Maxima

Пережогин А.С.

Создан: 19 сентября 2012 г. Изменен: 16 октября 2012 г.

Содержание

1 Введение в Maxima

Система компьютерной алгебры Maxima реализована для различных платформ. Доступны две основные реализации: для Windows и для Linux. Изначально Maxima предоставляла возможность работы через командную строку, но в процессе развития появились пользовательские интерфейсы. В данном пособии мы будем работать с пользовательским интерфейсом wxMaxima.

Исторически система Maxima появилась благодаря Вильяму Шелтеру, который в 1982 сделал модификацию исходных кодов от системы Macsyma и продолжал поддержку программы на правах академической лицензии до своей смерти в июле 2001.

В пособии мы рассматриваем реализацию Maxima под операционную систему Linux. Пользователи Windows смогу без особых трудностей применять данное пособие для дистрибутива под своей операционной системой.

2 Запуск Maxima. Базовые принципы работы

Запуск Maxima осуществляется в консоли Linux командой

```
localhost$ maxima
```

Базовые операции сложение, умножение, вычитание и деление представлены с помощью: +,*.- и /.

```
(%i1) x^2+x+1;
(%o1) x^2+x+1;
```

Строка ввода команды может заканчиваться символом ; (вывести на экран) или $ (блокировка вывода на экран).

По умолчанию каждое выражение сохраняется в служебных переменных $\%i1$, $\%o1$. Так же с системе предусмотрено использование пользовательских переменных при объявлении выражений. Присвоим в переменную func1 выражение $x^2 + x + 1$.

```
(%i1) func1:x^2+x+1;
(%o1) x^2+x+1;

(%i2) func1;
(%o2) x^2+x+1
```

Пример решения квадратного уравнения. Функция solve аналитически находит корни уравнения.

```
(%i1) func1:x^2-2*x-8 ;
(%o1) x^2-2*x-8;

(%i2) solve(func1, x);
(%o2) [x=-2,x=4]
```

Значения возвращаются в виде списка из двух элементов.

2.1 Отложенные вычисления

Maxima допускает 2 возможности работы с выражениями: прямые вычисления и отложенные вычисления. В первом случае, система выполняет все допустимы действия над выражение, а во втором случае, позволяет объявить выражение, но вычисления отложить до специальной команды пользователя по вычислению выражения. При это допускается как частичное вычисление выражения, так и полное вычисление. Функции, которые допускают отложенные вычисления помечаются символом ' перед именем функции.

Сравнить выполнение функции дифференцирования $diff$.

```
(%i1) diff(1/sqrt(1+x^3),x);
(%i2) diff(1/sqrt(1+x^3),x);
```

2.2 Функция ev – окружение для вычислений

Все математические операции выполняются в пределах своих окружений. Maxima имеет в своем арсенале очень удобную команду ev, которая дает возможность управления параметрами вычислений в пределах сложных выражений.

Например, зададим в общем виде многочлен второй степени $a * x^2 + b * x + c$ и найдем его корни при фиксированном значении параметров a, b, c

```
(%i1) ev(solve(a*x^2+b*x+c,x), a=1, b = 2, c = 1);
```

В самом общем виде функция ev имеет 2 параметра: первый параметр – выражение, второй – ключи функций, которые определяют какие вычисления необходимо производить.

```
(%i1) ev(exp(a), a = 3/2);
(%o1) e^(3/2)

(%i1) ev(exp(a), a = 3/2, float);
(%o1) 4.481689070338065
```

2.3 Очищение памяти

В процессе вычислений требуется очистить значение какой-либо переменной. Простейший способ – использовать функцию kill(). В качестве аргумента она принимает имя переменной, значение которой требует очистить.

```
(%i1) A:x+5;
(%o1) x+5

(%i2) kill(A);
(%o2) DONE

(%i3) A;
(%o3) A
```

Очистить все переменные можно с помощью команды $kill(all)$.

2.4 Базовые операторы

Оператором является тот символ, который выполняет специфическую операцию. Существует большое количество операторов в Maxima. Рассмотрим основные из них.

2.4.1 Оператор присвоения

: – оператор присвоения, который записывает значение выражения в указанную переменную.

```
(%i1) A:x+5;
(%i2) A;
```

2.4.2 Оператор определения функции

В математике используется оператор =, чтобы указать зависимость функции от аргумента следующим образом $y(x) = \sin(x)$. В Maxima используется оператор := – оператор определения функции.

Пример объявления функции $y(x) = x^2$.

```
(%i1) y(x) := x^2;
(%o2) y(x) := x^2;

(%i2) y(2);
(%o2) 4
```

2.4.3 Оператор изменения указателя

:: – оператор изменения ссылки на значение.

2.4.4 Оператор ! – факториал

! – оператор вычисления факториала числа.

```
(%i1) 5!
(%o2) 120
```

2.4.5 Оператор !!

!! – оператор вычисления произведения четных или нечетных чисел в зависимости от четного или нечетного аргумента.

```
(%i1) 5!!
(%o2) 15
```

2.4.6 Оператор *sqrt*

sqrt() – оператор извлечения квадратного корня

2.5 Дифференцирование и интегрирование выражений

2.5.1 Тригонометрические функции

В Maxima доступен широкий набор тригонометрических функций:
sin(x) – синус, cos(x) – косинус, tan(x) – тангенс, cot(x) – котангенс.

2.5.2 Дифференцирование выражений

Функция дифференцирования выражения имеет вид diff(выражение, переменная дифференцирования, порядок дифференцирования)

```
(%i1) diff(sin(x)*cos(x),x)
(%i2) diff(sin(x)*cos(x),x,2)
```

Чтобы каждый раз не указывать зависимость функции от переменных можно использовать зарезервированную команду для объявления зависимости.

```
(%i1) DEPENDS([U],[r,t],[r,t],[x,y])
(%i2) diff(U,x)
```

Вычисление смешанной производной выполняется командой

```
(%i3) diff(U,x,1,y,1)
```

2.5.3 Интегрирование выражений

Функция интегрирования выражений имеет формат integrate(выражение, переменная интегрирования)

```
(%2i) integrate(sin(x),x)
(%3i) integrate(sqrt(x+1),x)
(%4i) integrate(x^2+2x-c,x)
```

Возможны ситуации, когда результат интегрирования зависит от значения параметров интегрирования.

```
(%i31) integrate(a*x^n,x);
Is n+1 zero or nonzero?
nonzero;
```

Можно заранее предопределить значение параметра a

```
(%i5) assume(n+1>0);
(%i6) integrate(a*x^n,x);
```

2.5.4 Решение дифференциальных уравнений

Решение обыкновенных дифференциальных уравнений выполняется функцией ode2.

Зададим зависимость $y(x)$

```
(%i2) depends(y,x);
```

Решим уравнение $x^2\frac{dy(x)}{dx} + 3xy = sin(x)/x$.

```
(%i2) ode2(x^2*diff(y,x) + 3*x*y = sin(x)/x, y,x )
```

Подстановка начальных условий выполняется с помощью функций $ic1$, $ic2$,, для граничных условий $bc2$.

Пример задания начальных условий для уравнения второго порядка

```
(%i8) eqn2: diff(y,x,2) + y = 4*x;
(%i9) soln2: ode2(eqn2, y, x);
(%i10) ic2(soln2, x=0, y=1, diff(y,x)=3);
```

2.6 Контрольные задания

- Продифференцировать функцию $\sin(x) + x^3 + x^2 + x$
- Решить уравнение $x^2 + 2 * x + 1$
- Решить дифференциальное уравнение $diff(y, x) + x * y = sin(x)$.

Список литературы

[1] Maxima book – Documentation

[2] Maxima homepage http://maxima.sourceforge.net/