Занятия V и VI

Решение уравнений и неравенств

Основная функция solve

Для решения линейных и нелинейных уравнений и неравенств в аналитическом виде используется достаточно универсальная и гибкая функция solve(eqn, var) или solve({eqn1,eqn2,...},{var1,var2,...}), где eqn — уравнение, содержащее функцию ряда переменных, var — переменная, по которой ищется решение. Если при записи eqn не используются знак равенства или знаки отношения, считается, что solve ищет корни уравнения eqn=0.
Характер решений можно изменить с помощью глобальных переменных:

- _SolutionsMayBeLost — при значении true дает решение, которое при обычном применении функции solve возвращает значения NULL;
- _MaxSols — задает максимальное число решений;
- _EnvAllSolutions — при значении true задает выдачу всех решений.

В решениях могут встречаться следующие обозначения:

- _NN — указывает на неотрицательные решения;
- _B — указывает на решения в бинарной форме;
- _Z — указывает на то, что решение содержит целые числа;
- %N — при текстовом формате вывода задает общие члены решения и обеспечивает более компактную форму его представления.

Функция solve старается дать решение в аналитическом виде. Это не означает, что ее нельзя использовать для получения корней уравнений в численном виде. Просто для этого придется использовать функции evalf или convert. Если результат решения представлен через функцию RootOf, то зачастую можно получить все корни с помощью функции allvalues.

Решение одиночных нелинейных уравнений

Решение одиночных нелинейных уравнений вида $f(x) = 0$ легко обеспечивается функцией solve(f(x),x).

ПРИМЕРЫ

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

$$1, -\frac{1}{2}+\frac{1}{2}\sqrt{5}, -\frac{1}{2}-\frac{1}{2}\sqrt{5}$$

```
> solve(x^(3/2)=3,x);
```

$$3^{(2/3)}$$

```
> evalf(%);
```

2.080083823

```
> solve(sqrt(ln(x))=2,x);
```

$$e^4$$

```
> evalf(%);
```

54.59815003

Часто бывает удобно представлять уравнение и его решение в виде отдельных объектов, отождествленных с определенной переменной:

```
> eq:=(2*x^2+x+3=0);
```

$$eq := 2\,x^2 + x + 3 = 0$$

```
> s:=[solve(eq,x)];
```

$$s := \left[-\frac{1}{4} + \frac{1}{4} I \sqrt{23},\ -\frac{1}{4} - \frac{1}{4} I \sqrt{23} \right]$$

В частности, это позволяет легко проверить решение (даже если оно не одно, как в приведенном примере) подстановкой (subs):

```
> subs(x=s[1],eq);
```

$$2\left(-\frac{1}{4} + \frac{1}{4} I \sqrt{23} \right)^2 + \frac{11}{4} + \frac{1}{4} I \sqrt{23} = 0$$

```
> subs(x=s[2],eq);
```

$$2\left(-\frac{1}{4} - \frac{1}{4} I \sqrt{23} \right)^2 + \frac{11}{4} - \frac{1}{4} I \sqrt{23} = 0$$

```
> evalf(%);
```

$$0. + 0.\,I = 0.$$

Сводящиеся к одному уравнению равенства вида $f1(x) = f2(x)$ также решаются функцией solve(f1(x)=f2(x),x):

```
> solve(x^4=-x-1,x);
```

$$\mathrm{RootOf}(\_Z^4 + \_Z + 1, index = 1),\ \mathrm{RootOf}(\_Z^4 + \_Z + 1, index = 2),$$
$$\mathrm{RootOf}(\_Z^4 + \_Z + 1, index = 3),\ \mathrm{RootOf}(\_Z^4 + \_Z + 1, index = 4)$$

```
> evalf(%);
```

$$.7271360845 + .9340992895\,I,\ -.7271360845 + .4300142883\,I,$$
$$-.7271360845 - .4300142883\,I,\ .7271360845 - .9340992895\,I$$

```
> solve({exp(x)=sin(x)},x);
```

$$\{x = \mathrm{RootOf}(\_Z - \ln(\sin(\_Z)))\}$$

```
> evalf(%);
```

$$\{x = .3627020561 - 1.133745919\,I\}$$

```
> solve(x^4=2*x,x);
```

$$0,\ 2^{(1/3)},\ -\frac{1}{2} 2^{(1/3)} + \frac{1}{2} I \sqrt{3}\, 2^{(1/3)},\ -\frac{1}{2} 2^{(1/3)} - \frac{1}{2} I \sqrt{3}\, 2^{(1/3)}$$

```
> evalf(%);
```

$$0.,\ 1.259921050,\ -.6299605250 + 1.091123636\,I,\ -.6299605250 - 1.091123636\,I$$

Функция evalf позволяет получить решения, выраженные через функцию RootOf, в явном виде.

Решение тригонометрических уравнений

Функция solve может использоваться для решения тригонометрических уравнений:

```
> solve(sin(x)=.2,x);
```

.2013579208

```
> solve(sin(x)-1/2,x);
```

$$\frac{1}{6}\pi$$

```
> solve(cos(x)=.5,x);
```

1.047197551

Обратите внимание, что найдено только главное решение. Для нахождения всех периодических решений нужно выполнить команду:

```
> _EnvAllSolutions:=true:
```

_EnvAllSolutions := true

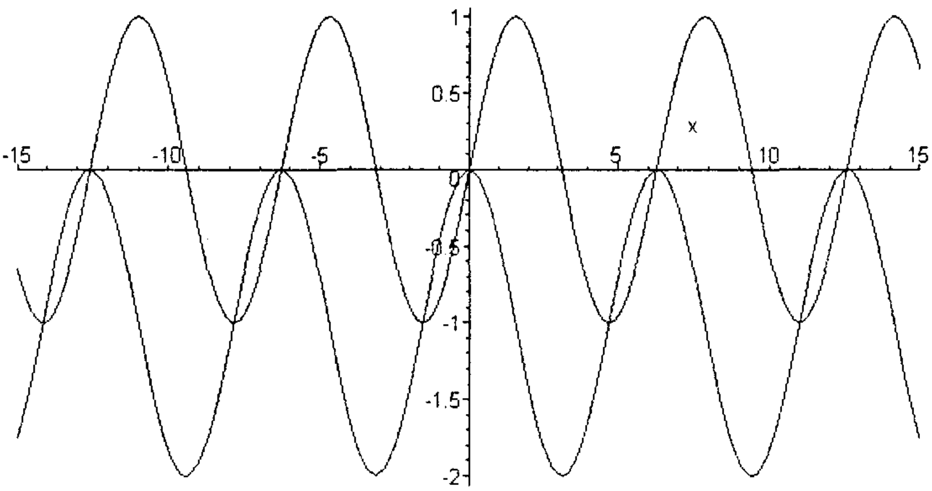
Указанная в ней системная переменная отвечает за поиск всех периодических решений, когда ее значение равно true, и дает поиск только главных решений при значении false, принятом по умолчанию. Так что теперь можно получить следующее:

```
> solve(sin(x)=1/2,x);
```

$$\frac{1}{6}\pi+\frac{2}{3}\pi\_B1\sim+2\pi\_Z1\sim$$

Решение тригонометрического уравнения, имеющего периодические решения

```
> restart;f1:=sin(x):f2:=cos(x)-1:
> plot([f1,f2],x=-15..15,color=black);
```




```
> solve(f1=f2,x);
```

$$-\frac{1}{2}\pi, 0$$

```
> evalf(%);
```

-1.570796327, 0.

```
> _EnvAllSolutions:=true:solve(f1=f2,x);
```

$$-\frac{1}{2}\pi+2\pi\_Z1, 2\pi\_Z2$$

В решениях встречаются переменные _B1~ и _Z1~, означающие ряд натуральных чисел. Благодаря этому через них можно представить периодически повторяющиеся решения.

Примеры решения уравнений с обратными тригонометрическими функциями показаны ниже:

```
> eqns := 2*arcsin(x) - arccos(5*x);
```

$$eqns := 2\arcsin(x) - \arccos(5\,x)$$

```
> solve( eqns, {x} );
```

$$\left\{ x = \frac{1}{4}\frac{-5+\sqrt{33}}{\sqrt{\left(-\frac{5}{4}+\frac{1}{4}\sqrt{33}\right)^2 - \frac{21}{8} + \frac{5}{8}\sqrt{33}}} \right\}$$

```
> eqns := arccos(x) - arctan(x/2);
```

$$eqns := \arccos(x) - \arctan\left(\frac{1}{2}x\right)$$

```
> solve( eqns, {x} );
```

$$\left\{ x = \frac{\sqrt{-2+2\sqrt{2}}}{\sqrt{(\sqrt{2}-1)^2 - 2 + 2\sqrt{2}}} \right\}$$

Решение систем линейных уравнений

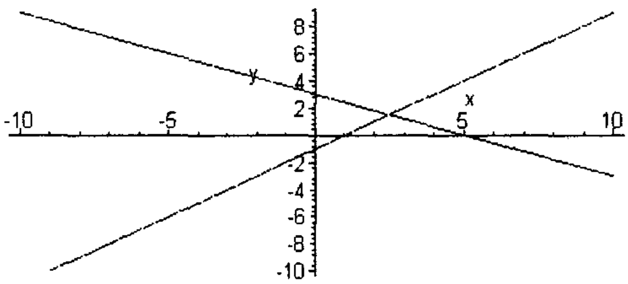
Для решения систем линейных уравнений созданы мощные матричные методы, которые будут описаны отдельно. Однако функция solve также может с успехом решать системы линейных уравнений. Такое решение в силу простоты записи функции может быть предпочтительным.

Примеры решения системы из двух линейных уравнений с графической иллюстрацией

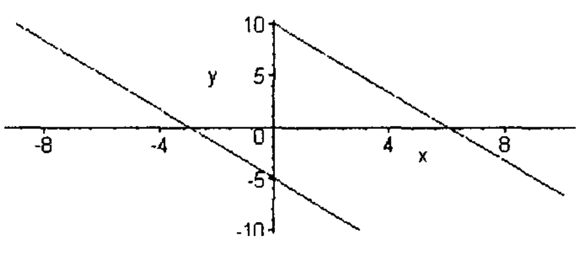
```
> restart; with(plots):
Warning, the name changecoords has been redefined

>  sys :=  { 3*x + 5*y = 15   ,    y = x - 1 }:
>  solve( sys, {x,y} );
```

$$\{y = \frac{3}{2}, x = \frac{5}{2}\}$$

```
> implicitplot(sys,x = -10..10,y = -10..10,color = black);
```




```
> sys:= { 5*x + 3*y = 30, 10*x + 6*y = -30}:
> solve( sys, {x,y} );
> implicitplot(sys,x = -10..10,y = -10..10,color = black);
```

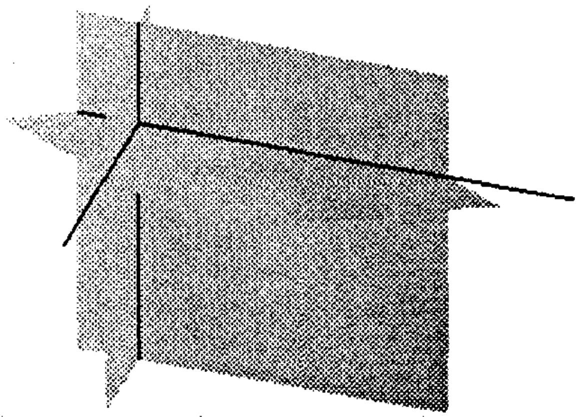


Пример решения системы из трех линейных уравнений с графической иллюстрацией решения

```
> restart; with(plots):
Warning, the name changecoords has been redefined

>  sys :=  { z = 4,  x+y= 10, x-y = 5  }:
> solve( sys, {x,y,z} );
```

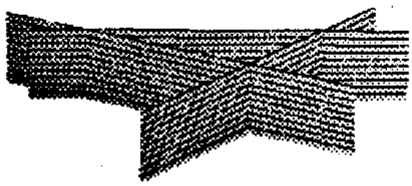
$$\{z = 4, y = \frac{5}{2}, x = \frac{15}{2}\}$$

```
> display(implicitplot3d(sys, x = -10..10, y = -10..10, z=-10..10, shading =
  xyz,style=patchnogrid,  orientation = [158,70]), spacecurve({[t,10-t,4],
  [t,t-5,4], [15/2,5/2,t]},t= -10..10, color = black, thickness = 2));
```



Графическая иллюстрация особых случаев решения системы из трех линейных уравнений

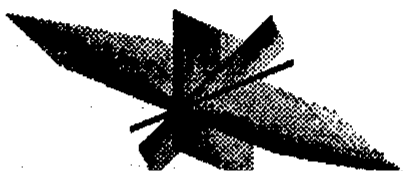
```
> sys :=  { 2*x - 3*y = 10,  2*x+y= 7,  y = 2*x + 4  }:
> solve( sys, {x,y,z} );
> implicitplot3d( sys, x = -10..10, y = -10..10, z=-10..10,
    style= patchcontour, orientation =[-25,30]);
```



```
> sys :=  { x + y + z = 1,  x + 3*y - z = 2, 2*x + 4*y = 3 }:
> solve( sys, {x,y,z} );
```

$$\{x = \frac{1}{2} - 2z, z = z, y = -\frac{1}{2} + z\}$$

```
>  display(implicitplot3d( sys, x = -10..10, y = -10..10, z=-10..10,
   orientation = [-30,110], style = patchnogrid), spacecurve([1/2 - 2*t,
   1/2 + t,t],t = -10..10, color = black, thickness = 3));
```



Следующий пример показывает решение системы из четырех линейных уравнений:

```
> sys := { 4*x1 + 7*x2 - x3 + 3*x4 = 11,
              -2*x1 + 2*x2 - 6*x3 + x4 = 4,
              x1 - 3*x2 + 4*x3 - x4 = -3,
              3*x1 - 5*x2 - 7*x3 + 5*x4 = 8 }:
> solve( sys, {x1, x2, x3, x4 } );
```

$$\{x2=\frac{8}{19},x3=\frac{-81}{19},x1=\frac{135}{19},x4=\frac{-156}{19}\}$$

Эта система имеет решение, но его простая графическая иллюстрация уже невозможна.

Случай решения неполной системы уравнений (уравнений — 3, а неизвестных — 4) иллюстрирует следующий пример:

```
> sys := { x1 + 2*x2 + 3*x3 + 4*x4 = 51,
              x1 - 3*x2 + 4*x3 +  x4  = 32,
              x1 + 2*x2 - 6*x3 +  x4  = -23   }:
> solve( sys, {x1, x2, x3, x4 } );
```

$$\{x2=\frac{2}{5}x1-\frac{11}{15},x4=-\frac{3}{5}x1+\frac{139}{15},x3=\frac{1}{5}x1+\frac{77}{15},x1=x1\}$$

Решение систем нелинейных и трансцендентных уравнений

Функция solve может использоваться для решения систем нелинейных и трансцендентных уравнений. Для этого система уравнений и перечень неизвестных задаются в виде множеств.

```
> restart;
> solve({x*y=a,x+y=b},{x,y});
```

$$\{y=\mathrm{RootOf}(\_Z^2-\_Z\,b+a),x=-\mathrm{RootOf}(\_Z^2-\_Z\,b+a)+b\}$$

```
> allvalues(%);
```

$$\{y=\frac{1}{2}b+\frac{1}{2}\sqrt{b^2-4\,a},x=\frac{1}{2}b-\frac{1}{2}\sqrt{b^2-4\,a}\},$$
$$\{y=\frac{1}{2}b-\frac{1}{2}\sqrt{b^2-4\,a},x=\frac{1}{2}b+\frac{1}{2}\sqrt{b^2-4\,a}\}$$

```
> s:=solve({x*y=2,x+y=3},{x,y});
```

$$s:=\{y=1,x=2\},\{y=2,x=1\}$$

```
> assign(s);x;y;
```

1

2

```
> unassign('x');y:='y';
```

$$y:=y$$

```
> [x,y];
```

$$[x,y]$$

В этих примерах хорошо видна техника работы с функциями solve и assign. В конце примеров показано восстановление неопределенного статуса переменных x и y с помощью функции unassign и снятие определения переменных с помощью заключения их в прямые апострофы.

Функция RootOf

В решениях уравнений нередко появляется функция RootOf, означающая, что корни нельзя выразить в радикалах. Эта функция применяется и самостоятельно в виде RootOf(expr) или RootOf(expr, x), где expr — алгебраическое выражение или равенство, x — имя переменной, относительно которой ищется решение. Если x не указана, ищется универсальное решение по переменной _Z. Когда expr задано не в виде равенства, решается уравнение expr=0. Для получения решений вида RootOf в явном виде может использоваться функция allvalues.

ПРИМЕРЫ

```
> RootOf(x^2+1=0,x);
```

$\mathrm{RootOf}(\_Z^2+1)$

```
> allvalues(%);
```

$I, -I$

```
> RootOf(a*b^2+a/b,b);
```

$\mathrm{RootOf}(\_Z^3+1)$

```
> allvalues(%);
```

$-1, \frac{1}{2}+\frac{1}{2}I\sqrt{3}, \frac{1}{2}-\frac{1}{2}I\sqrt{3}$

```
> RootOf(x^3-1,x)mod 7;
```

$\mathrm{RootOf}(\_Z^3+6)$

```
> allvalues(%);
```

$-6^{(1/3)}, \frac{1}{2}6^{(1/3)}-\frac{1}{2}I\sqrt{3}\,6^{(1/3)}, \frac{1}{2}6^{(1/3)}+\frac{1}{2}I\sqrt{3}\,6^{(1/3)}$

```
> evalf(%);
```

$-1.817120593, .9085602965-1.573672596\,I, .9085602965+1.573672596\,I$

```
> RootOf(x^2-2*x+1,x)mod 5;
```

1

Решение неравенств

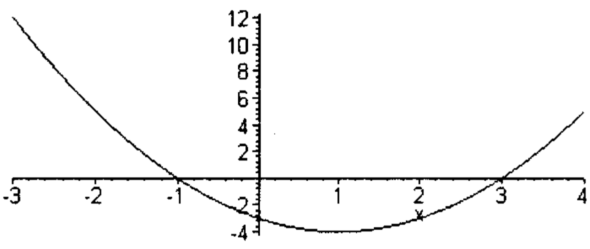
Неравенства в математике встречаются почти столь же часто, как и равенства. Они вводятся знаками отношений, например: > (больше), < (меньше) и т. д. Решение неравенств существенно расширяет возможности функции solve. При этом неравенства задаются так же, как и равенства.

Примеры, иллюстрирующие решение неравенств

```
> solve(x^2-2*x-3>0,x);
```

$\mathrm{RealRange}(-\infty, \mathrm{Open}(-1)), \mathrm{RealRange}(\mathrm{Open}(3), \infty)$

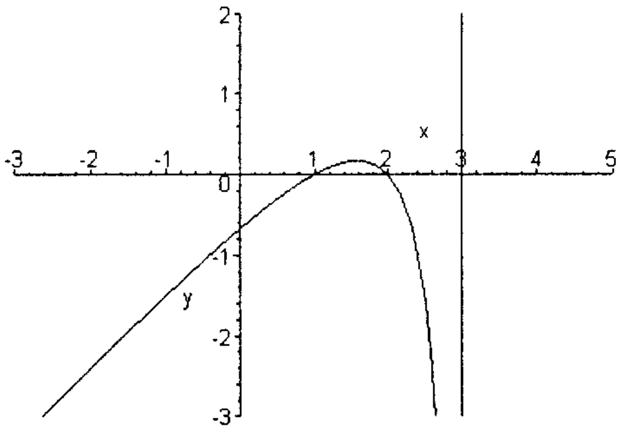
```
> plot(x^2-2*x-3,x=-3..4,color=black);
```



```
> solve((x-1)*(x-2)/(x-3)<1,x);
```

$$\mathrm{RealRange}(-\infty, \mathrm{Open}(3))$$

```
> plot((x-1)*(x-2)/(x-3),x=-3..5,y=-3..2,color=black);
```



Примеры решения неравенств в аналитической форме:

```
> solve(5*x>10,x);
```

$$\mathrm{RealRange}(\mathrm{Open}(2), \infty)$$

```
> solve(5*x>=10,x);
```

$$\mathrm{RealRange}(2, \infty)$$

```
> solve(ln(x)>2,x);
```

$$\mathrm{RealRange}(\mathrm{Open}(e^2), \infty)$$

```
> solve(exp(x)>10,x);
```

$$\mathrm{RealRange}(\mathrm{Open}(\ln(10)), \infty)$$

```
> solve(a*x>b,{x});
```

$$\{-\mathrm{signum}(a)\, x < -\frac{\mathrm{signum}(a)\, b}{a}\}$$

```
> eqns := abs(z)^2/(z+1) < exp(2)/(exp(1)-1);
```

$$eqns := \frac{|z|^2}{z+1} < \frac{e^2}{e-1}$$

```
> solve( eqns, {z} );
```

$$\left\{z < \frac{1}{2}\frac{e^2 + \sqrt{(e^2)^2 + 4\,e^2\,e - 4\,e^2}}{e-1}, \frac{1}{2}\frac{e^2 - \sqrt{(e^2)^2 + 4\,e^2\,e - 4\,e^2}}{e-1} < z\right\}, \{z < -1\}$$

```
> eqns := exp(x)*x^2 >= 1/2;
```

$$eqns := \frac{1}{2} \le e^x\, x^2$$

```
> solve( eqns, {x} );
```

$$\{2\ \mathrm{LambertW}\left(-1,-\frac{1}{4}\sqrt{2}\right) \le x, x \le 2\ \mathrm{LambertW}\left(-\frac{1}{4}\sqrt{2}\right)\}, \{2\ \mathrm{LambertW}\left(-1,-\frac{1}{4}\sqrt{2}\right) \le x\}$$

```
> eqns := abs( (z+abs(z+2))^2-1 )^2 = 9;
```

$$eqns := \left|(z+|z+2|)^2-1\right|^2 = 9$$

```
> solve( eqns, {z} );
```

$\{z=0\}, \{z \le -2\}$

```
> evalf(%);
```

$\{-2.617866616 \le x, x \le -1.487962064\}, \{.5398352768 \le x\}$

```
> eqns := { x^2<1, y^2<=1, x+y<1/2 };
```

$$eqns := \{x^2 < 1, y^2 \le 1, x+y < \frac{1}{2}\}$$

```
> solve( eqns, { x, y } );
```

$$\{y \le 1, -1 \le y, x+y < \frac{1}{2}, -1 < x, x < 1\}$$

```
> solve({x*y*z>0,x>-1,y+z>10},{x,y,z});
```

$\{z=0, -1 < x, 10 < y\}, \{y=0, -1 < x, 10 < z\}$

Решение в численном виде — функция fsolve

Для получения численного решения нелинейного уравнения или системы нелинейных уравнений в форме вещественных чисел удобно использовать функцию:

```
fsolve( eqns, vars, options )
```

Эта функция может быть использована со следующими параметрами:

- `complex` — находит один или все корни полинома в комплексной форме;
- `fulldigits` — задает вычисления для полного числа цифр, заданного функцией `Digits`;
- `maxsols=n` — задает нахождение только `n` корней;
- `interval` — задается в виде `a..b` или `x=a..b`, или `{x=a..b, y=c..d, ...}` и обеспечивает поиск корней в указанном интервале.

Функция `fsolve` дает решения сразу в форме вещественных или комплексных чисел.

ПРИМЕРЫ

```
> fsolve(sin(x)=Pi/4,x);
```

.9033391108

```
> fsolve(sin(x)=1/2,x=4..8);
```

6.806784083

```
> fsolve(2*x^2+x-1=10,x);
```

-2.608495283, 2.108495283

```
> fsolve(x^5-x,x);
```

-1., 0., 1.000000000

```
> fsolve(x^5-x,x,complex);
```

-1.000000000, -1.000000000 *I*, 0., 1.000000000 *I*, 1.000000000

```
> eqns := abs(x)*x+exp(x) > 0;
```

$eqns := 0 < |x|\,x + e^x$

```
> solve( eqns, {x} );
```

$\{-2\ \mathrm{LambertW}\left(\frac{1}{2}\right) < x\}$

```
> f := sin(x+y) - exp(x)*y = 0:
g := x^2 - y = 2:
fsolve({f,g},{x,y},{x=-1..1,y=-2..0});
```

$\{x = -.6687012050, y = -1.552838698\}$

Заметим, что локализация поиска корней в заданном интервале позволяет отыскивать такие решения, которые не удается получить с помощью функций solve и fsolve в обычном применении. В последнем из приведенных примеров дается решение системы нелинейных уравнений, представленных уравнениями f и g. Чтобы еще раз показать различие между функциями solve и fsolve, рассмотрим пример решения с их помощью одного и того же уравнения erf(x) = 1/2:

```
> solve(erf(x)=1/2,x);
```

RootOf(2 erf(_Z) – 1)

```
> fsolve(erf(x)=1/2);
```

.4769362762

Функция solve в этом случае находит нетривиальное решение в комплексной форме через функцию RootOf, тогда как функция fsolve находит обычное приближенное решение.

Решение рекуррентных уравнений — rsolve

Для решения рекуррентных уравнений используется функция rsolve:

```
rsolve(eqns, fcns)
rsolve(eqns, fcns, 'genfunc'(z))
rsolve(eqns, fcns, 'makeproc')
```

Здесь eqns — одиночное уравнение или система уравнений, fcns — функция, имя функции или множество имен функций, z — имя, генерирующее функциональную переменную.

ПРИМЕРЫ

```
> restart;
> rsolve(f(n)=-2*f(n-1)-f(n-2),f(k));
```

$$(-f(0)-f(1))(k+1)(-1)^k+(f(1)+2f(0))(-1)^k$$

```
> rsolve({f(n)=-3*f(n-1)-2*f(n-2),f(1..2)=1},{f});
```

$$\{f(n)=-3(-1)^n+(-2)^n\}$$

```
> rsolve({y(n)=n*y(n-1),y(0)=1},y);
```

$$\Gamma(n+1)$$

```
> rsolve({y(n)*y(n-1)+y(n)-y(n-1)=0,y(0)=a},y);
```

$$\frac{a}{1+n\,a}$$

```
> rsolve({F(n)=F(n-1)+F(n-2),F(1..2)=1},F,'genfunc'(x));
```

$$-\frac{x}{-1+x+x^2}$$

```
> rsolve({y(n+1)+f(n)=2*2^n+n,f(n+1)-y(n)=n-2^n+3,y(k=1..5)=2^k-1,f(5)=6},{y,f});
```

$$\{f(n)=n+1,\ y(n)=2^n-1\}$$

ПРИМЕР вычисления функцией rsolve n-го числа Фибоначчи

```
> eq1 := {f(n+2) = f(n+1) + f(n) , f(0) = 1 , f(1) = 1};
```

$$eq1 := \{f(n+2)=f(n+1)+f(n),\ f(0)=1,\ f(1)=1\}$$

```
> a1:=rsolve(eq1,f);
```

$$a1 := \frac{2}{5}\,\frac{\sqrt{5}\left(2\,\frac{1}{-1+\sqrt{5}}\right)^n}{-1+\sqrt{5}}+\frac{\frac{2}{5}\sqrt{5}\left(-2\,\frac{1}{1+\sqrt{5}}\right)^n}{1+\sqrt{5}}$$

Числа Фибоначчи — целые числа. Поэтому представленный результат выглядит как весьма сомнительный. Но на самом деле он точный и с его помощью можно получить числа Фибоначчи. Ниже показан процесс получения чисел Фибоначчи для n = 5, 7, 10 и 20:

```
> [normal(subs(n=5,a1),expanded),normal(subs(n=7,a1),expanded),
                    normal(subs(n=10,a1),expanded),normal(subs(n=20,a1),expanded)];
```

$$[8, 21, 89, 10946]$$

Решение уравнений в целочисленном виде — isolve

Иногда бывает нужен результат в форме только целых чисел. Для этого используется функция `isolve(eqns, vars)`, дающая решение в виде целых чисел. Приведем примеры:

```
> isolve({2*x-5=3*y});
```

$$\{x=4+3\_Z1,\ y=1+2\_Z1\}$$

```
> isolve(y^4-z^2*y^2-3*x*z*y^2-x^3*z);
```

$$\{z = \frac{\_Z3\ \_Z2^4}{\mathrm{igcd}(-\_Z1^2\,(\_Z2^2 - \_Z1^2), -\_Z1^3\ \_Z2, \_Z2^4)},$$
$$x = -\frac{\_Z3\ \_Z1^2\,(\_Z2^2 - \_Z1^2)}{\mathrm{igcd}(-\_Z1^2\,(\_Z2^2 - \_Z1^2), -\_Z1^3\ \_Z2, \_Z2^4)},$$
$$y = -\frac{\_Z3\ \_Z1^3\ \_Z2}{\mathrm{igcd}(-\_Z1^2\,(\_Z2^2 - \_Z1^2), -\_Z1^3\ \_Z2, \_Z2^4)}\}$$

Функция msolve

Функция `msolve(eqns,vars,m)` или `msolve(eqns,m)` обеспечивает решение вида Z mod m (то есть при подстановке решения левая часть при делении на m дает остаток, равный правой части уравнения). При отсутствии решения возвращается объект `NULL` (пустой список).

ПРИМЕРЫ

```
> msolve({3*x-4*y=1,7*x+y=2},12);
```
$$\{y = 5, x = 3\}$$
```
> msolve(2^i=3,19);
```
$$\{i = 13 + 18\ \_Z1\sim\}$$
```
> msolve(8^j=2,x,17);
```
$$\{j = 3 + 8\,x\}$$

Поиск экстремумов функций

С помощью функции `fsolve` легко находятся значения независимой переменной x функций вида $f(x)$, при которых $f(x) = 0$ (корни этого уравнения). При этом данная функция позволяет (в отличие от функции `solve`) изолировать корни функции $f(x)$ указанием примерного интервала их существования. Ряд функций служит для вычисления экстремумов, максимумов и минимумов функций, а также для определения их непрерывности. Одна из таких функций, `extrema`, позволяет найти экстремумы выражения `expr` (как максимумы, так и минимумы) при ограничениях `constrs` и переменных `vars`, по которым ищется экстремум:

```
extrema(expr, constrs)
extrema(expr, constrs, vars)
extrema(expr, constrs, vars, 's')
```

Ограничения `contrs` и переменные `vars` могут задаваться одиночными объектами или списками ряда ограничений и переменных. Найденные координаты точки экстремума присваиваются переменной `'s'`. При отсутствии ограничений в виде равенств или неравенств вместо них записывается пустой список {}.

ПРИМЕРЫ

```
> extrema(a*x^2+b*x+c,{},x,'s');s;
```
$$\{-\frac{1}{4}\frac{b^2 - 4\,c\,a}{a}\}$$
$$\{\{x = -\frac{1}{2}\frac{b}{a}\}\}$$

```
> extrema(x*exp(-x),{},x,'s');s;
```

$$\{e^{(-1)}\}$$

$$\{\{x=1\}\}$$

```
> extrema(sin(x)^2,{},x,'s');s;
```

$$\{0, 1\}$$

$$\{\{x=0\}, \{x=\frac{1}{2}\pi\}\}$$

```
> extrema(x+y/z,x^2+y^2+z^2=1,{x,y,z},'s');s;
```

$$\{\max(1-\mathrm{RootOf}(\_Z^4+1)^2, -1+\mathrm{RootOf}(\_Z^4+1)^2),$$
$$\min(1-\mathrm{RootOf}(\_Z^4+1)^2, -1+\mathrm{RootOf}(\_Z^4+1)^2)\}$$

$$\{\{z=\mathrm{RootOf}(\_Z^4+1), x=-1, y=\mathrm{RootOf}(\_Z^4+1)^3\},$$
$$\{x=1, z=\mathrm{RootOf}(\_Z^4+1), y=-\mathrm{RootOf}(\_Z^4+1)^3\}\}$$

Поиск минимумов и максимумов аналитических функций

Часто нужно найти минимум или максимум заданной функции. Для поиска минимумов и максимумов выражений (функций) expr служат функции стандартной библиотеки:

```
minimize(expr, opt1, opt2, ..., optn)
maximize(expr, opt1, opt2, ..., optn)
```

Эти функции могут разыскивать максимумы и минимумы для функций как одной, так и нескольких переменных. С помощью опций opt1, opt2, ..., optn можно указывать дополнительные данные для поиска. Например, параметр 'infinity' означает, что поиск минимума или максимума выполняется по всей числовой оси, а параметр location (или location=true) дает расширенный вывод результатов поиска — выдается не только значение минимума (или максимума), но и значения переменных в этой точке.

ПРИМЕРЫ

Примеры использования функции minimize:

```
> minimize(x^2-3*x+y^2+3*y+3);
```

$$\frac{-3}{2}$$

```
> minimize(x^2-3*x+y^2+3*y+3, location);
```

$$\frac{-3}{2}, \left\{\left[\ \{y=\frac{-3}{2}, x=\frac{3}{2}\}, \frac{-3}{2}\right]\right\}$$

```
> minimize(x^2-3*x+y^2+3*y+3, x=2..4, y=-4..-2, location);
```

$$-1, \{[\ \{x=2, y=-2\}, -1\ ]\}$$

```
> minimize(x^2+y^2,x=-10..10,y=-10..10);
```

$$0$$

```
> minimize(x^2+y^2,x=-10..10,y=-10..10,location);
```

$$0, \{[\ \{y=0, x=0\}, 0\ ]\}$$

```
> minimize(abs(x*exp(-x^2)-1/2), x=-4..4);
```

$$\frac{1}{2}-\frac{1}{2}\sqrt{2}\ \mathrm{e}^{(-1/2)}$$

```
> minimize(abs(x*exp(-x^2)-1/2), x=-4..4,location=true);
```

$$\frac{1}{2}-\frac{1}{2}\sqrt{2}\ \mathrm{e}^{(-1/2)}, \left\{\left[x=\frac{1}{2}\sqrt{2}, \frac{1}{2}-\frac{1}{2}\sqrt{2}\ \mathrm{e}^{(-1/2)}\right]\right\}$$

Примеры использования функции maximize:

```
> maximize(x*exp(-x));
```

$$\mathrm{e}^{(-1)}$$

```
> maximize(x*exp(-x),location);
```

$$\mathrm{e}^{(-1)}, \{[\ \{x=1\}, \mathrm{e}^{(-1)}]\}$$

```
> maximize(sin(x)/x,x=-2..2,location);
```

$$\infty, \{[\ \{x=0\}, \infty\ ]\}$$

```
> maximize(exp(-x)*sin(y),x=-10..10,y=-10..10,location);
```

$$\mathrm{e}^{10}, \left\{\left[\ \{y=-\frac{3}{2}\pi, x=-10\}, \mathrm{e}^{10}\right], \left[\ \{x=-10, y=\frac{5}{2}\pi\}, \mathrm{e}^{10}\right], \left[\ \{y=\frac{1}{2}\pi, x=-10\}, \mathrm{e}^{10}\right]\right\}$$

Поиск минимума функции Розенброка и построение ее графика

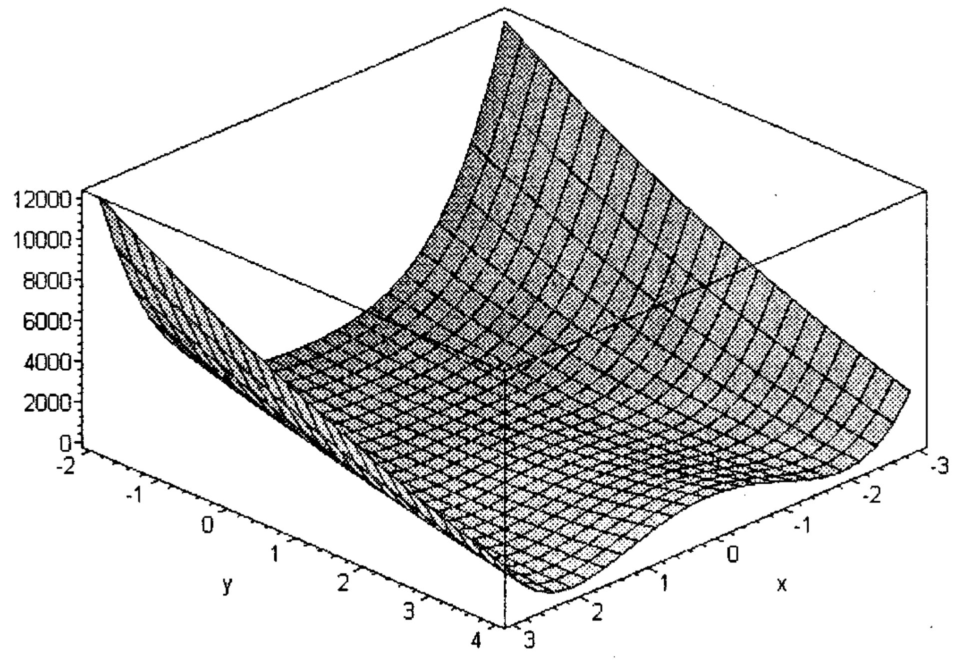
```
> rf:=(x,y)->100*(y-x^2)^2+(1-x)^2;
```

$$rf := (x, y) \to 100\,(y-x^2)^2 + (1-x)^2$$

```
> minimize(rf(x,y),location);
```

$$0, \{[\{x=1, y=1\}, 0]\}$$

```
> plot3d(rf(x,y),x=-3..3,y=-2..4,axes=BOXED);
```


12000
10000
8000
6000
4000
2000
0
-2
-1
0
1
2
3
4
y
3
2
1
0
-1
-2
-3
x