Solución de ecuacións e sistemas de ecuacións

MAPLE pode resolver ecuacións e inecuacións (con unha ou varias incógnitas) de xeito **simbólico** (obte-la expresión analítica da solución; isto só é posíbel en ecuacións relativamente simples) ou **numérico** (obter valores numéricos aproximados das solucións).

Resolución simbólica

A solución simbólica obtense co comando

solve(equations, variables)

onde

equations - ecuación ou inecuación (pode ser un set ou lista de ecuacións ou inecuacións).

variables - (opcional) nome ou lista de nomes das incógnitas do sistema (se non o introducimos considerará a todas as existentes).

Se MAPLE non é capaz de atopar unha solución devolve NULL.

Se como primeiro argumento introducimos unha expresión, MAPLE interpretao como expresión=0.

```
> solve(x+y=0, x); # os argumentos son unha ecuación e
  resólvese para a incognita x
```

$$-y \qquad \textbf{(1.1.1)}$$

```
> solve(x+y=0, {x,y}); # os argumentos son unha ecuación e
  resólvese para as incognitas x e y (un set).
```

$$\{x = -y,\ y = y\} \qquad \textbf{(1.1.2)}$$

```
> solve(x+y=0); # o mesmo efecto que no caso anterior
```

$$\{x = -y,\ y = y\} \qquad \textbf{(1.1.3)}$$

```
> solve(x+y, x); # os argumentos son unha expresión e
  resólvese para a incognita x
```

$$-y \qquad \textbf{(1.1.4)}$$

```
> solve({x+y=0}, {x}); # os argumentos son sets(conxuntos) de
  ecuacións e incognitas
```

$$\{x = -y\} \qquad \textbf{(1.1.5)}$$

Polo tanto, a función **solve** pode empregarse para despexar unha variábel en función de outra(s):

```
> solve(2*y-(x-1)^2 = 2, y);
```

$$\frac{1}{2}x^2 - x + \frac{3}{2} \qquad \textbf{(1.1.6)}$$

```
> solve({x+(1/4)*y+z = 1,  3.2*x+1.3*y+4.2*z = 5,  8.7*x+19*
  y+11.2*z = 94}, [x, y, z]); #sistema de 3 ecuacións con 3
  incognitas: as ecuacións van entre chaves (é un conxunto de
  ecuacións)
```

(1.1.7)

$$[[x = 0.4969502408,\ y = 5.187800963,\ z = -0.7939004815]] \quad \textbf{(1.1.7)}$$

```
> solve({x^2 = 9,  x+y < 10}, [x, y]); #resolvendo inecuacións
```

$$[[x = 3,\ y < 7],\ [x = -3,\ y < 13]] \quad \textbf{(1.1.8)}$$

```
> solve({a*x^2+b*x+c},{x}); # toma a expresión igualada a 0
```

$$\left\{x = \frac{1}{2}\ \frac{-b+\sqrt{b^2-4\,a\,c}}{a}\right\},\ \left\{x = -\frac{1}{2}\ \frac{b+\sqrt{b^2-4\,a\,c}}{a}\right\} \quad \textbf{(1.1.9)}$$

```
> x;
```

$$x \quad \textbf{(1.1.10)}$$

NOTA IMPORTANTE: se nos da a solución en termos de **RootOf**, podemos executar **allvalues(%)** e ás veces si calcula unha solución explícita (non en termos de **RoofOf**). Tamén podemos usar **evalf(%)** para obter a(s) solución(s) en punto flotante.

```
> s := solve({sin(x+y)-y·exp(x), x^2-y-2}, {x, y});
Warning, solutions may have been lost
```

$$s := \{x = RootOf(e^{-Z}\_Z^2 - \sin(\_Z^2 + \_Z - 2) - 2\,e^{-Z}),\ y = RootOf(e^{-Z}\_Z^2 - \sin(\_Z^2 + \_Z - 2) - 2\,e^{-Z})^2 - 2\} \quad \textbf{(1.1.11)}$$

```
> evalf(s);
```

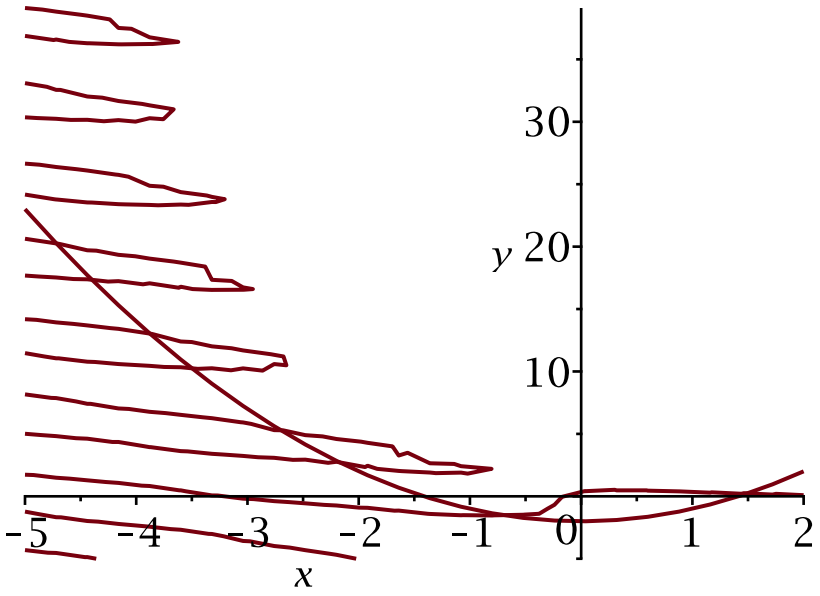
$$\{x = 1.490927732 + 0.\,I,\ y = 0.222865502 + 0.\,I\} \quad \textbf{(1.1.12)}$$

```
> evalf(allvalues(s));
```

$$\{x = -0.6687012050,\ y = -1.552838698\},\ \{x = -2.742322948,\ y = 5.520335151\},\ \{x = 1.490927732,\ y = 0.222865502\},\ \{x = 2.518820998 - 0.8147435813\,I,\ y = 3.680652117 - 4.104386481\,I\},\ \{x = 3.461326200 - 0.8260768945\,I,\ y = 9.29837603 - 5.718643196\,I\},\ \{x = 0.6078432441 - 1.104304339\,I,\ y = -2.850014664 - 1.342487864\,I\},\ \{x = 2.518820998 + 0.8147435813\,I,\ y = 3.680652117 + 4.104386481\,I\},\ \{x = 3.461326200 + 0.8260768945\,I,\ y = 9.29837603 + 5.718643196\,I\},\ \{x = 0.6078432441 + 1.104304339\,I,\ y = -2.850014664 + 1.342487864\,I\},\ \{x = -2.096487462,\ y = 2.395259678\},\ \{x = -0.9101370208 - 1.831561999\,I,\ y = -4.526269960 + 3.333944762\,I\},\ \{x = -0.9101370208 + 1.831561999\,I,\ y = -4.526269960 - 3.333944762\,I\} \quad \textbf{(1.1.13)}$$

Podemos comprobar as solucións representando gráficamente con **implicitplot** o sistema de ecuacións:

```
> with(plots) : implicitplot([sin(x+y)-y·exp(x), x^2-y-2], x=-5
    ..2, y=-5..40);
```



Maple, por defecto, non asigna as solucións ás variabeis. Podemos utiliza-lo comando **assign**.

```
> restart;
> res:=solve({cos(x)+y=9}, {x});
```

$$res := \{x = \pi - \arccos(y-9)\} \quad \textbf{(1.1.14)}$$

```
> x;
```

$$x \quad \textbf{(1.1.15)}$$

```
> assign(res): x;
```

$$\pi - \arccos(y-9) \quad \textbf{(1.1.16)}$$

Podes introducir hipóteses sobre as solucións ou sobre os parámetros (constantes) que aparecen na ecuación co comando **assuming**:

```
> solve(x^2-a, x) assuming a::negative;
```

$$I\sqrt{-a},\ -I\sqrt{-a} \quad \textbf{(1.1.17)}$$

```
> solve(x^2-a, x)
```

$$\sqrt{a},\ -\sqrt{a} \quad \textbf{(1.1.18)}$$

Pódese comproba-las solucións substituindo as solucións nas ecuacións orixinais co comando **eval** que neste contexto realiza a substitución

eval(e, x, = a) $e\Big|_{x=a}$

ou tamén co comando **subs**: **subs(x= a,expr)**

```
> restart;
> ecs:={x+2*y=3, y+1/x=1}; # Set de ecucións
```

$$ecs := \left\{x + 2y = 3,\ y + \frac{1}{x} = 1\right\} \quad \textbf{(1.1.19)}$$

```
> sols:=solve(ecs, {x, y});
```

(1.1.20)

$$sols := \{x = -1, y = 2\}, \left\{x = 2, y = \frac{1}{2}\right\} \quad \textbf{(1.1.20)}$$

```
> sols[1]; # unha solución
```

$$\{x = -1, y = 2\} \quad \textbf{(1.1.21)}$$

```
> sols[2]; # outra solución
```

$$\left\{x = 2, y = \frac{1}{2}\right\} \quad \textbf{(1.1.22)}$$

```
> eval(ecs, sols[1]);
```

$$\{1 = 1, 3 = 3\} \quad \textbf{(1.1.23)}$$

```
> eval(ecs, sols[2]);
```

$$\{1 = 1, 3 = 3\} \quad \textbf{(1.1.24)}$$

```
> subs(sols[1], ecs);
```

$$\{1 = 1, 3 = 3\} \quad \textbf{(1.1.25)}$$

Resolución numérica

Utilízase o comando **fsolve** que resolve únicamente ecuacións sen símbolos (só pode ter números):

fsolve(equations, complex)

onde

equations - ecuación, lista ou set de ecuacións

complex - (opcional) nome literal que hai que poñer cando se queren atopar solucións complexas.

A declaración do comando con tódalas opcións é:

fsolve(equations, variables, complex, fulldigits, interval, starting_values, options)

máis adiante mencionanse algunhas das opcións máis relevantes.

```
> restart;
> pol := 2*x^5-11*x^4-7*x^3+12*x^2-4*x = 0;
```

$$pol := 2x^5 - 11x^4 - 7x^3 + 12x^2 - 4x = 0 \quad \textbf{(1.2.1)}$$

```
> fsolve(pol);
```

$$-1.334383488,\ 0.,\ 5.929222024 \quad \textbf{(1.2.2)}$$

```
> pol2:=3*x^4-16*x^3-3*x^2+13*x+16;
```

$$pol2 := 3x^4 - 16x^3 - 3x^2 + 13x + 16 \quad \textbf{(1.2.3)}$$

```
> fsolve(pol2, x); # só mostra solucións reais se o polinomio
  ten coeficientes reais (con coeficientes complexos calcula
  tódalas solucións).
```

$$1.324717957,\ 5.333333333 \quad \textbf{(1.2.4)}$$

```
> fsolve(pol2, x, complex); # para mostra-las solucións
  complexas
```

$$-0.662358978622373 - 0.562279512062301\,I,\ -0.662358978622373 + 0.562279512062301\,I,\ 1.32471795724475,\ 5.33333333333333 \quad \textbf{(1.2.5)}$$

Podes obter resultados similares usando solve combinado con evalf:

```
> evalf(solve(pol2));
```

$$5.333333333,\ 1.324717958,\ -0.6623589786+0.5622795125\,I,\ -0.6623589786-0.5622795125\,I \qquad \textbf{(1.2.6)}$$

Para limita-lo número de solucións utilízase a opción **maxsols**.

```
> fsolve(pol2, {x}, maxsols=1); # mostra só unha solución
```

$$\{x=1.324717957\} \qquad \textbf{(1.2.7)}$$

```
> fsolve(sin(x)=0, {x}); # especifíacase o argumento variabeis
```

$$\{x=0.\} \qquad \textbf{(1.2.8)}$$

Se o programa só nos proporciona unha solución que non nos interesa, podemos forzalo a que proporcione outras solucións coa opción **avoid.**

```
> fsolve(sin(x)=0, {x}, avoid={x=0});
```

$$\{x=-3.141592654\} \qquad \textbf{(1.2.9)}$$

Tamén se pode especificar un intervalo de búsqueda.

```
> fsolve(pol2, {x}, -Pi..Pi);
```

$$\{x=1.324717957\} \qquad \textbf{(1.2.10)}$$

```
> f:=sin(x+y)-exp(x)*y=0;
```

$$f:=\sin(x+y)-e^{x}y=0 \qquad \textbf{(1.2.11)}$$

```
> g:=x^2-y=2;
```

$$g:=x^{2}-y=2 \qquad \textbf{(1.2.12)}$$

```
> fsolve({f,g}, {x,y},{x=-1..1, y=-2..0});
```

$$\{x=-0.6687012050,\ y=-1.552838698\} \qquad \textbf{(1.2.13)}$$

E, finalmente, tamén podemos definir un punto de comezo (opción starting_values):

```
> fsolve(sin(x), x=3.25);
```

$$3.141592654 \qquad \textbf{(1.2.14)}$$

Resolución de ecuacións recurrentes

As ecuacións recurrentes son da forma:

$$f(n)=F(\{f(n-k),\ k=1,\ \ldots,\ K\}),\ f(n_1)=F_1,\ldots,\ f(n_{1+K})=F_K$$

É dicir, coñecemos o termo *n* en función de *K* termos anteriores, e coñecemos *K* termos iniciais $F_1, \ldots, F_K$. Pódense resolver co comando **rsolve({expresión recurrente, valores iniciais}, incógnitas)**, onde a expresión e os valores inicialis son os anteriores, e a incógnita é o valor a obter. Por exemplo, os números de Fibonacci están definidos por $f(1)=f(2)=1,\ f(n)=f(n-1)+f(n-2),\ n>2$. Neste caso, podemos resolver esta ecuación recurrente (é dicir, obter *f* como función de *n*) usando:

```
> rsolve({f(n)=f(n-1)+f(n-2),f(1)=1,f(2)=1},f(n));
```

$$\frac{1}{5}\sqrt{5}\left(\frac{1}{2}\sqrt{5}+\frac{1}{2}\right)^{n}-\frac{1}{5}\sqrt{5}\left(-\frac{1}{2}\sqrt{5}+\frac{1}{2}\right)^{n} \qquad \textbf{(1.3.1)}$$

```
> evalf(%)
```

$$0.4472135954\,1.618033988^{n} - 0.4472135954\,(-0.6180339880)^{n} \qquad \textbf{(1.3.2)}$$

Solucións enteiras de ecuacións

A función **isolve(ecuacións, vars)** permite resolver ecuacións obtendo só solucións enteiras. Se as solución están parametrizadas (é dicir, se 3 é solución, $3k\ \forall k \in \mathbb{Z}$ tamén é solución), **vars** son os parámetros destas solucións. Por exemplo, dada a ecuación (ten 2 incógnitas, e polo tanto só podemos poñer unha incógnita como función da outra) $3x-5y=7$, podemos resolvela con:

```
> restart;isolve(3*x-5*y=7);
```

$$\{x = 4 + 5\,\_Z1,\ y = 1 + 3\,\_Z1\} \qquad \textbf{(1.4.1)}$$

```
> isolve(3*x-5*y=7,k);
```

$$\{x = 4 + 5\,k,\ y = 1 + 3\,k\} \qquad \textbf{(1.4.2)}$$

Vemos polo tanto que k é o valor que parametriza o conxunto de solucións. Dándolle un valor a k obtemos os valores de x e y:

```
> subs(k=1, %);
```

$$\{x = 9,\ y = 4\} \qquad \textbf{(1.4.3)}$$