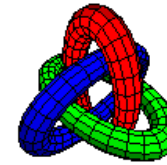


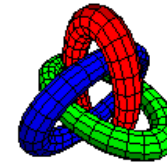
Komplexe Zahlen



Einführung in die
mathematische
Software
WS 2002/03

- Maple rechnet innerhalb der komplexen Zahlen!
- Eingabe i.a. in der Form $a+b*I$
- Umwandlung der komplexen Zahl z in Polarform durch `convert(z, polar)`
- direkte Eingabe in Polarform mit `polar(länge, winkel)`
- Maple geht davon aus, daß Unbestimmte aus \mathbb{C} sind => Vereinfachung math. Ausdrücke oft nicht möglich
daher: `evalc(ausdr)` vereinfacht `ausdr` unter der Annahme, daß alle Unbestimmten reell; das Ergebnis von `evalc(ausdr)` wird i.a. in der Form $a+I*b$ ausgegeben

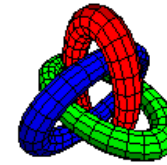
Funktionen in \mathbb{C}



Einführung in die
mathematische
Software
WS 2002/03

- $\text{Re}(z)$, $\text{Im}(z)$... Real- bzw. Imaginärteil von z
- $\text{conjugate}(z)$ berechnet die konjugiert komplexe Zahl zu z
- $\text{abs}(z)$ und $\text{argument}(z)$ liefern den Betrag und den Polarwinkel von z
- $\text{signum}(z)$ ergibt $z / |z|$
- $\text{csgn}(z)$:
 - 0 für $z=0$
 - +1 für $\text{Re}(z) > 0$ oder zugleich $\text{Re}(z) = 0$ und $\text{Im}(z) > 0$
 - -1 sonst

Folgen – Listen – Mengen



Einführung in die
mathematische
Software
WS 2002/03

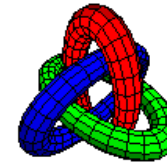
Um mehrere Objekte zusammenzufassen, unterstützt Maple die folgenden Datentypen:

- Folgen (sequence): z.B. x, y, z
- Listen (list): z.B. $[x, y, z]$
- Mengen (set): z.B. $\{x, y, z\}$

Zusätzlich gibt es noch:

- Felder (array)
- Tabellen (table)

Folgen (1)



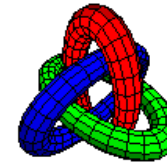
Einführung in die
mathematische
Software
WS 2002/03

- Reihe von Ausdrücken, die durch Beistriche getrennt sind (geordnet!), Verschachtelung nicht möglich

Erzeugung einer Folge durch:

- Aneinanderreihen von Ausdrücken; z.B. x, y, z
- die Funktion `seq`:
 - `seq(ausdr, i=m..n)`: in den Ausdruck `ausdr` werden für i eingesetzt: $m, m+1, m+2, \dots, m+k < n$ mit $n < m+k+1$
 - `seq(ausdr, i=X)`: alle Werte der Liste oder Menge X werden in `ausdr` eingesetzt
 - Bindung der Variablen i wird nicht berücksichtigt!
 - `ausdr` kann eine Folge sein, geschrieben in der Form `seq(. .)`
 \Rightarrow `seq` läßt sich verschachteln

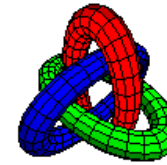
Folgen (2)



Einführung in die
mathematische
Software
WS 2002/03

- den $\$$ -Operator:
 - $\text{ausdr}\$i=m..n$ entspricht $\text{seq}(\text{ausdr}, i=m..n)$
 - i wird vollständig ausgewertet \Rightarrow i.a. $'\text{ausdr}'\$'i'=m..n$
 - $\text{ausdr}\$n$ entspricht n -mal der Ausdruck ausdr
 - $\$n..m$ entspricht $n, n+1, \dots, n+k \leq m$ mit $m < n+k+1$
 - Verschachtelung durch $'\text{ausdr}\$i=k..l'\$j=m..n$ möglich
 - Syntax des $\$$ -Operators kompakter, aber nicht so leistungsfähig wie der seq Befehl
- Beispiele:
 - $\text{seq}(i, i=1..10)$ oder $\$1..n \Rightarrow$ Zahlen 1,2,3,...10
 - $\text{seq}(\text{seq}(i*j, i=1..10), j=1..10);$ oder
 $'i*j\$i=1..10'\$j=1..10; \Rightarrow$ kleine $1*1$

Folgen (3)



Einführung in die
mathematische
Software
WS 2002/03

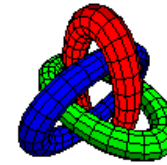
Zugriff auf Elemente einer Folge `flg`:

- einzelne Elemente `flg[n]`
- Teilfolge `flg[m..n]`
- `m, n` negativ \Rightarrow es wird am Ende der Folge begonnen zu zählen (z.B. ist `flg[-2]` das vorletzte Element)

Änderung einzelner Folgeelemente:

- `folge := op(subsop(n=x, [flg]))`
 - Umwandlung in Liste (`[flg]`)
 - dem `n`-ten Element wird der Wert `x` zugewiesen (`subsop`)
 - Umwandlung in Folge (`op`)
 - wenn `x=NULL` \Rightarrow Element wird gelöscht

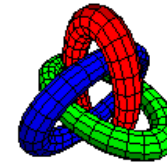
Listen



Einführung in die
mathematische
Software
WS 2002/03

- Folge zwischen eckigen Klammern z.B. $[x, y, z]$
- geordnet und können beliebig verschachtelt werden (im Unterschied zu Folgen)
- Zugriff auf Elemente wie bei Folgen
- Erweiterung: mit $lste := [a, b, op(lste), y, z]$ wird $lste$ am Anfang durch a, b und am Ende durch y, z erweitert
- Änderung von Elementen mit dem Befehl $subsop(n=x, m=y, \dots, lste)$; dabei wird in der Liste $lste$ dem n -te Element der Wert x , dem m -ten der Wert y zugeordnet; wenn ein Wert `NULL` ist, wird das Element gelöscht

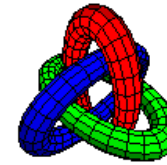
Mengen



Einführung in die
mathematische
Software
WS 2002/03

- Folge zwischen geschweiften Klammern z.B. $\{x, y, z\}$
- nicht geordnet (!), d.h. Maple verändert die Reihenfolge der Elemente, Mengen können verschachtelt werden
- doppelte Einträge werden gelöscht
- \Rightarrow Zugriff auf Elemente wie bei Folge, aber nicht immer sinnvoll (siehe oben)
- folgende Operationen sind für Mengen definiert:
 - $M \text{ union } N$ oder ``union`(M,N,O,...)`: Vereinigung
 - $M \text{ minus } N$ oder ``minus`(M,N)`: Differenz
 - $M \text{ intersect } N$ oder ``intersect`(M,N,O,...)`: Durchschnitt
 - $M \text{ subset } N$ oder ``subset`(M,N)`: Ist M eine Teilmenge von N ?

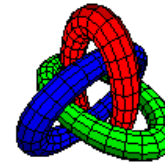
Kovertierungen



Einführung in die
mathematische
Software
WS 2002/03

- eine Folge `flg` umwandeln
 - in eine Liste: `[flg]`
 - in eine Menge: `{ flg }`
- eine Liste `lst` umwandeln
 - in eine Folge: `op(lst)`
 - in eine Menge: `convert(lst, set)` oder `{ op(lst) }`
- eine Menge `mng` umwandeln
 - in eine Folge: `op(mng)`
 - in eine Liste: `convert(mng, list)` oder `[op(mng)]`

Listen-, Mengenoperationen(1)

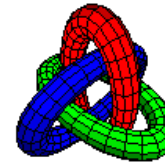


Einführung in die
mathematische
Software
WS 2002/03

Die folgenden Operatoren sind für Listen als auch für Mengen gültig: N ist stets eine Liste oder Menge

- `member(x, N)`: gibt an, ob x in N enthalten ist
- `select(f, N)` gibt nur jene Elemente von N aus, die bei der boolschen Funktion f den Wahrheitswert `true` liefern
- bei `select(f, N, x, y, ...)` werden $x, y, ...$ der Funktion f als Argumente übergeben; typische Beispiele:
 - $f = \text{type}$ und $x = \text{integer} \Rightarrow$ alle ganzen Zahlen aus N werden ausgegeben; siehe Hilfeseite `type`
 - $f = \text{has}$ und $x = a \Rightarrow$ Ausdrücke aus N mit der Variablen a werden ausgegeben
- `remove(...)` gibt die Komplementärmenge zu `select(...)` aus

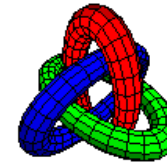
Listen-, Mengenoperationen(2)



Einführung in die
mathematische
Software
WS 2002/03

- $\text{map}(f, N)$ wendet die Funktion f auf alle Elemente von N an, f kann auch eine Liste von Funktionen sein
- **Beispiele:** $N := \{0, 1.0, 3*x, 4*x+5*y, 9\}$
 - $M := \text{select}(\text{type}, N, \text{integer}) \Rightarrow M = \{0, 9\}$
 - $\text{select}(\text{has}, N, x) \Rightarrow \{4x+5y, 3x\}$
 - $\text{remove}(\text{type}, N, \text{integer}) \Rightarrow \{1.0, 4x+5y, 3x\}$
 - $\text{map}(\text{sqrt}, M) \Rightarrow \{0, 3\}$

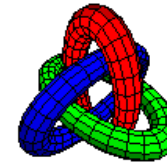
Ausdrücke



Einführung in die
mathematische
Software
WS 2002/03

- Maple zerlegt jeden mathematischen Ausdruck in einzelne Operanden z.B.
 - Summe in die Summanden
 - Produkt in die Faktoren
 - Listen, Mengen in die Elemente
 - etc.
- um herauszufinden von welchem Typ ein Ausdruck ist, gibt es die Funktion `whattype(ausdr)`
- mit `type(ausdr, dtype)` kann man überprüfen, ob ein Ausdruck von einem bestimmten Datentyp `dtype` ist
- die Befehle `select`, `remove`, `map`, `op` können auf beliebige Ausdrücke angewendet werden

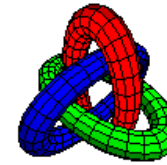
Substitution



Einführung in die
mathematische
Software
WS 2002/03

- Mit der Funktion $\text{subs}(x=t, \text{ausdr})$ ist es möglich Teile eines Ausdrucks ausdr zu ersetzen
- Maple zerlegt dazu ausdr rekursiv in seine Operanden und ersetzt dabei x durch t
- Probleme, wenn x selbst ein Ausdruck ist, z.B.
 - $\text{subs}(a+b=t, (a+b+c)^2) \Rightarrow (a+b+c)^2$
- subs können als erstes Argument auch eine Liste oder Menge von gewünschten Substitutionen übergeben werden
- die Substitution $\text{algsb}(x=t, \text{ausdr})$ stützt sich nicht auf die Zerlegung in Operanden \Rightarrow
 - $\text{algsb}(a+b=t, (a+b+c)^2)$ liefert korrektes Ergebnis $(t+c)^2$

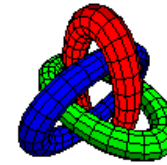
Vereinfachung – `simplify`



Einführung in die
mathematische
Software
WS 2002/03

- `simplify(ausdr)` versucht einen Ausdruck zu vereinfachen
- „einfach“ ist ein relativer Begriff; je nach Verwendungszweck ist ein Ausdruck „einfach“
- die von `simplify` verwendeten Umwandlungsregeln können eingeschränkt werden: `simplify(ausdr, regel)`
 - `regel=trig` => nur trig. Umwandlungen werden durchgeführt
 - `regel=power` => Potenzregeln
 - `regel=sqrt` => Umwandlungen mit Quadratwurzeln
 - `regel=radical` => Umwandlungen mit allgemeine Wurzel ausdr.
 - Hilfeseiten unter `?simplify/regel` z.B. `?simplify/sqrt`

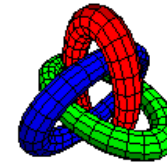
Nebenbedingungen



Einführung in die
mathematische
Software
WS 2002/03

- Maple setzt voraus, daß unbestimmte Variablen innerhalb des komplexen Zahlenbereichs sind \Rightarrow viele Ausdrücke können nicht weiter vereinfacht werden oder der vereinfachte Ausdruck ist komplizierter als notwendig
- deshalb ist es sinnvoll Nebenbedingungen für Variablen anzugeben:
`simplify(ausdr, assume=eig)`: alle Variablen haben die Eigenschaft `eig`:
 - `integer`, `odd`, `even`, `posint` (ganze Zahlen)
 - `real`, `positive`, `negative`, `nonneg`, `ration` (reelle, rationale Zahlen)
 - Liste aller möglichen Eigenschaften: `?property`
- Nebenbed. für einzelne Variable: `assume(x, eig)` \Rightarrow gilt für alle folgenden Berechnungen; statt `x`, wird ab diesem Zeitpunkt `x~` geschrieben (`~ ...` für diese Variable gilt eine Nebenbedingung)

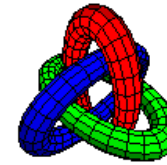
expand – factor – combine (1)



Einführung in die
mathematische
Software
WS 2002/03

- die primäre Anwendung von `expand(ausdr)` ist Produkte in Summen umzuwandeln
- bei trigonometrischen bzw. logarithmischen Funktionen wird der Funktionoperand auf eine möglichst einfache Form gebracht
- `factor` und `combine` bilden das Gegenstück zu `expand`
 - `factor(ausdr)` versucht `ausdr` zu faktorisieren
 - `factor` berücksichtigt nur Wurzeln, die im Zahlenbereich der Koeffizienten liegen z.B. $x^2 - 5$ kann nicht faktorisiert werden, aber $x^2 - 5.0$ schon
 - `factor` ist insbesondere für Polynome und rationale Funktionen gedacht

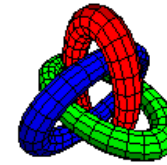
expand – factor – combine (2)



Einführung in die
mathematische
Software
WS 2002/03

- mit `combine(ausdr)` können allgemeine Ausdrücke zusammengefaßt werden
- manchmal ist es notwendig `combine` anzugeben, welche Regeln zum Vereinfachen verwendet werden sollen:
`combine(ausdr, rgl)`, dabei kann `rgl` auch eine Liste von Regeln sein z.B.
 - `ln` und `exp` für Logarithmen und Exponentialfunktionen
 - `trig` für trigonometrische Funktionen
 - `power` für Potenzen

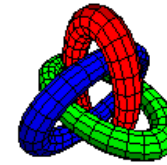
sort – collect (1)



Einführung in die
mathematische
Software
WS 2002/03

- Summen mit vielen Summanden übersichtlicher anschreiben durch Sortieren nach der Ordnung bzw. durch Herausheben von gemeinsamen Variablen
- `sort(f)` sortiert den Ausdruck f nach der Summe der Potenzen:
z.B. `sort(y+x^2*y +x^2+1)` $\Rightarrow x^2*y+x^2+y+1$
- mit `sort(f,x)` wird nach den Potenzen von x sortiert, und bei `sort(f,[x,y...])` nach den Potenz von x , danach nach den Potenzen von y
- x kann auch eine Funktion sein z.B. trigonometrische oder Exponentialfunktion
- auch Listen können mit `sort` geordnet werden
- eine Anordnung ist nicht immer möglich!!

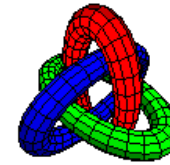
sort – collect (2)



Einführung in die
mathematische
Software
WS 2002/03

- `collect(f, x)` hebt aus allen Summanden von f die Variable x in der höchsten Potenz heraus die möglich ist und faßt solche Summanden zusammen z.B.:
$$\text{collect}(y+x^2*y+x^2+1+2*x, x) \Rightarrow (y+1)x^2++2*x+y+1$$
- mit `collect(f, [x, y, ...])` wird zuerst x herausgehoben, dann y , ...
- auch bei `collect` können für x, y, \dots Funktionen eingesetzt werden

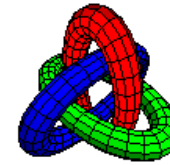
Lösen von Gleichungen (1)



Einführung in die
mathematische
Software
WS 2002/03

- Gleichungen unter Maple: `ausdr1 = ausdr2` (im Unterschied zum Bindungsoperator `:=`)
- man kann durch `lhs(glg)` (lefthandside) bzw. `rhs(glg)` (righthandside) auf den Ausdruck auf der linken bzw. rechten Seite des „`=`“ zugreifen
- Standardbefehl um Gleichungen zu lösen: `solve(glg, x)`; d.h. es wird die Gleichung `glg` nach der Variablen `x` gelöst
- `x` kann weggelassen werden, wenn nur eine Variable auftritt
- `solve(fkt)` berechnet die Nullstellen der Funktion `fkt`

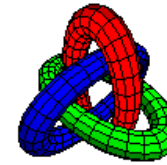
Lösen von Gleichungen (2)



Einführung in die
mathematische
Software
WS 2002/03

- `solve` versucht stets alle Lösungen symbolisch zu berechnen (daher exakt)
- mit `solve({g1g1, g1g2, ...}, {x, y, ...})` können Gleichungssysteme gelöst werden
- die Angabe der Variablen $\{x, y, \dots\}$ kann weggelassen werden, wenn die Anzahl der Gleichungen gleich der Anzahl der Variablen ist
- auch viele nicht-lineare Gleichungen bzw. Gleichungssysteme können gelöst werden

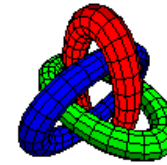
RootOf – allvalues



Einführung in die
mathematische
Software
WS 2002/03

- Manchmal treten in den Lösungen von Gleichungen Ausdrücke der Art `RootOf(. .)` auf
z.B. `RootOf(5*_Z^2-1, . . .)`; steht für alle Lösungen der Gleichung $5 \cdot x^2 - 1$;
- Maple behandelt `RootOf`-Ausdrücke wie Variablen
- durch `allvalues(ausdr)` werden alle Lösungen des `RootOf` Ausdrucks berücksichtigt und eingesetzt
- wenn `allvalues` die `RootOf`-Ausdrücke symbolisch nicht auflösen kann, so werden sie numerisch berechnet und wieder als `RootOf` Ausdruck mit einem angehängten Label ausgegeben
=> der Ausdruck kann mit `evalf` numerisch berechnet werden
z.B.: `RootOf(_Z^6+_Z+1, index = 1)`

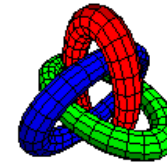
periodische Lösungen



Einführung in die
mathematische
Software
WS 2002/03

- beim Lösen von z.B. trigonometrischen Gleichungen können periodisch Lösungen auftreten
- um periodische Lösungen darzustellen verwendet Maple folgende Variablennamen:
 - `_Z` für Variablen in den ganzen Zahlen
 - `_NN` für nicht-negative ganze Zahlen
 - `_B` für binäre Werte (nur 0 und 1)
- Damit Maple periodische Lösungen ausgibt muß die Systemvariable `_EnvAllSolutions:=true` gesetzt werden
- Beispiel: `solve(sin(x)=sin(2*x),x)`

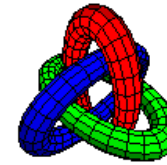
Glgcn numerisch lösen



Einführung in die
mathematische
Software
WS 2002/03

- `fsolve` löst Gleichungen + Gleichungssysteme numerisch
- Syntax von `fsolve` entspricht der Syntax von `solve`
- i.a. wird nur eine Lösung berechnet
- Ausnahme: bei Polynomen werden alle reellen Lösungen berechnet; mit der zusätzlichen Option `complex` werden auch die komplexen Lösungen berechnet: `fsolve(glg,x,complex)`
- wenn `fsolve` keine gewünschten Lösungen liefert, so kann ein Zahlenbereich vorgegeben werden, in dem gesucht werden soll: `fsolve(glg, x, x=m..n)`

Lösungen weiterverwenden



Einführung in die
mathematische
Software
WS 2002/03

- wenn die Lösungen dauerhaft an Variablen gebunden werden sollen \Rightarrow `assign` verwenden
- vorübergehende Bindungen, Lösungen in andere Ausdrücke einsetzen mit
 - `subs` (siehe Substitution)
 - `eval(ausdr, x=n)`: in `ausdr` wird für `x` der Wert `n` eingesetzt z.B. `eval(x^2+x, x=5)`;
als zweites Argument kann man `eval` auch eine Liste oder Menge von Gleichungen übergeben z.B. `{x=3, y=6, ...}`