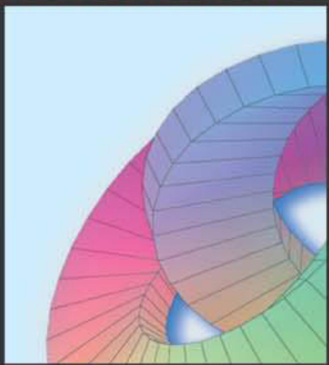
COMPANION WEBSITE
CWS
www.pearson-studium.de



Michael Kofler
Gerhard Bitsch
Michael Komma

st
scientific tools

Maple

Einführung, Anwendung, Referenz

5. Auflage

ADDISON-WESLEY

Pearson Studium

Michael Kofler, Gerhard Bitsch, Michael Komma

Maple

Einführung, Anwendung, Referenz

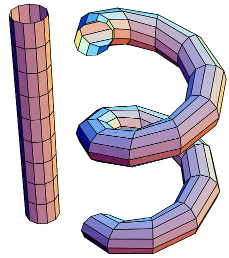
5., vollständig überarbeitete Auflage

Pearson Studium

ein Imprint der Pearson Education Deutschland GmbH

Kapitel 13

Gleichungen analytisch und numerisch lösen



Dieses Kapitel fasst die wichtigsten Befehle zum Lösen einzelner Gleichungen oder ganzer Gleichungssysteme zusammen, wobei sowohl analytische als auch numerische Lösungsverfahren behandelt werden.

`solve`
ist die Standardfunktion zur analytischen (symbolischen) Lösung von Gleichungen.

`fsolve`
verwendet verschiedene numerische Verfahren, um Gleichungen zu lösen.

`isolve` und `rsolve`
sind zwei Varianten zu `solve`. `isolve` ermittelt ausschließlich ganzzahlige Lösungen, `rsolve` versucht, aus der rekursiven Definition einer Funktion die allgemeine Funktionsvorschrift abzuleiten.

Verweis: Neben den oben erwähnten `solve`-Varianten existieren in Maple zahlreiche weitere Kommandos, die im weiteren Sinne ebenfalls zum Lösen von Gleichungssystemen dienen: `simplify` ist in der Lage, aus einem Gleichungssystem mit mehreren Variablen einzelne Variablen zu eliminieren (S. 178). `LinearSolve` löst Gleichungssysteme, die als Matrizen angeschrieben sind (S. 228), `dsolve` löst Differentialgleichungen (Kapitel 18), `extrema` löst Extremwertaufgaben unter Berücksichtigung von Nebenbedingungen und schließlich führen `minimize` und `maximize` aus dem Package `simplex` lineare Optimierungen durch (Kapitel 23). Weitere Informationen zur Anwendung des `solve`-Kommandos finden Sie in der Maple-Datei `Examples\Solve.mws`.

Gleichungen analytisch lösen (solve)

Die Standardfunktion zum Lösen von Gleichungen heißt `solve`. Der Funktion wird im ersten Argument die zu lösende Gleichung, im zweiten Argument die Variable übergeben.

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

$$-1, -2$$

Wenn statt einer Gleichung nur ein Term angegeben wird, erzeugt Maple automatisch eine Gleichung, die den Term gleich 0 setzt. Wenn auf die Angabe von Variablen verzichtet wird, löst Maple die Gleichung für alle der darin vorkommenden unbekannten Symbole. Welche der Symbole dabei in Abhängigkeit von den anderen aufgelöst werden, ist unbestimmt und kann von Sitzung zu Sitzung differieren.

```
solve(x^2+3*x+2);
```

$$-1, -2$$

Die Angabe der Variable, nach der die Gleichung aufgelöst werden soll, ist erforderlich, wenn in einer Gleichung mehrere Unbekannte auftreten. Maple kann dann nicht erkennen, welche Symbole Variablen und welche später einzusetzende Parameter der Gleichung sind.

```
solve( a*x^2+b*x+c);
```

$$\{c = -ax^2 - bx, x = x, a = a, b = b\}$$

```
solve( a*x^2+b*x+c,x);
```

$$\frac{-b + \sqrt{b^2 - 4\,ac}}{2\,a}, \frac{-b - \sqrt{b^2 - 4\,ac}}{2\,a}$$

`solve` versucht, sämtliche Lösungen zu ermitteln. Die Reihenfolge der Lösungen ist unbestimmt.

```
f:=expand( (x-1)^2*(x+Pi)*(3*x-1/x) );
```

$$f := 3\,x^4 + 2\,x^2 + 3\,x^3\pi + 2\,x\pi - 6\,x^3 + 2\,x - 6\,x^2\pi + 2\,\pi - 1 - \frac{\pi}{x}$$

```
solve(f);
```

$$\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}, 1, -\pi$$

Lösungen mit Gleitkommazahlen

`solve` versucht, die Lösung von Gleichungen symbolisch zu ermitteln.

```
solve(x^3=8);
```

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

Wenn in der Gleichung numerische Werte auftreten (also 1.5 statt $3/2$), dann werden diese Werte zuerst durch Brüche angenähert, um die Berechnung symbolisch durchzuführen. Die Lösung wird anschließend wieder in Gleitkommazahlen zurückverwandelt.

```
solve(x^3=8.0);
```

$$2.0, -1.0 - 1.732050808\, I, \\ -1.0 + 1.732050808\, I$$

`solve` in Kombination mit numerischen Werten ist aber nicht gleichwertig mit dem Kommando `fsolve`, das etwas weiter unten behandelt wird. Beispielsweise liefert `fsolve` standardgemäß (außer bei Polynomen) nur eine einzige Lösung (nicht mehrere).

```
fsolve(x^3=8);
```

$$2$$

```
fsolve(x^3=8, x, complex);
```

$$-1.0 - 1.732050808\, I, \\ -1.0 + 1.732050808\, I, 2.0$$

Außerdem ist `fsolve` natürlich nicht in der Lage, Gleichungen mit symbolischen Parametern zu lösen, dies ist das Aufgabengebiet von `solve`.

```
solve(x^2=\frac{3}{2}+a,x);
```

$$\frac{1}{2}\sqrt{6+4\,a}, -\frac{1}{2}\sqrt{6+4\,a}$$

```
fsolve(x^2=1.5+a,x);
  Error, (in fsolve) a is in the
    equation, and is not solved for
```

Periodische Lösungen bei trigonometrischen Gleichungen

`solve` kommt auch mit trigonometrischen Gleichungen zurecht.

```
solve(sin(x)=cos(x));
```

$$\frac{\pi}{4}$$

Maple kann sogar periodische Lösungen trigonometrischer Gleichungen (in der Art $x = \pi/4 + n\pi$) bestimmen. Dazu muss vor dem `solve`-Kommando die Variable `_EnvAllSolutions` auf `true` gesetzt werden. Maple formuliert die Lösung dann mit den Variablen $\_Z$ für ganze Zahlen, $\_NN$ für nicht-negative ganze Zahlen oder $\_B$ für binäre Werte (0/1). Diese Variablen sind in der globalen Umgebung nicht definiert (wie das gemacht wird, sehen Sie in Kapitel 29) und können daher auch nicht substituiert werden. Die Lösungen zeigen nur, wie man von den Grundlösungen zu den weiteren (periodisch auftretenden) Lösungen gelangt.

```
_EnvAllSolutions:=true:
solve(sin(x)=cos(x));
```

$$1/4\,\pi + \pi\,\text{`\_Z~`}$$

```
s:=solve(sin(x^2)=1/2);
```

$$s := \frac{1}{6}\sqrt{6\,\pi + 24\,\pi\,\_B1\tilde{} + 72\,\pi\,\_Z1\tilde{}}, \\ -\frac{1}{6}\sqrt{6\,\pi + 24\,\pi\,\_B1\tilde{} + 72\,\pi\,\_Z1\tilde{}}$$

Gleichungssysteme

Wenn `solve` ganze Gleichungssysteme lösen soll, müssen sowohl die Gleichungen als auch die darin vorkommenden Variablen als Mengen geschrieben werden (geschwungene Klammern). Sofern im Gleichungssystem außer den Variablen, nach denen es gelöst werden soll, keine weiteren freien Variablen vorkommen, kann auf die Angabe der Variablenmenge auch verzichtet werden. `solve` gruppiert die einzelnen Lösungen in Mengen und gibt die Folge dieser Mengen als Ergebnis aus.

```
solve( {5*x^2 - 5*y^2 - 3*x + 9*y=0, 5*x^3+5*y^3-15*x^2-13*x*y-y^2=0}, {x,y});
```

$$\{y=0, x=0\}, \{y=-1, x=2\}, \{y=2, x=1\}, \left\{x=\frac{42}{25}, y=\frac{63}{25}\right\}$$

Bei der Lösung von Gleichungssystemen müssen die Anzahl der Gleichungen und die Anzahl der angegebenen Lösungsvariablen übereinstimmen.

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

$$\{y=1, z=5, x=-2\}$$

Maple kommt auch mit vielen nicht-linearen Gleichungssystemen zurecht: Im Folgenden Beispiel treten auch trigonometrische Funktionen und e-Terme auf. Mit `evalf` erfolgt eine numerische Auswertung des Ergebnisses. Anschließend wird das Ergebnis zur Kontrolle in die Ausgangsgleichungen eingesetzt.

```
eq:={exp(x)+exp(y)+exp(z)=1, sin(x)+sin(y)=1, y+z=1}:
solve(eq, {x,y,z});
```

$$\{y=-\arcsin(-1+\sin(\%2)),\ x=\%2,\ z=\arcsin(-1+\sin(\%2))+1\}$$

$$\%1 := e^{\arcsin(-1+\sin(\_Z))}$$

$$\%2 := \mathrm{RootOf}(\_Z - \ln(-1-e^{(\arcsin(-1+\sin(\_Z))+1)}\,\%1+\%1)+\arcsin(-1+\sin(\_Z)))$$

```
evalf(%);
```

$$\{x=1.926433253-1.711725280\,I,\ y=-1.000578839-1.314917616\,I,$$
$$z=2.000578839+1.314917616\,I\}$$

```
evalf(subs(%,eq));
```

$$\{1.000000005-.21\,10^{-8}\,I=1.,\ 1.000000000=1.,\ 1.000000001-.2\,10^{-9}\,I=1.\}$$

Verweis: Der Umgang mit Ungleichungssystemen wird am Ende dieses Kapitels behandelt (S. 208). Zur Elimination einer Variablen aus mehreren Gleichungen ist `simplify` mit der Angabe von Nebenbedingungen geeignet (S. 178). In manchen Fällen lassen sich Umformungen auch mit `algsubs` durchführen (S. 169). Informationen zum Umgang mit linearen Gleichungssystemen finden Sie schließlich noch in Kapitel 14, in Zusammenhang mit Vektor- und Matrizenrechnung (S. 228).

Die Befehle RootOf und allvalues

Für Polynome von höherem Grad als 4 gibt es keine uniformen Lösungsformeln mit Radikalen. `solve` ist daher nicht immer in der Lage, die Lösungen explizit anzugeben. Da nach dem Fundamentalsatz der Algebra aber für ein (univariates) Polynom vom Grad n stets n Lösungen existieren, muss sich Maple hier anders behelfen. Mit der Funktion `RootOf` beschreibt Maple symbolisch die Lösungen eines Polynoms. Dazu wird die Gleichung eines Polynoms zur Darstellung der aus ihm resultierenden Nullstellen verwendet. Ausdrücke mit `RootOf` können wie andere Ergebnisse weiterverarbeitet werden. `allvalues` ermittelt alle Lösungen des `RootOf`-Terms. Falls dabei eine symbolische Auswertung nicht möglich ist, arbeitet `allvalues` automatisch numerisch, gibt aber die Lösung mit einem `RootOf`-Ausdruck mit einem angehängten Label zurück, das die numerische Näherung beschreibt. Um die Lösungen zu unterscheiden, werden sie durch einen Indexausdruck gekennzeichnet. Mit `evalf` kann man diese Näherung anzeigen. (`allvalues` führt in dieser Situation zu einer Fehlermeldung.)

```
s:=solve(x^6+x+1,x);
```

$$s := RootOf(\_Z^6 + \_Z + 1, index = 1), RootOf(\_Z^6 + \_Z + 1, index = 2),$$
$$RootOf(\_Z^6 + \_Z + 1, index = 3), RootOf(\_Z^6 + \_Z + 1, index = 4),$$
$$RootOf(\_Z^6 + \_Z + 1, index = 5), RootOf(\_Z^6 + \_Z + 1, index = 6)$$

```
evalf(s);
```

$$-0.7906671888 - 0.3005069203I\ , ..., 0.9454023333 + 0.6118366938I$$

Gleichungen numerisch lösen (fsolve)

Das Kommando `fsolve` hat dieselbe Syntax wie `solve`, die Lösungssuche erfolgt allerdings numerisch. Der wesentliche Unterschied gegenüber `solve` besteht darin, dass `fsolve` normalerweise (außer bei Polynomen) nur eine einzige Lösung ermittelt, selbst dann, wenn `solve` mühelos mehrere erkennt. Aus diesem Grund sollte `fsolve` nur dann eingesetzt werden, wenn eine exakte (symbolische) Lösung nicht existiert oder wenn die zugrunde liegenden Gleichungen von vornherein Gleitkomma-Koeffizienten enthalten.

Die nebenstehende logarithmische Gleichung bereitet `fsolve` keine Schwierigkeiten.

```
fsolve(ln(1+x)+ln(3+x)+x=5);
```

$$2.192154954$$

Bei Polynomen ermittelt `fsolve` normalerweise alle reellen Lösungen. Durch die Option `complex` werden auch die komplexen Lösungen berechnet. Beachten Sie, dass `fsolve` nur bei Polynomen alle Lösungen ermittelt.

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

$$0.4533976515$$

```
fsolve( x^3+2*x-1, x,complex);
```

$$-0.2266988258 - 1.467711509I\,,$$
$$-0.2266988258 + 1.467711509I\,,$$
$$0.4533976515$$

Falls `solve` in der Lage ist, eine Lösung zu finden, sollten numerische Werte nicht durch `fsolve`, sondern durch die numerische Auswertung der symbolischen Lösung mit `evalf` berechnet werden. Der Vorteil besteht darin, dass dann alle Lösungen (und nicht nur eine einzige) ermittelt werden. Außerdem ist dieses Verfahren in der Regel numerisch genauer, da nicht sofort mit einem Näherungsverfahren gearbeitet wird.

```
fsolve({x+y=9/2, x*y=5}, {x,y});
```

$$\{x = 2.500000000, y = 2.0\}$$

```
solve({x+y=9/2, x*y=5}, {x,y}):
evalf(%);
```

$$\{x = 2.500000000, y = 2.0\}\,,$$
$$\{y = 2.500000000, x = 2.0\}$$

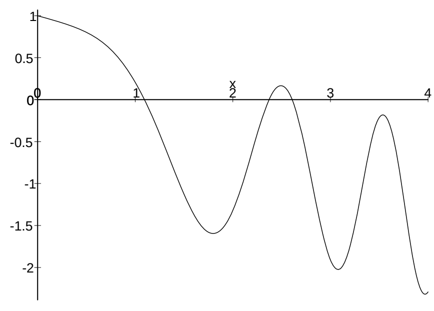
Es gibt Fälle, wo `fsolve` trotz einer scheinbar trivialen Gleichung keine Lösung liefert. In diesem Fall fehlt oft nur das zusätzliche Schlüsselwort `complex`, das `fsolve` auffordert, auch komplexe Lösungen zu berücksichtigen.

```
fsolve(x^2+5);
fsolve(x^2+5,x, complex);
```

$$-2.236067978I\,, 2.236067978I$$

Bei Funktionen, die keine Polynome sind, findet `fsolve` oft gar keine oder nur eine einzige Lösung, obwohl mehrere Lösungen existieren. Im vorliegenden Beispiel sollen die Nullstellen der Funktion $\cos(x^2) - x/3$ ermittelt werden. Ein kurzer Plot der Funktion ist für die Lösungssuche enorm hilfreich.

```
f:=cos(x^2)-x/3:
plot(f,x=0..4);
```



`fsolve` liefert auf Anhieb die mittlere der fünf Nullstellen.

```
fsolve( f );
```

$$2.371500399$$

Für die restlichen Nullstellen ist es erforderlich, Maple einen Tipp zu geben, in welchem Bereich ein Lösungswert zu erwarten ist. Diese Tipps werden in der Art `x=von..bis` formuliert.

```
fsolve( f,x, x=0..2);
```

$$1.094269232$$

```
fsolve(f, x, x=2.5..3);
```

$$2.607739533$$

```
fsolve(f, x, x=-2.5..-1.5 );
```

$$-1.996097497$$

```
fsolve(f, x, x=-1.9..-1.0 );
```

$$-1.439140143$$

Bei Berechnungen von Nullstellen mit schlechter Konditionierung ist die Option `fulldigits` nützlich. Diese Option verhindert, dass Maple die Rechengenauigkeit während der Suche nach einer Lösung vorübergehend reduziert, um so eine höhere Geschwindigkeit zu erreichen. Maple (und jedes andere CAS) ist mit der numerischen Lösung des folgenden Beispiels aber überfordert. Es zeigt eben, dass es Probleme gibt, die numerisch nicht zufriedenstellend behandelt werden können.

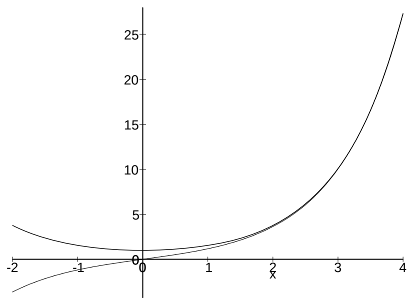
```
Digits:=40:      f:=sinh(x)-cosh(x):
lsg1:=fsolve(f); ;
```

$$lsg1 := 370.9460584024576331921177293763225157195$$

$$MitRelease4 : lsg2 := 57.31148716486415372856890701142667105759$$

Bei der Gleichung $\sinh(x) - \cosh(x)$ führt die Erhöhung der Rechengenauigkeit nur zu einem anderen falschen Ergebnis. Die beiden Hyperbelfunktionen nähern sich für $x \to \infty$ asymptotisch einander an, schneiden sich aber nie.

```
plot( {sinh(x), cosh(x)}, x=-2..4);
```



Mit symbolischen Methoden kommt man hier weiter. Beim ersten Versuch mit `is` muss Maple passen, wir müssen natürlich mitteilen, dass x reell ist.

```
is(cosh(x)-sinh(x)>0);
```

$$FAIL$$

```
is(cosh(x)-sinh(x)>0) assuming x::real;
```

true

Damit ist nachgewiesen, dass sich die Funktionen nie schneiden, weil für alle reellen Zahlen $x > 0$ stets $\sinh(x) < \cosh(x)$ ist. Dass der Grenzwert der Differenz 0 ist, kann Maple ebenfalls beweisen.

```
limit(cosh(x)-sinh(x),x=infinity);
```

$$0$$

Lösungen von Gleichungen weiterverwenden (eval, subs, assign)

Zur Weiterverarbeitung der Ergebnisse, die `solve` bzw. `fsolve` liefern, bestehen prinzipiell drei Möglichkeiten: Entweder werden die Ergebnisse verwendet, um mit der schon mehrfach behandelten Funktion `subs` die Variablen im Term durch die Lösung zu ersetzen, oder mit `eval` werden die Variablen vorübergehend zur Auswertung des Terms gebunden oder `assign` wird zur bleibenden Bindung der Werte an die Variablen verwendet. Im Regelfall ist die zweite Variante praktischer und flexibler. Sie erzeugt keine dauerhafte Bindung wie die dritte Variante und bewirkt eine Auswertung des Terms im Gegensatz zu einer Substitution wie die erste Variante.

Im vorliegenden Beispiel wird die erste Lösung der quadratischen Gleichung (also $x = 3$) in den Ausdruck $\sin(\frac{1}{1+x^2})$ eingesetzt. Beachten Sie die Verwendung der eckigen Klammern für den Zugriff auf das erste Teilergebnis der Lösungsfolge.

```
s:=solve( x^2-x-6 ,{x}});
```

$$s := x = 3, x = -2$$

```
eval(sin(1/(1+x^2)),s[1]);;
```

$$\sin(1/10)$$

Analog geht man bei Gleichungssystemen mit mehreren Variablen vor. Dort sind die Variablennamen nämlich bereits in der Lösung enthalten.

```
s:=solve( {x^2+y^2=5, x+y=3} );
```

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

```
eval(3*x^2+5*y-x*y,s[2]);
```

$$11$$

Zur Verarbeitung aller Lösungen (und nicht nur einer einzelnen) bietet sich die Funktion `seq` (S. 137) an. Im Beispiel rechts werden die Lösungen der Reihe nach vor der Auswertung an die Variable i gebunden.

```
seq( eval(3*x^2+5*y-x*y,i), i=s);
```

$$15, 11$$

Mit `assign` kann eine ganze Liste von Gleichungen der Art `var=..` in Bindungen umgesetzt werden. Im Beispiel rechts wird nochmals auf das obige Resultat Bezug genommen. x und y enthalten nun die Werte 1 und 2 und können nicht mehr als freie Variablen in Gleichungen verwendet werden.

```
assign( s[2] );
x,y;
```

$$1, 2$$

Das folgende Beispiel zeigt, wie die `simplify`-Funktion auf mehrere Lösungen angewandt wird. Der Versuch, die Lösung s direkt mit `simplify` zu bearbeiten, scheitert daran, dass `simplify` immer nur einen Term und keine Termfolge vereinfachen kann (die Terme nach dem ersten werden als zusätzliche Optionen interpretiert). Man wandelt daher die von `solve` gelieferte Folge durch Einschluss in eckige Klammern in eine Liste um:

```
_EnvAllSolutions:=true:
s:=solve(sin(x)+cos(x)=1/sqrt(2));
```

$$s := \arctan\left(\frac{\frac{1}{4}\sqrt{2}-\frac{1}{4}\sqrt{6}}{\frac{1}{4}\sqrt{2}+\frac{1}{4}\sqrt{6}}\right)+2\pi\,\_Z\tilde{},\ \arctan\left(\frac{\frac{1}{4}\sqrt{2}+\frac{1}{4}\sqrt{6}}{\frac{1}{4}\sqrt{2}-\frac{1}{4}\sqrt{6}}\right)+\pi+2\pi\,\_Z\tilde{}$$

```
simplify([s]);
```

$$[-\arctan(\frac{-1+\sqrt{3}}{1+\sqrt{3}})+2\pi\,\_Z\tilde{},\ -\arctan(\frac{1+\sqrt{3}}{-1+\sqrt{3}})+\pi+2\pi\,\_Z\tilde{}]$$

Das folgende Beispiel zeigt, wie die Lösungen einer trigonometrischen Gleichung überprüft werden können. Die Lösungen werden dazu der Reihe nach in die Ausgangsgleichungen eingesetzt und mit `simplify` vereinfacht. Falls `simplify` nicht wie in diesem Beispiel auf Anhieb funktioniert, spart eine numerische Auswertung mit `evalf` oft eine Menge Zeit.

```
_EnvAllSolutions:=false:
eq:=sin(x)^2+cos(x)=1/2: s:=solve( eq, x);
```

$$s := \arctan\left(\frac{1}{2}\,\frac{\sqrt{2}\,3^{1/4}}{-\frac{1}{2}\sqrt{3}+\frac{1}{2}}\right)+\pi,\ -\arctan\left(\frac{1}{2}\,\frac{\sqrt{2}\,3^{1/4}}{-\frac{1}{2}\sqrt{3}+\frac{1}{2}}\right)-\pi,$$
$$\arctan(\frac{1}{2}\sqrt{-2\sqrt{3}},\ \frac{1}{2}\sqrt{3}+\frac{1}{2}),\ \arctan(-\frac{1}{2}\sqrt{-2\sqrt{3}},\ \frac{1}{2}\sqrt{3}+\frac{1}{2})$$

```
seq(simplify(eval(eq,x=i)), i=s);
```

$$\frac{1}{2}=\frac{1}{2},\ \frac{1}{2}=\frac{1}{2},\ \frac{1}{2}=\frac{1}{2},\ \frac{1}{2}=\frac{1}{2}$$

```
seq(evalf(subs(x=i, eq)), i=s);
```

$$.4999999992 = .5000000000,\ .4999999992 = .5000000000,\ .5000000000 = .5000000000,$$
$$.5000000000 = .5000000000$$

Manchmal sollen die (komplexen) Lösungspunkte einer Gleichung grafisch angezeigt werden. Wir werten aus Gründen der besseren Übersichtlichkeit die Lösung numerisch aus, dann muss eine neue Liste erstellt werden, in der Real- und Imaginärteil voneinander getrennt sind. Dabei hilft wieder die Funktion `seq` weiter.

```
s:=solve( sin(x)^2+cos(x)+3 );
```

$$s := \arctan(\frac{1}{2}\sqrt{-14+2\sqrt{17}},\ \frac{1}{2}-\frac{1}{2}\sqrt{17}),\ \arctan(-\frac{1}{2}\sqrt{-14+2\sqrt{17}},\ \frac{1}{2}-\frac{1}{2}\sqrt{17}),$$
$$\arctan(\frac{1}{2}\sqrt{-14-2\sqrt{17}},\ \frac{1}{2}+\frac{1}{2}\sqrt{17}),\ \arctan(-\frac{1}{2}\sqrt{-14-2\sqrt{17}},\ \frac{1}{2}+\frac{1}{2}\sqrt{17})$$

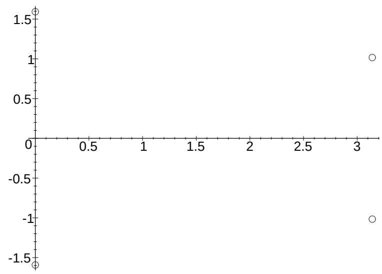
```
sf:=map(evalf,[s]);
```

$$sf := [3.141592654 - 1.015558753\,I,\ 3.141592654 + 1.015558753\,I,\ 1.593277489\,I,\ -1.593277489\,I]$$

```
pts:=[seq( [Re(i),Im(i)], i=sf)];
```

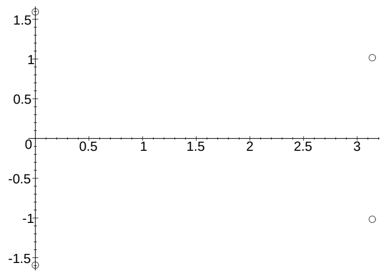
$$pts := [[3.141592654,\ -1.015558753],\ [3.141592654,\ 1.015558753],\ [0,\ 1.593277489],\ [0,\ -1.593277489]]$$

```
plot(pts, style=point, symbol=circle );
```



Im plots-Package steht die Funktion complexplot zur Verfügung. Diese Funktion kann komplexe Punkte unmittelbar darstellen, die manuelle Trennung von Real- und Imaginärteil kann entfallen. Die Lösung muss lediglich in eckige Klammern gestellt werden.

```
with(plots);
complexplot([s],
            style=point, symbol=circle);
```



Lösung rekursiver Gleichungen

Die Funktion rsolve eignet sich zur Bestimmung allgemeiner Funktionsvorschriften aus Funkionen, die in rekursiver Form gegeben sind.

Wir beginnen mit einem einfachen Beispiel. Wir definieren die Summe nsum der ersten n natürlichen Zahlen rekursiv auf folgende Weise: