень полинома регрессии и N - число точек. Она по ряду заданных равномерно расположенных точек позволяет рассчитать некоторое число N последующих точек, т. е. по существу выполняет экстраполяцию произвольной (но достаточно гладкой и предсказуемой) зависимости.

Занятие 4.
Графические средства Maxima.

Графики, которые строятся на основе результатов вычислений также рассматриваются как формулы.

В Maxima встроено несколько типов разных графиков, которые можно разбить на две группы: двумерные и трехмерные графики.

Все основные типы графиков и инструменты работы с ними расположены на рабочей панели Graph (Графические) семейства Math (Математические) (рис.2,б):

– График кривой в двумерной декартовой системе координат (X-Y Plot).

– График кривой в полярной системе координат (Polar Plot).

– Поверхность (Surface).

– Контурный график (Contour Plot).

– Столбиковая трехмерная (3D) диаграмма (3D Bar Plot).

– Точечный трехмерный (3D) график (3D Scatter Plot).

– Векторное поле (Vector Field).

Аналогично панели Graph (Графические) список всех типов графиков Maxima расположен в одноименном подменю меню Insert (Вставка).

Двумерные графики

В Maxima существует несколько способов задания кривых в декартовой системе координат, однако первый шаг для всех один и тот же.

Первым шагом есть введения специальной заготовки для будущего графика - так называемой графической области. Ввести графическую область как для декартового, так и для любого другого графика можно: из панели Graph (Графические), командой одноименного меню Insert (Вставка) или нажатием комбинации клавиш Shift+2.

Графическая область представляет собой две вложенные рамки. Во внутренней отображаются непосредственно кривые зависимости. Пространство между рамками служит для визуализации разного рода служебной информации. Графическую область можно увеличивать и уменьшать с помощью специальных маркеров, расположенных на ее внешней рамке. Перемещать по документу и удалять графические области можно так же, как простые формулы. Окно форматирования вида графической области (Properties (Свойства) также целиком совпадает с аналогичным окном для формул. Открыть его можно с помощью одноименной команды контекстного меню графика (вызывается щелчком правой кнопкой мыши на графической области).

В окне Properties (Свойста) могут быть полезными два параметра, расположенных на вкладке Display.

– Highlight Region (Цветная область). Установив этот флажок можно на палитре Choose Color (Выбор цвета) определить наиболее подходящий цвет заливки для графической области.

– Show Border (Показать границу). Параметр отвечает за отображение внешней границы графической области. По умолчанию граница не визуализируется.

После того как графическая область будет введена, в общем случае нужно задать два размерных вектора, которые определяют значение координат точек. Сделать это можно разными способами.

Наиболее простым методом задания координатной сетки есть так называемый быстрый метод. При его применении пользователь задает только имя переменной и вид функции, а шкалы осей и величину шага между узловыми точками автоматически определяет система. Чтобы построить кривую функции быстрым методом, можно выполнить следующую последовательность действий.

1. Ввести графическую область.

2. В специальном маркере, расположенном в центре под внутренней рамкой графической области, задать имя переменной.

3. В центральный маркер, расположенный по левую сторону от внутренней рамки, ввести функцию или имя функции.

К недостаткам рассмотренного метода относится прежде всего то, что область изменения переменной для всех функций определяется одинаково: от -10 до 10.

Для того, чтобы изменить область изменения, нужно просто уменьшить интервал изменения переменной или функции. Для этого необходимо выделить графическую область щелчком левой кнопки мыши. Непосредственно под крайними значениями (для оси X) или по левую сторону от них (для оси Y) появятся цифры, которые отражают максимальные и минимальные величины координат узловых точек графика. Чтобы изменить их значения необходимо удалить старые величины и ввести другие. Изменения границ по оси X вызывает автоматический перерасчет крайних значений по оси Y.

На практике же, как правило, приходится определять границы сразу по обеим осям. Это связано с тем, что хорошо подобрать интервал по оси значений функции системе удается далеко не всегда. Это можно сделать для настраивания вида графика рассмотренной функции, изменяя диапазон как по оси X, так и по оси Y.

В ряде случаев намного удобнее задать векторы данных самостоятельно. Выполнить это можно с помощью оператора ранжированной переменной (вводится из панели Matrix (Матричные)).

Чтобы задать вектор значений переменной с помощью оператора Range Variable (Ранжированная переменная), выполняется следующая последовательность действий.

1. Ввести имя переменной вместе с оператором присваивания.
2. Задать левую границу интервала построения и поставить запятую.
3. Ввести оператор ранжированной переменной.
4. В левом маркере введенного оператора задать вторую точку на промежутке (тем самым определяется шаг).
5. В правый маркер оператора ранжированной переменной вводится значение правой границы на интервале.

В результате переменная и функция будут заданы в виде двух размерных векторов, по которым будет построен график.

Использование способа построения графика с помощью оператора ранжированной переменной имеет очень важное преимущество перед быстрым методом, поскольку позволяет задавать произвольным образом шаг между узловыми точками.

Построить график в Mathcad можно и по готовым векторам или таблицам данных, полученных, например, при эксперименте или выполнении лабораторной работы.

В Mathcad на одну графическую область можно поместить до 16 кривых. Чтобы добавить к уже имеющемуся графику еще один, можно выполнить следующую последовательность действий.

1. Установить курсор по правую сторону от выражения, которое определяет координаты последнего ряда данных по оси Y (предварительно выделив его).
2. Опустить курсор на строку ниже, нажать на знак запятой (,) и в маркер, который появился, ввести выражение для новой функции или имени функции.

С помощью описанного метода можно построить графики функций одной переменной. Если же кривые, которые нужно отобразить на одной области, зависят от разных переменных, то их, полностью аналогично добавлению новых функций, следует ввести через запятую в нижний маркер в том же порядке, в котором вводились соответствующие им функции.

Задание графиков в полярной системе координат с технической точки зрения не имеет ровно никаких принципиальных отличий от создания графиков на декартовой плоскости. Для начала нужно ввести графическую область. Выполнить это можно или с помощью специальной кнопки Polar Plot (Полярный график) панели Graph (Графические), или комбинацией клавиш Ctrl+7. Как и в случае зависимости X-Y, для полярного графика существует два основных метода построения: быстрый способ построения и использование ранжированных переменных. При задании полярной системы координат по быстрому методу система автоматически определит область изменения угла от 0 до 360°. В отличие от области изменения угла, величину диапазона полярного радиуса можно задать произвольным образом непосредственно на графической области.

Для форматирования графика необходимо дважды нажать на область графика. Для управления отображением линий на графику существует вкладка Traces (Линии) (рис.2, а), где приведен формат каждой линии и элементы управления изменением формата. Поле Legend Label (Описание) задает описание линии, которое отображается, если снять флажок Hide Legend (Закрыть описание) (рис. 2,б). Маркеры для отдельных точек можно выбрать из списка Symbol (Символ), из списка Line (Тип линии) выбирается тип линии, а из списка Color (Цвет) - цвет графика. Список Type (Тип) определяет средство связи отдельных точек графика, а список Weight (Толщина) - толщину линии на графике (рис.2,в).

Форматирование данных графика выполняется с использованием диалогового окна Result Format (рис.2, г).

Аналогично можно построить и отформатировать график в полярных координатах. Для его построения нужно воспользоваться командой Insert/ Graph/Polar Plot.

Трехмерные графики

Для построения трехмерных графиков можно использовать наиболее простой и практически важный, быстрый метод построения трехмерного графика (QuickPlot). В его основе лежит тот же принцип, который используется и при быстром задании двумерной зависимости: пользователь определяет только вид функции, а все параметры построения, такие как шаг между узловыми точками, диапазон шкал осей и система координат, задаются автоматически системой.

Типы трехмерных графиков следующие:

Contour Plot - график линий уровня (график поверхности);

3D Bar Plot - график трехмерной гистограммы;

3D Scatter Plot - график множества точек;

Vector Field Plot - график векторного поля. График векторного поля немного отличается от других типов двумерных графиков. Его содержание заключается в построении некоторого вектора в каждой точке плоскости XY. Чтобы задать вектор на плоскости, необходимы два скалярных числа. Поэтому в Mathcad принято, что векторное поле задает комплексная матрица. Действительные части каждого ее элемента задают проекцию вектора на ось X, а мнимые - на ось Y.

Чтобы создать трехмерный график, нужно нажать кнопку с изображением каждого из типов трехмерных графиков на панели инструментов Graph (Графики). В результате появится пустая область графика с тремя осями (рис. 3) и единым заполнителем в нижнем левом углу. В этот заполнитель ввести или имя z функции z(x,y) двух переменных для быстрого построения трехмерного графика, или имя матричной переменной z, которая задает функцию z(x,y) на плоскости XY.

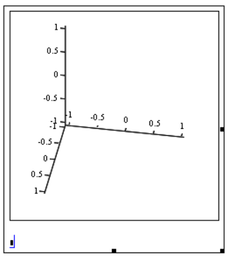


Рисунок 1 – Область для создания трехмерных графиков

Построение с использованием быстрого метода построения трехмерного графика

Последовательность создания трехмерного графика с использованием быстрого метода построения трехмерного графика (QuickPlot) следующая.

1. Сначала необходимо ввести графическую область трехмерного графика. Аналогично зависимости X-Y, сделать это можно тремя стандартными способами: нажатием кнопки

Surface Plot (Поверхность) панели Graph (Графические), использованием одноименной команды меню Insert (Вставка) или нажатием комбинации клавиш Ctrl+2.

Для построения трехмерных графиков существует только один маркер заполнения. В общем случае в нем должен быть прописан массив, который содержит координаты узловых точек по всем трем осям.

2. После того как графическая область введена, следует задать вид функции, которая определяет трехмерную область. В отличие от X-Y-зависимостей, просто ввести ее выражения в маркер нельзя - при этом будет выдано сообщение об ошибке: This variable is undefined (Данная переменная не определенна). В маркер графической области вводится имя заданной функции, для которой строится трехмерный график. Однако, в отличие от двумерного случая, прописанным должен быть лишь непосредственно текст имени, без переменных в скобках.

При использовании данной методики поверхность задается на стандартном интервале от -5 до 5 для переменных. Такой диапазон во многих случаях может быть неприемлемый. Для форматирования параметров графиков быстрого построения существует специальная вкладка Quick Plot Data (Данные графика быстрого построения) окна форматирования трехмерных графиков 3D-Plot Format. Открывается это окно двойным нажатием левой кнопки мыши на графической области или с помощью команды Format (Формат) ее контекстного меню (рис. 2).

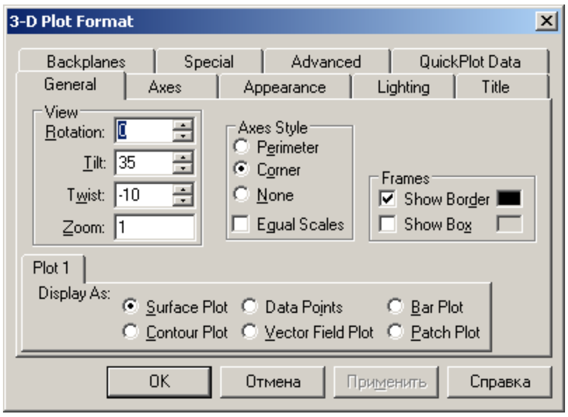


Рис. 2.

Все параметры настройки графика быстрого построения расположены на вкладке Plot1. Вкладка Plot 1 (График 1) содержит три меню настраивания, два из которых: Range 1 и Range 2 (Ряд 1 и Ряд 2), идентичны друг другу. Эти меню отвечают за характеристики сетки построения поверхности вдоль каждой из осей переменных (соответствие переменной ряда определяется последовательностью введения ее при задаче имени функции) и содержат следующие параметры настраивания:

– Start (Начало). В поле данного параметра можно произвольным образом задать начальную точку построения прямоугольника по данной оси.

– End (Конец). В поле данного параметра определяется конечная точка интервала.

– # of Grids (Количество линий сетки). Параметр определяет, на какое количество отрезков будет разбит интервал построения для выбранной переменной (что отвечает числу отображенных линий сетки). Эта величина обратная шагу изменения переменной.

Третье меню вкладки Plot 1 (График 1) - Coordinate System (Система координат) определяет, в какой системе координат следует отобразить данную зависимость. Возможные следующие варианты:

– Cartesian (Декартова). График отображается в декартовой системе координат.

– Spherical (Сферическую). График отображается в сферической системе координат.

– Cylindrical (Цилиндрическая). График отображается в цилиндрической системе координат.

Способ построения трехмерного графика с помощью матрицы значений

Матрица значений представляет собой таблицу из трех колонок: в первой будут расположены координаты точек по оси X, во второй -по оси Y, в третьей - по оси Z. В Mathcad существует специальная функция matrix(m,n,f) (матрица). Функция формирует матрицу, элементы которой равны значениям функции f(x,y), исходя из того условия, что x=i, y=j (т.е. переменные определяются равными соответствующим матричным индексам данного элемента). Количество строк создаваемой матрицы определяется в первом маркере имени функции (параметр m), количество колонок - во втором (параметр n).

Аналогично двумерному случаю, задать поверхность можно, используя оператор ранжированной переменной по готовым матрицам.

Способ построения с помощью специальной матричной функции *CreateMesh*

Функция CreateMesh(F,s,sl,t,tl,sgrid,tgrid,fmap) вводится в маркер графической области и имеет пустые маркеры, в которые последовательно вводятся:

– имя матрицы значений или функции F;
– начальное значение первой переменной s;
– начальное значение второй переменной sl;
– конечное значение первой переменной t;
– конечное значение второй переменной tl;
– число линий сетки по первый переменной sgrid;
– количество линий сетки по второй переменной tgrid;
– карта отображения frnap.

Кроме поверхностей в пространстве можно задавать и разного рода линии. Для этого существует специальная функция CreateSpace(F,t,tl,tgrid,frnap) (Создать пространство). Она имеет пять маркеров, в которые последовательно вводятся имя массива данных или системы параметрических уравнений, начальное и конечное значения параметра, количество разбивок промежутка параметра, карта отображения.

Параметрическое закручивание разрешает создавать графики, которые заданы в параметрической форме. Последовательность действий при использовании алгоритма параметрического закручивания следующая.

1. Задать уравнение любой функции f(x).
2. Задать систему параметрического закручивания и соединить ее в один массив:

A(u,v) :=u,
B(u,v):=f(u)cos(v),
C(u,v):=f(u)sin(v),

$$M(u,v):=\begin{pmatrix} A(u,v) \\ B(u,v) \\ C(u,v) \end{pmatrix}.$$

3. Внести в маркер следующую запись: CreateMesh(M, s,sl,t,tl,sgrid,tgrid).

Занятие 5.
Задание функций. Вычисление интегралов и производных.

Вычисление производных и интегралов

Аналогично большинству других наиболее важных математических операций, в Maxima существует численное и символьное дифференцирование. Символьный метод имеет преимущества в том плане, что результат можно получить в виде функции, которую можно будет использовать в дальнейших расчетах. Численный же подход имеет преимущества в некоторых специфических задачах. Maxima позволяет вычислять как обычную производную, так и производные более высоких порядков, а также частные производные (рис. 1).

Оператор простого дифференцирования на панели Calculus для вычисления первой производной имеет два маркера, принцип заполнения которых следующий: в верхний вводится функция, в нижний - переменная, по которой вычисляется производная.

Рисунок 1 – Диалоговое окно для вычисления производных и интегралов

Результат может быть представлен в символьном виде, если использовать оператор символьного вывода **→,** а потом обратиться к символьному процессору Symbolic/Evaluate (Символика/Вычислить в символах) (рис. 2).

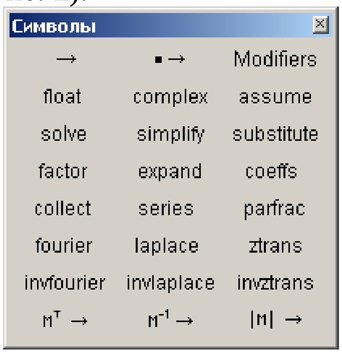


Рисунок 2 –Меню символьного процессора Symbolic для вычисления в символах

При символьном дифференцировании можно оперировать с функциями нескольких переменных. Оператор дифференцирования может соединяться с любым вычислительным или символьным оператором. Особенно полезным есть оператор Simplify, так как выражение производной выдается в неупрощенном виде. Для упрощения ответа следует использовать операторы Collect (Приводить подобные), Factor (Раскладывает выражение на множители) и Expand (Раскрывать скобки).

Чтобы получить численное значение производной в нужной точке исходя из результатов символьного расчета, нужно сделать следующее:

1. Найти функцию производной, используя оператор символьного вывода (**→**).
2. Присвоить переменной соответствующее числовое значение.
3. Скопировать полученное выражение для производной и вычислить его символьно.

Панель Calculus (Вычисление) содержит два оператора интегрирования. Первый, Indefinite Integral (Неопределенный интеграл), позволяет определить вид функции, которая интегрируется (рис. 15). Оператор неопределенного интеграла содержит два маркера, которые заполняются соответственно принятому в математике представлению: в левый вводится функция (или имя функции), под знак дифференциала - переменная интегрирования.

Чаще всего результат интегрирования представляет собой громоздкое выражение. В этом случае его следует упрощать. Наиболее универсальный инструмент, который для этого используется - оператор Simplify (Упростить). Однако иногда выражение можно упростить (оператор Collect), разложив по степеням (оператор Expand) или приведя дробь к общему знаменателю (оператор Factor). Чтобы задействовать нужный символьный оператор, следует выделить выражение интеграла и нажать соответствующую кнопку на панели Symbolic (Символьные). Применить к результату интегрирования можно и сразу несколько символьных операторов.

Нахождение определенного интеграла выполняется подобно тому как вычисляется неопределенный интеграл. Для интегрирования необходимо обратиться на панели Символьные к функции *simplify*. Ввести оператор интегрирования. В соответствующих местах заполнить имя первой переменной и границы интегрирования. Если необходимо вычислить кратные интегралы, то на месте введения функции под интегралом ввести еще один оператор интегрирования, границы интегрирования и подынтегральную функцию. Аналогично выполняется интегрирование по нескольким переменным.

Можно определить интеграл в символьном виде, например,

$$\int_a^b x\,dx \rightarrow \frac{1}{2}b^2 - \frac{1}{2}a^2 .$$

Для числового интегрирования Maxima предлагает воспользоваться встроенными программами вычисления интегралов (рис. 3). Для того, чтобы обратиться к приближенному расчету, необходимо в контекстном меню выбрать один из методов интегрирования.

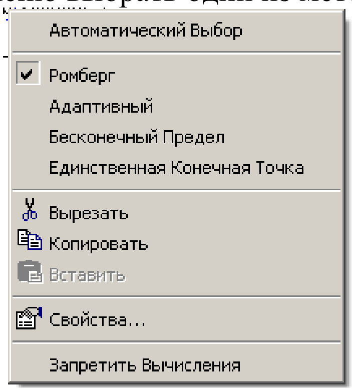


Рисунок 3 –Меню со встроенными программами для числового интегрирования

Занятие 6.
Средства структурного программирования при определении функций.

Для написания программ служит панель математических инструментов Programming, которая может быть вызвана щелчком на соответствующей кнопке панели инструментов Math. После щелчка на любой кнопке панели инструментов Programming в программу вставляется тот или иной оператор.

Все операторы вставляются только щелчком на соответствующей кнопке и ни в коем случае не набираются с клавиатуры!

Создание программы.

Для того, чтобы превратить обычное однострочное выражение в многострочное (программу), достаточно щелкнуть на кнопке с надписью *Add Line*. Это приведет к тому, что в рабочей области документа появится вертикальная черта, а справа от нее 2 поля ввода, в которые можно ввести 2 строки программы. Если далее нужно будет добавить еще строки, то достаточно снова щелкнуть на кнопке *Add Line*.

Внутри программы можно использовать глобальные переменные, но лучше использовать локальные (доступ к которым можно осуществить только из самой программы). Для присваивания значения локальной переменной используется символ ←.

Любая программа должна возвращать некоторое значение, как результат вычислений: это может быть как число так и функция. Возвращаемое значение записывается в последней строке программы, либо с помощью оператора *return*.

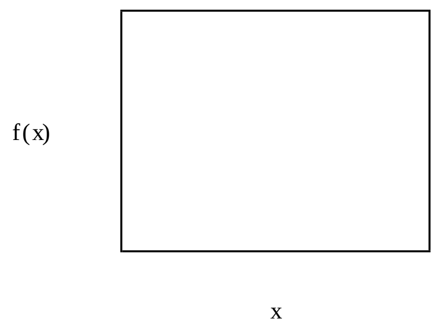
Условный оператор.

Для проверки условий в Maxima служит оператор *if*. Этот оператор имеет 2 поля ввода (справа и слева от слова *if*). В правое поле ввода вводится условие, а в левое поле ввода вводится команда или последовательность команд, которые следует выполнить в случае истинности условия. Если невыполнение условия должно привести к выполнению какого-либо другого программного кода, можно в строке, следующей за оператором *if*, вставить оператор *otherwise*. В поле ввода слева от этого оператора необходимо ввести строку программы, которая будет выполняться только в том случае, если не выполнилось условие, заданное в операторе *if*.

Если в программе введено подряд несколько строк с оператором *if*, то выражение слева от *otherwise* ,будет выполнено только в том случае, если не выполняются условия, заданные во всех операторах *if*.

Рассмотрим, например, описание кусочно-заданной функции:

$$f(x) := \left| \begin{array}{l} x \ \text{ if } \ x > 0 \\ (-x) \ \ \text{otherwise} \end{array} \right.$$



Цикл с условием (*while*).

Оператор while имеет 2 поля ввода (справа и снизу).

- В поле ввода справа от слова *while* следует ввести условие, при истинности которого выполняется цикл..
- В поле ввода ниже слова *while* следует ввести тело цикла – одна или несколько строк программы (для введения нескольких строк используется оператор *Add Line*), выполнение которых нужно повторить несколько раз.

Рассмотрим использование цикла *while* для вычисления приближенного значения квадратного корня (методом касательных):

$$\mathrm{sqrt}(a,\varepsilon) := \left| \begin{array}{l} es \leftarrow 1 \\ \text{while } \left|es^2 - a\right| \ge \varepsilon \\ \quad es \leftarrow \frac{1}{2}\cdot\left(es + \frac{a}{es}\right) \\ es \end{array} \right.$$

$$\mathrm{sqrt}\left(37, 10^{-8}\right) = 6.08276253$$

Цикл с параметром (*for*).

Оператор *for* имеет три поля ввода :

- В поле ввода между словом *for* и знаком $\in$, следует указать имя переменной-счетчика.
- В поле ввода после знака $\in$ следует указать диапазон значений, которые будет принимать переменная-счетчик (вместо диапазона можно указать имя массива, из которого должны браться значения переменной-счетчика).
- В поле ввода под словом *for* следует ввести тело цикла.

Рассмотрим функцию, вычисляющую факториал.

$$\mathrm{fakt}(n) := \left| \begin{array}{l} f \leftarrow 1 \\ \text{for } i \in 1..n \\ \quad f \leftarrow f\cdot i \\ f \end{array} \right.$$

$\mathrm{fakt}(5) = ∎$

$\mathrm{fakt}(3.5) = ∎$

$\mathrm{fakt}(-4) = ∎$

Рекурсия.

Для того, чтобы сделать программу рекурсивной, нужно организовать вызов ею самой себя внутри программы. Например, рассмотрим рекурсивную функцию, вычисляющую факториал:

$$\mathrm{fakt}(n) := \left| \begin{array}{l} 1 \ \text{ if } \ n = 0 \\ (n\cdot \mathrm{fakt}(n-1)) \ \text{ otherwise} \end{array} \right.$$

$\mathrm{fakt}(5) = ∎$

$\mathrm{fakt}(3.5) = ∎$

$\mathrm{fakt}(-4) = ∎$

Эта функция, в отличие от предыдущей не может работать для нецелых и отрицательных чисел.

Обработка ошибок.

Система Maxima предоставляет пользователю возможность перенаправлять программу в случае возникновения ошибки (деление на 0, выход за пределы массива). Для этого существует оператор on error, который содержит 2 поля ввода (справа и слева):

- Справа вводится выражение, которое следует вычислить
- Слева вводится выражение, которое следует вычислить, если в правом выражении окажется ошибка.

Например, рекурсивная программа вычисления факториала выдает ошибку для вычисления факториала нецелых и отрицательных чисел. Изменим программу так, чтобы в таких случаях факториал был равен 0:

$$\mathrm{fakt}(n) := 0 \ \text{ on error } \left| \begin{array}{l} 1 \ \text{ if } \ n = 0 \\ n \cdot \mathrm{fakt}(n-1) \ \text{ otherwise} \end{array} \right.$$

$\mathrm{fakt}(5) = 120$

$\mathrm{fakt}(-4) = 0$

$\mathrm{fakt}(3.5) = 0$

Иногда возникает обратная ситуация: система Maxima не видит никакой стандартной ошибки, но необходимо, чтобы появлялась надпись об ошибке. Например, в нерекурсивной программе вычисления факториала выдается ответ для отрицательных значений, а хотелось бы, чтобы возникала ошибка. В таких случаях используется конструкция следующего вида:

error (“[текст ошибки]”) *if* [условие]

Например:

```
fakt(n) := | f ← 1
           | for i ∈ 1..n
           |   f ← f·i
           | error("vvedite n>0 ") if n ≤ 0
           | f  otherwise
```

$\mathrm{fakt}(5) = 120$

$\mathrm{fakt}(3.5) = 6$

$\mathrm{fakt}(-4) = \blacksquare$

vvedite n>0

Программы, составленные из нескольких операторов.

Maxima имеет возможность написания различных программ, содержащих несколько операторов, при этом операторы могут быть вложены друг в друга, если в какой-то оператор вложено несколько операторов, то их нужно объединять вертикальной линией (*Add Line*).

Рассмотрим программу, вычисляющую среднее арифметическое элементов произвольной матрицы.

$$A := \begin{pmatrix} 1 & 3 & 2 & 0 \\ 3 & 2 & 1 & 3 \\ 3 & 2 & 1 & 4 \end{pmatrix}$$

$$\mathrm{sum}(q) := \left| \begin{array}{l} s \leftarrow 0 \\ k \leftarrow 0 \\ \text{for } \ i \in 1..\,\mathrm{rows}(q) \\ \quad \text{for } \ j \in 1..\,\mathrm{cols}(q) \\ \qquad \left| \begin{array}{l} s \leftarrow s + q_{i,\,j} \\ k \leftarrow k + 1 \end{array} \right. \\ \text{return } \ \dfrac{s}{k} \end{array} \right.$$

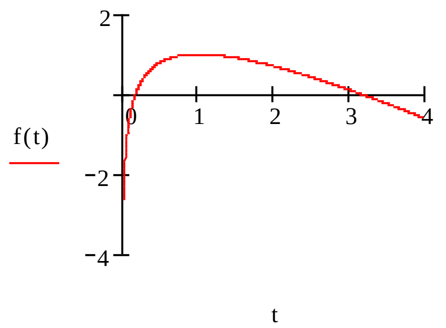
$$\mathrm{sum}(A) = \blacksquare$$

Занятие 7.
Процедуры решения уравнений, неравенств и поиска экстремальных значений функции.

Численный поиск корня уравнения.

Любой численный метод решения уравнения основан на уточнении какого-либо начального значения корня до заданной точности. Точность вычислений в Maxima задается встроенной переменной TOL и по умолчанию равна 0,001. Для задания начального значения корня удобно сначала построить график функции, задающей уравнение. Например, найдем корни уравнения $\ln x = x - 2$. Для задания начального значения x построим график функции $f(x) = \ln x - x + 2$.

$f(x) := \ln(x) - x + 2$



Для нахождения правого корня уравнения зададим точность TOL=0,000001 , начальное значение x=4 и используем функцию root(f(x),x) .

$x := 4$

$\mathrm{root}(f(x), x) = 3.146$

Можно локализовать корень не с помощью начального значения, а с помощью интервала, которому принадлежит корень, при этом границы интервала нужно задать как аргументы функции root. Например, найдем левый корень исходного уравнения как корень, принадлежащий интервалу [0.01, 1].

$\mathrm{root}(f(x), x, 0.01, 1) = 0.159$

Можно задать функцию r(x)=root(f(x),x), которая возвращает корень уравнения, полученный из начального приближения x. Например:

$r(x) := \mathrm{root}(f(x), x)$

$r(2) = 3.146$

$r(0.2) = 0.159$

Пользуясь этой функцией можно получить вектор корней уравнения:

$i := 0..2$

$x_i :=$

$$\begin{array}{|c|}\hline -1 \\ \hline 0.2 \\ \hline 4 \\ \hline \end{array} \qquad X_i := r(x_i) \qquad X_i = \begin{array}{|c|}\hline 3.146 \\ \hline 0.159 \\ \hline 3.146 \\ \hline \end{array}$$

Для получения комплексного корня начальное приближение следует задавать комплексным.

$f(x) := x^2 + 3$

$x := i$

$root(f(x), x) =$ ▮

Нахождение корней полиномов.

Для нахождения корней полинома в Maxima имеется встроенная функция polyroots(a), аргументом которой является вектор коэффициентов полинома $a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$. Например, для уравнения $x^3 + 2x^2 - 1 = 0$ вектор a имеет вид:

$i := 0..3$

$a_i :=$

$$\begin{array}{|c|}\hline -1 \\ \hline 0 \\ \hline 2 \\ \hline 1 \\ \hline \end{array} \qquad polyroots(a) = \begin{pmatrix} -1.618 \\ -1 \\ 0.618 \end{pmatrix}$$

Коэффициенты полинома и корни могут быть и комплексными. Например для уравнения $x^2 + 1 = 0$

$i := 0..2$

$a_i :=$

$$\begin{array}{|c|}\hline 1 \\ \hline 0 \\ \hline 1 \\ \hline \end{array} \qquad polyroots(a) = \begin{pmatrix} -i \\ i \end{pmatrix}$$

Для того, чтобы выделить из многочлена вектор коэффициентов можно воспользоваться символьными преобразованиями. Например решим уравнение $(x+1)(x^2-4)(x^2+x-2)=0$:

$$a := (x+1)\left(x^2-4\right)\left(x^2+x-2\right) \text{ coeffs}, x \rightarrow \begin{pmatrix} 8 \\ 4 \\ -10 \\ -5 \\ 2 \\ 1 \end{pmatrix}$$

$$polyroots(a) = \begin{pmatrix} -2 \\ -2 \\ -1 \\ 1 \\ 2 \end{pmatrix}$$

Нахождение корней уравнений аналитически (путем символьных преобразований).

Во многих случаях Maxima позволяет найти аналитическое решение. Для того, чтобы найти решение уравнения необходимо записать выражение и выделить в нем переменную (поставить указатель курсора возле переменной), и воспользоваться пунктом Solve for Variable (Переменная□Решение) из пункта меню Symbolic (Символика).

Можно также воспользоваться функцией root , поставив вместо знака «=» знак «□».

Численный поиск решения системы уравнений и неравенств.

Системы линейных и нелинейных уравнений и неравенств позволяет решать в Maxima блок *given* в сочетании с функцией *Find*.

После слова *given* записывается система уравнений и неравенств, подлежащих решению, при этом вместо знака «=» следует набирать Ctrl+=. Перед блоком given необходимо указывать начальные приближения для всех переменных, если нужно найти комплексный корень, то следует задавать комплексное начальное приближение. Признаком окончания системы служит функция *Find*, если надо найти точное решение или функция *Minerr*, если система не может быть решена точно, и требуется найти наилучшее приближение, обеспечивающее минимальную погрешность.

Функции *Minerr* и *Find* должны иметь столько же или меньше аргументов, сколько уравнений и неравенств содержит блок *given*. Если окажется, что блок содержит слишком мало уравнений или неравенств, то его можно дополнить тождествами или повторяющимися выражениями.

Например,

$x := 1 \qquad y := 1$

given

$$x^2 - y = 23$$

$$x^2 \cdot y = 50$$

$$\mathrm{Find}(x, y) = \begin{pmatrix} 5 \\ 2 \end{pmatrix}$$

Если необходимо найти решение при различных начальных приближениях, имеет смысл определить новую функцию. Например:

given

$$x^2 - y = 23$$

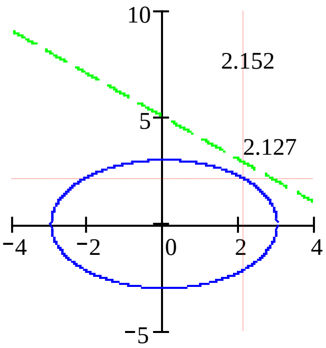
$$x^2 \cdot y = 50$$

$$f(x, y) := \mathrm{Find}(x, y)$$

$$f(1, 1) = \begin{pmatrix} 5 \\ 2 \end{pmatrix} \qquad f(-1, 1) = \begin{pmatrix} -5 \\ 2 \end{pmatrix}$$

$$f(i, 1) = \begin{pmatrix} 1.414i \\ -25 \end{pmatrix} \qquad f(-i, 1) = \begin{pmatrix} -1.414i \\ -25 \end{pmatrix}$$

Если система не имеет решения, но возникает необходимость найти значения переменных, при которых уравнения, входящие в систему удовлетворяются хотя бы приближенно, то используется функция miner. Например:



$x := 1 \qquad y := 1$

given

$$x^2 + y^2 = 9$$

$$y + x = 5$$

$$\mathrm{minerr}(x, y) = \begin{pmatrix} 2.152 \\ 2.127 \end{pmatrix}$$

Решение систем линейных уравнений.

Систему линейных уравнений можно решать численным методом, описанном в предыдущем пункте, но если определитель матрицы из коэффициентов при неизвестных в уравнениях системы будет равен 0 (система не имеет решения либо имеет множество решений), то численный метод не даст результата. Также численный метод может дать приближенный результат вместо точного. Поэтому для решения систем линейных уравнений можно воспользоваться методом Гаусса, матричным методом или формулами Крамера. Однородную же систему линейных уравнений можно решить только методом Гаусса. Рассмотрим решение систем линейных уравнений всеми этими способами:

а) численный метод

$x := -2 \quad y := 1 \quad z := 2$

Given

$2x - 4y + 3z = 1$

$x - 2y + 4z = 3$

$3x - y + 5z = 2$

$$\mathrm{Find}(x, y, z) = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

б) метод Гаусса

$2x - 4y + 3z = 1$

$x - 2y + 4z = 3$

$3x - y + 5z = 2$

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

$$\mathrm{rref}(A) = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \qquad C := \mathrm{rref}(A)$$

$$X := C^{\langle 4 \rangle}$$

$$X = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

в) матричный метод

$2x - 4y + 3z = 1$

$x - 2y + 4z = 3$

$3x - y + 5z = 2$

$$A := \begin{pmatrix} 2 & -4 & 3 \\ 1 & -2 & 4 \\ 3 & -1 & 5 \end{pmatrix} \qquad B := \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}$$

$$A^{-1} = \begin{pmatrix} 0.24 & -0.68 & 0.4 \\ -0.28 & -0.04 & 0.2 \\ -0.2 & 0.4 & 0 \end{pmatrix}$$

$$X := A^{-1} \cdot B$$

$$X = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

г) метод Крамера

$2x - 4y + 3z = 1$

$x - 2y + 4z = 3$

$3x - y + 5z = 2$

$$A := \begin{pmatrix} 2 & -4 & 3 \\ 1 & -2 & 4 \\ 3 & -1 & 5 \end{pmatrix} \qquad B := \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}$$

$$A1 := A \qquad A2 := A \qquad A3 := A$$

$$A1^{\langle 1 \rangle} := B \qquad A2^{\langle 2 \rangle} := B \qquad A3^{\langle 3 \rangle} := B$$

$$A1 = \begin{pmatrix} 1 & -4 & 3 \\ 3 & -2 & 4 \\ 2 & -1 & 5 \end{pmatrix} \qquad A2 = \begin{pmatrix} 2 & 1 & 3 \\ 1 & 3 & 4 \\ 3 & 2 & 5 \end{pmatrix} \qquad A3 = \begin{pmatrix} 2 & -4 & 1 \\ 1 & -2 & 3 \\ 3 & -1 & 2 \end{pmatrix}$$

$$X1 := \frac{|A1|}{|A|} \qquad X2 := \frac{|A2|}{|A|} \qquad X3 := \frac{|A3|}{|A|}$$

$$X1 = -1 \qquad X2 = 0 \qquad X3 = 1$$

Занятие 8.
Средства аналитических преобразований в Maxima.

При аналитических вычислениях результат получают в нечисловой форме в результате тождественных преобразований, среди которых более простыми есть раскрытия скобок.

С помощью символьного процессора Maxima можно решать инженерные задачи в аналитическом виде и проводить широкий спектр аналитических преобразований, таких как, упрощение выражений и алгебраические преобразования, алгебраические и матричные операции, основные действия математического анализа, и т.д.

Расписание алгебраического выражения - это математическое преобразование, которое переводит степени и произведения в более простые соотношения. При расписании тригонометрических выражений функции кратного аргумента превращаются в функции одинарного аргумента, и т.д.. Maxima разрешает упрощать логарифмические выражения, раскладывать на множители, приводить выражения к общему знаменателю, выносить множитель за скобки, раскладывать на элементарные дроби, выполнять подстановки и замены переменных.

Символьные вычисления можно выполнять в таких вариантах:

– с помощью команд меню;

– с помощью оператора символьного вывода, ключевых слов символьного процессора и обычных формул.

Для символьных вычислений с помощью команды предназначены главное меню Symbolic (Символика), которое объединяет математические операции. Для реализации второго подхода применяются все средства Maxima (например, Calculator, Evaluation, и т.п.).

Рассмотрим оба типа символьных вычислений на простом примере разложения на сомножители выражения sin (2x).

Первый способ (с помощью меню).

1. Введите выражение sin (2x).
2. Выделите его целиком .
3. Выберите в главном меню пункты Symbolics / Expand (Символика / Разложить).

После этого результат разложения выражения появится чуть ниже в виде еще одной строки

$$\sin(2x)$$

$$2\cdot\sin(x)\cdot\cos(x)$$

Символьные операции с помощью меню возможны лишь над каким-либо объектом (выражением, его частью или отдельной переменной). Для того чтобы правильно осуществить желаемое аналитическое преобразование, предварительно необходимо выделить тот объект, к которому оно будет относиться. Например, если в выражении sin(2x)+cos(2x) выделить только второе слагаемое, то разложиться только оно:

$$\sin(2x) + \cos(2x)$$

$$\sin(2x) + \left(2\cdot\cos(x)^2 - 1\right)$$

Второй способ символьных преобразований (с помощью оператора ->).

1. Введите выражение sin (2x).
2. Нажмите кнопку Expand (Разложить) на панели Symbolic (Символика).
3. Введите в местозаполнитель после появившегося ключевого слова expand имя переменной x, либо нажмите клавишу <Del>, чтобы просто удалить местозаполнитель.
4. Введите оператор символьного вывода (->)
5. Нажмите клавишу <Enter>, либо просто щелкните мышью за пределами выражения.

$$\sin(2x)\ \text{expand}, x\ \rightarrow 2\cdot\sin(x)\cdot\cos(x)$$

$$\sin(2x)\ \text{expand}\ \rightarrow 2\cdot\sin(x)\cdot\cos(x)$$

Если символьные вычисления осуществляются вторым способом, символьный процессор учитывает все формулы, предварительно введенные в документе . Например,

$x := 0$

$\sin(2x) + \pi$

$2\cdot\sin(x)\cdot\cos(x) + \pi$

$\sin(2x) + \pi$ expand $\rightarrow \pi$

Не всякое выражение поддается аналитическим преобразованиям. Если выражение не поддается аналитическим преобразованииям, то в качестве результата выводится само выражение :

$\cos(x)$ expand, x $\rightarrow \cos(x)$

С помощью меню Symbolic (Символика) можно выполнять такие операции:

Symbolic/Evaluate (Символика/Вычисление) символьное вычисление, в том числе с плавающей точкой (рис.1,а);

Symbolic/Simplify (Символика/Упрощение выражений)

упрощение выражения, символьный процессор Maxima стремится так преобразовать выражение, чтобы оно приобрело более простую форму. При этом используются различные арифметические формулы, приведение подобных слагаемых, тригонометрические тождества, пересчет обратных функций и др. Например,

$(x + 2y)\cdot z - z^2\cdot(x + 5y) + z$ simplify $\rightarrow z\cdot x + 2\cdot z\cdot y - z^2\cdot x - 5\cdot z^2\cdot y + z$

$x := 5 \qquad y := 3$

$(x + 2y)\cdot z - z^2\cdot(x + 5y) + z$ simplify $\rightarrow 12\cdot z - 20\cdot z^2$

$\sqrt{3} + x$ simplify $\rightarrow \sqrt{3} + 5$

$\sqrt{3.01} + x$ simplify $\rightarrow 6.7349351572897472412$

Symbolic/Expand (Символика/Разложение выражений) разложение выражений на элементарные;

Symbolic/Factor(Символика/Разложение на множители) разложение на множители;

Symbolic/Collect(Символика/Подобные) приведение подобных;

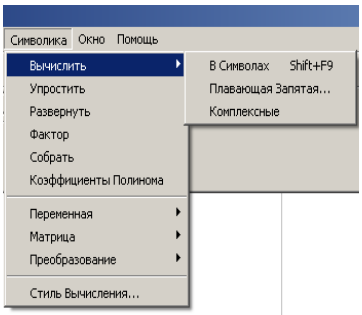
Symbolic/Polynomial Coefficients (Символика/Полиномиальные коэффициенты) вывод коэффициентов полиномов;

Symbolic/Variable(Символика/Переменная/...) решение уравнения; подстановка переменных; дифференцирование; интегрирование; разложение в ряды; разложение на элементарные дроби (рис.1,б));

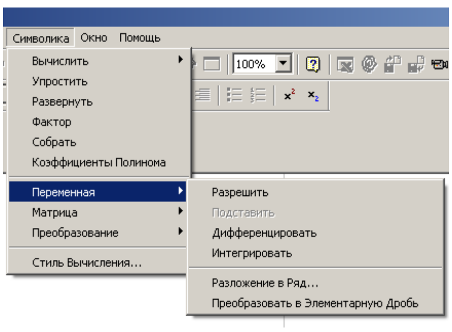
Symbolic/Matrix(Символика/Матрицы) действия с матрицами (рис.1,в);

Symboli/Transform(Символика/Интегральные преобразования) преобразование Фурье, Лапласа) (рис.1,г).

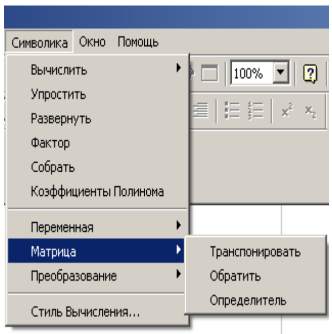
Последовательность выполнения вычислений можно задать с использованием Стиля Вычислений (рис.1, д).



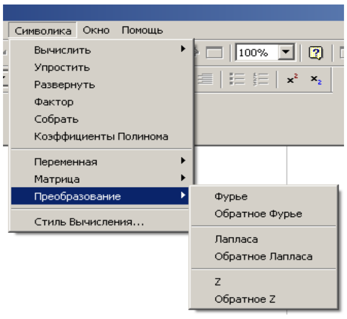
а)



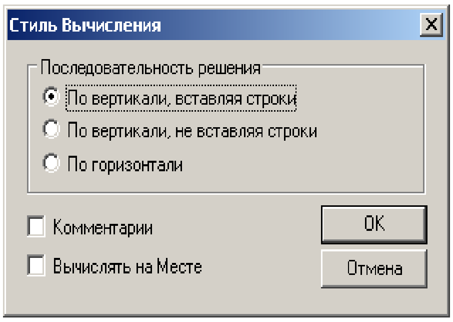
б)



в)



г)



д)

Рисунок 1 – Команды меню Symbolic

При проведении символьных вычислений можно использовать следующие символьные операторы:

1) ***float*** – указывает на то, что результат должен быть выведен в виде числа с плавающей запятой, после слова float должно быть указано количество знаков после запятой, которые должны быть выведены в результате; при символьных вычислениях с использованием оператора float, в отличие от обычных численных расчетов (=), выражение сначала вычисляется аналитически, поэтому максимальное количество знаков после запятой равно 250 (а при численных расчетах - 15). Например,

$$\sin\left(\frac{\pi}{4}\right) = 0.707$$

$$\sin\left(\frac{\pi}{4}\right) \rightarrow \frac{1}{2}\cdot\sqrt{2}$$

$$\sin\left(\frac{\pi}{4}\right) \text{ float}, 25 \;\rightarrow .70710678118654752440$$

2) ***complex*** – выводит комплексное число в виде a+bi

3) ***assume*** – наложение ограничений на параметры выражения

4) ***solve*** – решение уравнения или системы уравнения

5) ***simplify*** – упрощение выражения, символьный процессор Maxima стремится так преобразовать выражение, чтобы оно приобрело более простую форму. При этом используются различные арифметические формулы, приведение подобных слагаемых, тригонометрические тождества, пересчет обратных функций и др. Например,

$$(x + 2y)\cdot z - z^2\cdot(x + 5y) + z \text{ simplify } \rightarrow z\cdot x + 2\cdot z\cdot y - z^2\cdot x - 5\cdot z^2\cdot y + z$$

$$x := 5 \qquad y := 3$$

$$(x + 2y)\cdot z - z^2\cdot(x + 5y) + z \text{ simplify } \rightarrow 12\cdot z - 20\cdot z^2$$

$$\sqrt{3} + x \text{ simplify } \rightarrow \sqrt{3} + 5$$

$$\sqrt{3.01} + x \text{ simplify } \rightarrow 6.7349351572897472412$$

6) ***substitute*** – подстановка выражения вместо переменной. Например,

$$a\cdot x^2 + b\cdot x + c \text{ substitute}, x = 5 \;\rightarrow 25\cdot a + 5\cdot b + c$$

$$\sin\left(k\cdot x^2 + b\cdot x\right) \text{ substitute}, k = a\cdot x^2, b = x \;\rightarrow \sin\left(a\cdot x^4 + x^2\right)$$

7) ***factor*** – в зависимости от введенного выражения, данный оператор выполняет одно из следующих действий: сворачивает полином в произведение полиномов, раскладывает целое число на простые множители или приводит дроби к общему знаменателю. Например,

$$x^4 - 16 \text{ factor } \rightarrow (x - 2)\cdot(x + 2)\cdot\left(x^2 + 4\right)$$

$$28 \text{ factor } \rightarrow 2^2\cdot 7$$

8) ***expand*** - перация символьного разложения, или расширения, выражений. В ходе разложения

раскрываются все суммы и произведения, а сложные тригонометрические зависимости разлагаются с помощью тригонометрических тождеств.

9) ***coeffs*** – вычисление полиномиальных коэффициентов

$$(x+2y)\cdot z - z^2\cdot y\cdot(x+5y) + z \text{ coeffs}, x \rightarrow \begin{pmatrix} 2\cdot z\cdot y - 5\cdot z^2\cdot y^2 + z \\ z - z^2\cdot y \end{pmatrix}$$

$$(x+2y)\cdot z - z^2\cdot y\cdot(x+5y) + z \text{ coeffs}, z \rightarrow \begin{pmatrix} 0 \\ x+2\cdot y+1 \\ -y\cdot x - 5\cdot y^2 \end{pmatrix}$$

$$(x-4)\cdot(x-7)\cdot x + 99 \text{ coeffs}, x \rightarrow \begin{pmatrix} 99 \\ 28 \\ -11 \\ 1 \end{pmatrix}$$

10) ***collect*** – приведение выражения к полиному по заданной переменной (приведение подобных слагаемых):

$$(x+2y)\cdot z - z^2\cdot y\cdot(x+5y) + z \text{ collect}, x \rightarrow \left(z - z^2\cdot y\right)\cdot x + 2\cdot z\cdot y - 5\cdot z^2\cdot y^2 + z$$

$$(x+2y)\cdot z - z^2\cdot y\cdot(x+5y) + z \text{ collect}, y \rightarrow -5\cdot z^2\cdot y^2 + \left(2\cdot z - z^2\cdot x\right)\cdot y + z\cdot x + z$$

$$(x+2y)\cdot z - z^2\cdot y\cdot(x+5y) + z \text{ collect}, y, x \rightarrow -5\cdot z^2\cdot y^2 + \left(2\cdot z - z^2\cdot x\right)\cdot y + z\cdot x + z$$

$$(x+2y)\cdot z - z^2\cdot y\cdot(x+5y) + z \text{ collect}, x, y \rightarrow \left(z - z^2\cdot y\right)\cdot x + 2\cdot z\cdot y - 5\cdot z^2\cdot y^2 + z$$

11)***series*** – разложение функции в степенной ряд Тейлора, нужно указать имя переменной, по которой проводится разложение и порядок аппроксимации (количество степенных слагаемых)

$$\sin(x) \text{ series}, x, 10 \rightarrow x - \frac{1}{6}\cdot x^3 + \frac{1}{120}\cdot x^5 - \frac{1}{5040}\cdot x^7 + \frac{1}{362880}\cdot x^9$$

12) ***parfrac*** – разложение дроби на простейшие:

$$\frac{11x^2 + 9x + 1}{x^2 - 3x + 2} \text{ convert}, \text{parfrac}, x \rightarrow 11 - \frac{21}{(x-1)} + \frac{63}{(x-2)}$$

13) ***fourier, invfourier*** – прямое и обратное преобразование Фурье

14***) laplace, invlaplace*** – прямое и обратное преобразование Лапласа

15) ***ztrans, invztrans*** – прямое и обратное Z-преобразование.

Символьные операции с матрицами.

Для того, чтобы производить преобразования матриц в символьном виде используется оператор символьного вычисления →. Например:

$$A(a,x) := \begin{pmatrix} 1 & 1-x & a \\ -2x & x & 2x \\ a & 1-x & 1 \end{pmatrix} \qquad v(x) := \begin{pmatrix} 0 \\ x \\ 0 \end{pmatrix}$$

$$|A(a,x)| \rightarrow x - a^2 \cdot x \qquad A(a,x)\cdot v(x) \rightarrow \begin{bmatrix} x\cdot(1-x) \\ x^2 \\ x\cdot(1-x) \end{bmatrix}$$

$$\mathrm{lsolve}(A(a,x), v(x)) \rightarrow \begin{bmatrix} \frac{(x-1)}{(a+1)} \\ 1 \\ \frac{(x-1)}{(a+1)} \end{bmatrix} \qquad \mathrm{eigenvals}(A(a,x)) \rightarrow \begin{pmatrix} -a+1 \\ a+1 \\ x \end{pmatrix}$$

Решение систем уравнений в символьном виде.

Во многих случаях решение системы уравнений может быть найдено не только численно, но и аналитически. Для этого также используется блок given и функция Find, но вместо знака равенства после функции следует поставить знак символического преобразования $\rightarrow$.

Например:

given

$$x^2 + x\cdot y + y^2 = 13$$

$$x - y = 4$$

$$\mathrm{Find}(x,y) \rightarrow \begin{pmatrix} 2+\sqrt{3} & 2-\sqrt{3} \\ -2+\sqrt{3} & -2-\sqrt{3} \end{pmatrix}$$

Решение записано в вид матрицы, каждый столбец которой является решением.

Решить систему уравнений в символьном виде можно также с помощью оператора solve. Для этого систему уравнений записывают в виде матрицы, состоящей из 1 столбца и нужного количества строк. Например:

$$\begin{pmatrix} x^2 + x\cdot y + y^2 = 13 \\ x - y = 4 \end{pmatrix} \mathrm{solve}, x, y \rightarrow \begin{pmatrix} 2+\sqrt{3} & -2+\sqrt{3} \\ 2-\sqrt{3} & -2-\sqrt{3} \end{pmatrix}$$

В данном случае решением является каждая строка матрицы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК.

Основная литература

1. Аладьев, В.З. MAPLE 6:Решение математических, статистических и инженерно-физических задач / В.З. Аладьев, М.А. Богдявичюс.– М.: Лаборатория Базовых Знаний, 2001.– 824с.
2. Дьяконов, В.П. Maxima 11/12/13 в математике: справочник / В.П.Дьяконов.– М.: Горячая линия-Телеком, 2007.– 958с.
3. Московский, А.В. Издательская система LATEX 2ε : учеб. пособие для вузов / А.В. Московский, Ю.В. Московская.– Тула : Изд-во ТулГУ, 2008 .– 172 с.
4. Юров, В.И. Assembler : учебное пособие для вузов / В.И.Юров.– 2-е изд. – М.[и др.] : Питер, 2006 .– 637с.
5. Шилдт, Schildt G. Искусство программирования на Java / Г.Шилдт, Д.Холмс;пер.с англ.и ред. Г.В. Галисеева .– М.и др. : Вильямс, 2005 .– 331с.

Дополнительная литература

1. Дьяконов, В.П. MAPLE 9.5/10 в математике, физике и образовании / В.П.Дьяконов .– М.: СОЛОН-Пресс, 2006.– 720с.
2. Бидасюк, Ю.М. Mathsoft Maxima 12: самоучитель / Ю.М. Бидасюк .- М.; СПб.; Киев: Диалектика, 2006 .– 224с.
3. Гуссенс, М. Путеводитель по пакету LATEX и его Web-приложениям : Справочник / М.Гуссенс,С.Ратц;Пер.с англ.:Ю.В.Тюменцева,А.В.Чернышева под ред.Б.В.Тоботраса .— М. : Мир, 2001 .— 604с..
4. Зубков, С.В. Assembler для DOS, Windows и Unix / С.В.Зубков .– 3-е изд.,стер. – М. : ДМК, 2006 .– 608с.
5. Вязовик, Н.А. Программирование на Java : Курс лекций для вузов / Н.А.Вязовик .– М., 2003 .– 592с.

Периодические издания

1. Журнал «PC Magazine, Персональный компьютер сегодня».– М.: ЗАО "СК Прессs"

Программное обеспечение и Интернет-ресурсы

1. Львовский, С.М. Работа в системе LaTeX Дистанционный учебный курс. Интернет-университет информационных технологий – INTUIT.ru, 2009.
2. Вязовик, Н.А. Программирование на Java. Дистанционный учебный курс. Интернет-университет информационных технологий – INTUIT.ru, 2009.