

Справочник по формулам Maxima, используемым при работе с уравнениями

1. Ввод данных

Для задания уравнения в Maxima используйте следующий синтаксис:

eq: выражение;

Пример:

eq: $x^2 - 4 = 0$;

2. Решение уравнений относительно переменной "x"

Для решения уравнения относительно переменной x применяется функция *solve*:

solve(уравнение, x);

Пример:

solve($x^2 - 4 = 0$, x);

Результат: [$x = -2$, $x = 2$]

3. Решение уравнений относительно произвольной переменной

Если необходимо решить уравнение относительно другой переменной, укажите её во втором аргументе функции *solve*:

solve(уравнение, переменная);

Пример:

solve($x + y - 5 = 0$, y);

Результат: [$y = 5 - x$]

4. Использование команды подстановки

Для подстановки значений в выражения используется функция *subst*:

subst(значение, выражение);

Пример:

```
subst(x = 2, x^2 + 3*x + 1);
```

Результат: 11

5. Нахождение приближенных решений

Для вычисления численных решений применяется функция *find_root*:

```
find_root(уравнение, переменная, начальное_значение, конечное_значение);
```

Пример:

```
find_root(x^2 - 2, x, 1, 2);
```

Результат: 1.41421356237

6. Решение систем линейных алгебраических уравнений

Для решения систем линейных уравнений используется функция *linsolve*:

```
linsolve([уравнение1, уравнение2, ...], [переменная1, переменная2, ...]);
```

Пример:

```
linsolve([x + y = 5, x - y = 1], [x, y]);
```

Результат: [x = 3, y = 2]

7. Проверка найденных решений

Для проверки решений можно использовать функцию *ev*:

```
ev(выражение, подстановки);
```

Пример:

```
ev(x^2 - 4, x = 2);
```

Результат: 0

8. Решение дифференциальных уравнений

Для решения обыкновенных дифференциальных уравнений (ОДУ) в Maxima используется функция *ode2*:

```
ode2(уравнение, зависимая_переменная, независимая_переменная);
```

Пример:

```
ode2(diff(y, x) = x*y, y, x);
```

Результат: $y = c * e^{(x^2/2)}$

Для решения уравнений второго порядка с начальными условиями используется функция `ic2`:

```
ic2(решение_уравнения, независимая_переменная, значение_независимой_переменной,  
значение_зависимой_переменной, значение_производной);
```

Пример:

```
sol: ode2(diff(y, x, 2) - y = 0, y, x);  
ic2(sol, x, 0, 1, 0);
```

Результат: $y = \cosh(x)$

9. Нахождение корней многочленов

Для нахождения корней многочленов используется функция `allroots`, которая вычисляет все корни численно:

```
allroots(многочлен);
```

Пример:

```
allroots(x^3 - 2*x + 1);
```

Для нахождения рациональных корней можно использовать функцию `solve`:

```
solve(многочлен = 0, переменная);
```

Пример:

```
solve(x^3 - 2*x + 1 = 0, x);
```