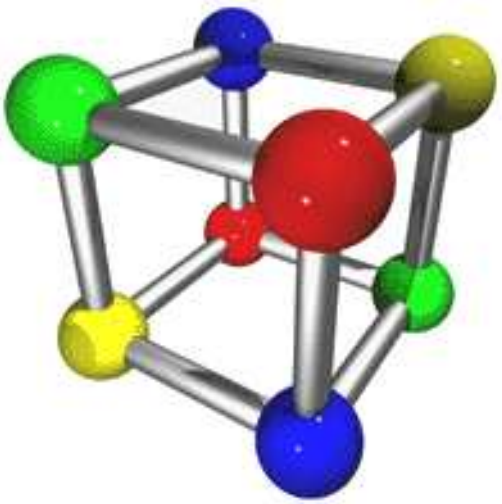
MuPAD 1.4
im Überblick

W. Oevel



MuPAD

The Open Computer Algebra System

Strukturen in MuPAD 1.4: eine Übersicht

Es wird eine Übersicht über die in MuPAD Version 1.4 installierten Datenstrukturen, Konstanten, Funktionen und Bibliotheken angegeben. Genauere Informationen sind im Handbuch

The MuPAD Group. *MuPAD User's Manual.* Wiley–Teubner, 1996.

oder im Tutorium

W. Oevel, F. Postel, G. Rüscher und S. Wehmeier.
Das MuPAD–Tutorium. Springer, 1999.

zu finden bzw. durch das Hilfesystem zu erhalten.

Inhaltsverzeichnis

1 Informationen und Hilfe 2
2 Umgebungsvariablen 2
3 Vordefinierte Datentypen (Domains) 2
4 Erzeugung spezieller Datenstrukturen 3
5 Operatoren 4
6 Arithmetikfunktionen 5
7 Mathematische Konstanten 6
8 Spezielle mathematische Funktionen 6
9 Funktionen zur Behandlung von Polynomen 7
10 Umgang mit MuPAD–Objekten 8
11 Manipulation von Zeichenketten 9
12 Funktionen für Ein- und Ausgabe, Graphik 10
13 Steuerung der Auswertung 10
14 Funktionen in Prozedurdefinitionen 10
15 Mathematische Funktionen und Algorithmen 11
16 Die Bibliotheken 12
17 Kontrolle eigener Prozeduren 13
18 Technisches 13

1 Informationen und Hilfe

`help`, `?`	:	Anfordern von Hilfeseiten
`info`	:	Ausgabe von Kurzinformationen
`setuserinfo`	:	Informationen beim Ablauf von Algorithmen

2 Umgebungsvariablen

Umgebungsvariablen sind globale Variablen, die den Ablauf der in den MuPAD–Prozeduren implementierten Algorithmen beeinflussen. Die voreingestellten Werte können vom Nutzer im Laufe einer Sitzung abgeändert werden.

	Voreinstellung	
`DIGITS`	10	signifikante Dezimalstellen in Gleitkomma–rechnungen
`EVAL_STMT`	TRUE	steuert Zuweisungen innerhalb von Ausdrücken
`HISTORY`	[20, 3]	Anzahl der mit `last` zugreifbaren Ergebnisse
`LEVEL`	100	Auswertungstiefe
`LIB_PATH`		Pfadname des MuPAD–Programms und der Bibliotheken
`MAXDEPTH`	500	maximal zulässige Rekursionstiefe
`MAXLEVEL`	100	maximal zulässige Auswertungstiefe
`PRETTY_PRINT`	TRUE	2–dimensionale Ausgabe?
`PRINTLEVEL`	0	steuert die Bildschirmausgabe innerhalb von Schleifen, Prozeduren, etc.
`READ_PATH`		Pfadname für mit `read` einzulesende Dateien
`ROUNDING`	ToNearest	Rundungsmodus bei Gleitkommaoperationen
`TEXTWIDTH`	75	Textbreite der Ausgabe
`WRITE_PATH`		Pfadname für mit `write` etc. abzuspeichernde Dateien

3 Vordefinierte Datentypen (Domains)

Datentypen des MuPAD–Kerns:

`DOM_ARRAY`	:	Felder
`DOM_BOOL`	:	die logischen Werte `TRUE`, `FALSE` und `UNKNOWN`
`DOM_COMPLEX`	:	komplexe Zahlen
`DOM_DOMAIN`	:	Datenstrukturen
`DOM_EXEC`	:	einfache Funktionen
`DOM_EXPR`	:	Ausdrücke
`DOM_FAIL`	:	das Objekt `FAIL`
`DOM_FLOAT`	:	reelle Gleitkommazahlen
`DOM_FUNC_ENV`	:	Funktionsumgebungen
`DOM_IDENT`	:	Bezeichner
`DOM_INT`	:	ganze Zahlen

`DOM_LIST`	:	Listen
`DOM_NIL`	:	das Objekt `NIL`
`DOM_NULL`	:	das leere Objekt `null()`
`DOM_POINT`	:	Punkte (als graphische Primitive)
`DOM_POLY`	:	Polynome
`DOM_POLYGON`	:	Polygone (als graphische Primitive)
`DOM_PROC`	:	Prozeduren
`DOM_RAT`	:	rationale Zahlen
`DOM_SET`	:	Mengen
`DOM_STRING`	:	Zeichenketten
`DOM_TABLE`	:	Tabellen

In der MuPAD–Sprache definierte Datentypen der Standardbibliothek:

`CF`	:	Kettenbrüche
`gseries`	:	verallgemeinerte Reihenentwicklungen
`NUMERIC`	:	Vereinigung von `DOM_COMPLEX`, `DOM_FLOAT`, `DOM_INT` und `DOM_RAT`
`O`	:	Ordnungsterm von Reihenentwicklungen
`ode`	:	gewöhnliche Differentialgleichungen
`Poly(..)`	:	Polynomringe
`Puiseux`	:	Reihenentwicklungen
`rec`	:	Rekurrenzgleichungen
`rectform`	:	Kartesische Darstellung komplexer Zahlen
`RootOf`	:	alle Nullstellen eines Polynoms

Viele weitere Datenstrukturen sind mit ihren Erzeugern in der Bibliothek `Dom` installiert (Informationen mittels `info(Dom)`).

4 Erzeugung spezieller Datenstrukturen

`array`	:	erzeugt ein Feld (`DOM_ARRAY`)
`arrow`	:	Funktionsdefinition (erzeugt `DOM_EXEC`)
`asympt`	:	asymptotische Entwicklung (erzeugt `gseries`)
`contfrac`	:	Kettenbruchentwicklung (erzeugt `CF`)
`domain`	:	Erzeuger eines Domains
`domattr`	:	Setzen oder Lesen eines Domain–Attributs
`fun`, `func`	:	Funktionsdefinition (erzeugen `DOM_EXEC`)
`func_env`	:	Definition einer Funktionsumgebung (erzeugt `DOM_FUNC_ENV`)
`funcattr`	:	Setzen oder Lesen eines Attributs einer Funktionsumgebung
`new`	:	Erzeuger eines Domain–Elementes
`null`	:	erzeugt das „leere Objekt“ (`DOM_NULL`)
`O`	:	erzeugt den Ordnungsterm von Reihenentwicklungen
`ode`	:	erzeugt eine gewöhnliche Differentialgleichung
`point`	:	erzeugt die Datenstruktur `DOM_POINT` zum Zeichnen von Punkten
`poly`	:	erzeugt ein Polynom (`DOM_POLY`)
`polygon`	:	erzeugt die Datenstruktur `DOM_POLYGON` zum Zeichnen von Polygonen

`rec`	:	erzeugt eine Rekurrenzgleichung
`rectform`	:	Kartesische Darstellung komplexer Zahlen (erzeugt `rectform`)
`RootOf`	:	alle Nullstellen eines Polynoms (erzeugt `RootOf`)
`series`	:	allgemeine Reihenentwicklung (erzeugt `Puiseux` oder `gseries`)
`taylor`	:	Taylor–Entwicklung (erzeugt `Puiseux`)
`table`	:	erzeugt eine Tabelle (`DOM_TABLE`)

Viele Erzeuger weiterer Datenstrukturen sind in der Bibliothek `Dom` installiert, z.B.:

`Dom::IntegerMod(p)`	:	Erzeuger von ganzen Zahlen modulo p
`Dom::Matrix()`	:	Erzeuger von Matrizen

5 Operatoren

Mit folgenden Operatoren können die angegebenen Systemfunktionen in intuitiver Form aufgerufen werden, z.B., `a+b` statt `_plus(a,b)`, `x<1` statt `_less(x,1)`, usw.:

Operator	Bindungs-stärke	System-funktion	Bedeutung
`'`	1900	`D`	Differentialoperator
`.`	1700	`_concat`	Konkatenation
`@`	1500	`_fconcat`	Komposition von Funktionen
`not`	1400	`_not`	logische Verneinung
`@@`		`repcom`	Iteration
`!`	1300	`fact`	Fakultät
`^`	1200	`_power`	Potenzieren
`*`	1100	`_mult`	Multiplikation
`/`	1100	`_divide`	Division
`+`	1000	`_plus`	Addition
`-`	1000	`_subtract`	Subtraktion
`div`	900	`_div`	Quotient „modulo“
`mod`	900	`_mod`	Rest „modulo“
`intersect`	800	`_intersect`	Schnitt von Mengen
`minus`	700	`_minus`	Differenz von Mengen
`union`	600	`_union`	Vereinigung von Mengen
`..`	500	`_range`	Bereich
`=`	400	`_equal`	Gleichung
`<>`	400	`_unequal`	Ungleichung
`<` und `>`	400	`_less`	Größenvergleiche
`<=` und `>=`	400	`_leequal`	Größenvergleiche
`$`	300	`_seqgen`	Folgengenerator
`and`	200	`_and`	logisches „und“
`or`	100	`_or`	logisches „oder“
`,`		`_exprseq`	Erzeugen von Folgen
`;` und `:`		`_stmtseq`	Trennung zwischen Befehlen

6 Arithmetikfunktionen

Die folgenden Funktionen können auch mittels Operatoren oder Schlüsselwörtern aufgerufen werden, z.B., `a+b` statt `_plus(a,b)`, `-a` statt `_negate(a)`, etc.:

`_and`	:	logisches „und“
`_assign`	:	Zuweisungsfunktion
`_break`	:	Unterbrechung von Schleifen etc.
`_case`	:	Verzweigung
`_concat`	:	Konkatenation
`_div`	:	Quotient „modulo“
`_divide`	:	Division
`_equal`	:	Gleichung
`_exprseq`	:	Folgengenerator
`_fconcat`	:	Hintereinanderschaltung von Funktionen
`_for`	:	Schleife (aufwärts)
`_for_down`	:	Schleife (abwärts)
`_for_in`	:	Schleife (durchläuft die Operanden eines Ausdrucks)
`_if`	:	Verzweigung
`_index`	:	indizierter Aufruf
`_intersect`	:	Schnitt von Mengen
`_invert`	:	Invertierung
`_leequal`	:	Ungleichung $\leq$
`_less`	:	Ungleichung $<$
`_minus`	:	Differenz von Mengen
`_mod`	:	Rest „modulo“
`_mult`	:	Multiplikation
`_negate`	:	Multiplikation mit -1
`_next`	:	Sprung in einer Schleife
`_not`	:	logische Verneinung
`_or`	:	logisches „oder“
`_plus`	:	Addition
`_power`	:	Potenzierung
`_procdef`	:	Prozedurdefinition
`_quit`	:	Ende einer MuPAD–Sitzung
`_range`	:	Bereich
`_repeat`	:	Schleife
`_seqgen`	:	Folgengenerator
`_seqin`	:	Folgengenerator
`_stmtseq`	:	Folge von Befehlen
`_subtract`	:	Differenz
`_unequal`	:	Ungleichung <>
`_union`	:	Vereinigung von Mengen
`_while`	:	Schleife
`repcom`	:	Iteration von Funktionen („repeated composition“)

7 Mathematische Konstanten

`catalan()`	:	Catalan–Konstante: `float(catalan())=0.9159655941..`
`complexInfinity`	:	der unendlich ferne Punkt der komplexen Ebene
`E`	:	Eulersche Zahl $\exp(1) = 2.718281828..$
`EULER`	:	Eulersche Konstante $\gamma = 0.5772156649..$
`I`	:	imaginäre Einheit $\sqrt{-1}$
`infinity`	:	Unendlich
`PI`	:	$\pi = 3.141592653..$
`undefined`	:	nicht definierter Wert

8 Spezielle mathematische Funktionen

Folgende mathematischen Funktionen sind in MuPAD Version 1.4 implementiert, d.h., sie sind Systemfunktionen wie `expand`, `float`, `simplify`, usw. bekannt und werden von ihnen verarbeitet:

`acos`	:	Umkehrfunktion von `cos`
`acosh`	:	Umkehrfunktion von `cosh`
`acot`	:	Umkehrfunktion von `cot`
`acoth`	:	Umkehrfunktion von `coth`
`acsc`	:	Umkehrfunktion von `csc`
`acsch`	:	Umkehrfunktion von `csch`
`asec`	:	Umkehrfunktion von `sec`
`asech`	:	Umkehrfunktion von `sech`
`asin`	:	Umkehrfunktion von `sin`
`asinh`	:	Umkehrfunktion von `sinh`
`atan`	:	Umkehrfunktion von `tan`
`atanh`	:	Umkehrfunktion von `tanh`
`bernoulli`	:	Bernoulli–Zahlen und –Polynome
`besselJ`	:	Bessel–Funktion erster Art
`besselY`	:	Bessel–Funktion zweiter Art
`beta`	:	β–Funktion
`cos`	:	Cosinus–Funktion
`cosh`	:	Cosinus Hyperbolicus–Funktion
`cot`	:	Cotangens–Funktion
`coth`	:	Cotangens Hyperbolicus–Funktion
`csc`	:	$1/\sin(x)$
`csch`	:	$1/\sinh(x)$
`dilog`	:	Dilogarithmus–Funktion
`dirac`	:	Diracsche δ–Funktion
`eint`	:	$\int_x^\infty e^{-t}/t\,dt$
`erf`	:	$(2/\sqrt{\pi})\ \int_0^x e^{-t^2}\,dt$
`erfc`	:	$1 - \mathrm{erf}(x)$
`exp`	:	Exponentialfunktion
`fact`	:	Fakultätsfunktion
`gamma`	:	Γ–Funktion
`heaviside`	:	Sprungfunktion

`id`	:	identische Abbildung $x \to x$
`igamma`	:	unvollständige Γ–Funktion
`ln`	:	natürlicher Logarithmus
`phi`	:	Eulersche ϕ–Funktion
`psi`	:	Polygamma–Funktion
`sec`	:	$1/\cos(x)$
`sech`	:	$1/\cosh(x)$
`Si`	:	$\int_0^x \sin(t)/t\, dt$
`sin`	:	Sinus–Funktion
`sinh`	:	Sinus Hyperbolicus–Funktion
`sqrt`	:	Wurzelfunktion
`tan`	:	Tangens–Funktion
`tanh`	:	Tangens Hyperbolicus–Funktion
`zeta`	:	Riemannsche ζ–Funktion

9 Funktionen zur Behandlung von Polynomen

Folgende Funktionen verarbeiten mittels `poly` erzeugte Polynome vom Domain–Typ `DOM_POLY` und teilweise auch polynomiale Ausdrücke vom Domain–Typ `DOM_EXPR`:

`coeff`	:	Koeffizienten
`collect`	:	Zusammenfassen von Koeffizienten
`content`	:	größter gemeinsamer Teiler aller Koeffizienten
`decompose`	:	Zerlegen als Komposition von Polynomen
`degree`	:	Polynomgrad
`degreevec`	:	Exponenten des führenden Terms
`diff`, `D`, `Dpoly`	:	Differentiation
`discrim`	:	Diskriminante
`divide`	:	Division mit Rest
`evalp`	:	Auswertung an einem Punkt
`expr`	:	Konvertierung in einen Ausdruck
`Factor`, `factor`	:	Faktorisierung
`gcd`	:	größter gemeinsamer Teiler
`gcdex`	:	größter gemeinsamer Teiler (erweiterter Euklidischer Algorithmus)
`genpoly`	:	Polynomerzeugung mittels b–adischer Entwicklung
`icontent`	:	größter gemeinsamer Teiler aller Koeffizienten
`irreducible`	:	Test auf Irreduzibilität
`iszero`	:	Test auf 0
`lcm`	:	kleinstes gemeinsames Vielfaches
`lcoeff`	:	führender Koeffizient
`ldegree`	:	niedrigster Grad innerhalb eines Polynoms
`lmonomial`	:	führendes Monom
`lterm`	:	führender Term
`mapcoeffs`	:	Anwendung einer Funktion auf die Koeffizienten
`multcoeffs`	:	Multiplikation mit einem Skalar

`norm`	:	Norm
`nterms`	:	Anzahl der Terme
`nthcoeff`	:	n–ter Koeffizient
`nthmonomial`	:	n–tes Monom
`nthterm`	:	n–ter Term
`pdivide`	:	Pseudodivision
`Poly`	:	Erzeuger von Polynomringen
`poly`	:	Erzeuger eines Polynoms
`poly2list`	:	Konvertierung in eine Liste
`primpart`	:	primitiver Teil
`randpoly`	:	Erzeuger eines zufälligen Polynoms
`resultant`	:	Resultante
`sqrfree`	:	quadratfreie Faktorisierung
`tcoeff`	:	niedrigster Koeffizient

10 Umgang mit MuPAD–Objekten

`alias`	:	Definition von Abkürzungen
`anames`	:	Auflistung aller Bezeichner, die einen Wert haben
`append`	:	Anfügen an eine Liste
`assign`	:	Zuweisungsfunktion
`assign_elems`	:	Zuweisungsfunktion für Felder, Listen und Tabellen
`assume`	:	Setzen von Annahmen über Eigenschaften von Bezeichnern
`bin_op`	:	Definition eines binären Operators
`block`	:	Block von Befehlen mit lokalen Variablen
`bool`	:	Auswertung zu `TRUE` oder `FALSE`
`collect`	:	Zusammenfassen von Koeffizienten polynomialer Ausdrücke
`combine`	:	Zusammenfassen von Teilausdrücken
`contains`	:	Test auf Elemente in Listen, Mengen und Tabellen
`denom`	:	Nenner eines rationalen Ausdrucks
`domattr`	:	Setzen oder Lesen eines Domain–Attributs
`domtype`	:	Domain–Typ
`expand`	:	Expansion von Ausdrücken
`export`	:	Einladen von Bibliotheken
`expose`	:	Ausgabe der Operanden von Funktionsumgebungen und Domains
`expr`	:	Konvertierung diverser Datenstrukturen in Ausdrücke, Felder, etc.
`extnops`	:	Anzahl der Operanden eines Domains
`extop`	:	die Operanden eines Domains
`extsubsop`	:	Substitution von Operanden eines Domains
`funcattr`	:	Setzen oder Lesen eines Attributs einer Funktionsumgebung
`genident`	:	Erzeuger eines unbenutzten Bezeichners
`getprop`	:	Abfrage von Eigenschaften
`has`	:	Test auf Teilausdrücke

`hastype`	:	Test auf Teilausdrücke eines gegebenen Typs
`history`	:	die History–Tabelle einer MuPAD–Sitzung
`indets`	:	die Unbestimmten eines Ausdrucks
`insert_ordered`	:	Einsortieren in eine Liste
`is`	:	Abfrage von Eigenschaften eines Bezeichners
`length`	:	Komplexität eines Objekts
`linsert`	:	Einsetzen in eine Liste
`map`	:	Anwendung einer Funktion auf die Operanden eines Objekts
`maprat`	:	Anwendung einer Funktion auf einen „rationalisierten“ Ausdruck
`match`	:	Mustererkennung in Ausdrücken
`nops`	:	Anzahl der Operanden eines Objektes
`numer`	:	Zähler eines rationalen Ausdrucks
`op`	:	Operandenfunktion
`protect`	:	Setzen eines Schreibschutzes
`radsimp`	:	Vereinfachung von Wurzelausdrücken
`rationalize`	:	Konvertierung beliebiger Ausdrücke in rationale Ausdrücke
`rewrite`	:	Umschreiben von Ausdrücken
`select`	:	Auswählen von Operanden nach Eigenschaften
`seq`	:	Folgengenerator
`simplify`	:	Vereinfacher
`split`	:	Zerlegen eines Objektes nach Eigenschaften
`subs`	:	Substitution
`subsex`	:	Substitution komplexerer Ausdrücke
`subsop`	:	Substitution von Operanden
`sysassign`	:	Zuweisung (ignoriert Schreibschutz)
`sysorder`	:	Information über die Anordnung von Objekten im MuPAD–Kern
`testtype`	:	Typenvergleich
`traperror`	:	Abfangen eines Fehlers
`type`	:	Typ eines Objektes
`unalias`	:	Löschen einer Abkürzung
`unassign`	:	Löschen des Wertes eines Bezeichners
`unassume`	:	Löschen von Annahmen über Eigenschaften
`unprotect`	:	Aufheben des Schreibschutzes für Bezeichner
`zip`	:	Verknüpfung von Listen und Matrizen

11 Manipulation von Zeichenketten

`_concat`, `.`	:	Aneinanderfügen mehrerer Zeichenketten
`expr2text`	:	Konvertierung eines Ausdrucks in eine Zeichenkette
`ftextinput`	:	Texteingabe aus einer Datei
`revert`	:	Umkehrfunktion
`strlen`	:	Länge einer Zeichenkette
`strmatch`	:	Mustererkennung
`substring`	:	Extrahieren von Teilen

`tbl2text`	:	Konvertierung einer Tabelle in eine Zeichenkette
`text2expr`	:	Konvertierung einer Zeichenkette in einen Ausdruck
`text2list`	:	Konvertierung einer Zeichenkette in eine Liste
`text2tbl`	:	Konvertierung einer Zeichenkette in eine Tabelle
`textinput`	:	Interaktive Eingabe von Zeichenketten

Weitere Funktionen sind in der Bibliothek `string` installiert.

12 Funktionen für Ein- und Ausgabe, Graphik

`error`	:	Ausgabe einer Fehlermeldung
`fclose`	:	Schließen einer Datei
`finput`	:	Lesen einer Datei
`fopen`	:	Öffnen einer Datei
`fprint`	:	Beschreiben einer Datei
`fread`	:	Lesen einer Datei
`ftextinput`	:	Texteingabe aus einer Datei
`input`	:	Interaktive Zuweisung
`plot2d`	:	2–dimensionale Zeichnungen
`plot3d`	:	3–dimensionale Zeichnungen
`plotfunc`	:	Zeichnen von Funktionsgraphen
`print`	:	Bildschirmausgabe
`protocol`	:	Eröffnen und Schließen eines Sitzungsprotokolls
`read`	:	Einlesen einer Datei
`setuserinfo`	:	Setzen des „Informationslevels“ für `userinfo`
`textinput`	:	Interaktive Eingabe von Zeichenketten
`userinfo`	:	Ausgaben von Informationen beim Ablauf eines Algorithmus
`warning`	:	Ausgabe einer Warnung
`write`	:	Abspeichern von Werten in Dateien

13 Steuerung der Auswertung

`bool`	:	Auswertung logischer Ausdrücke zu `TRUE` oder `FALSE`
`context`	:	Auswertung in einem Kontext
`eval`	:	erzwungene Auswertung
`evalassign`	:	Zuweisung mit Auswertung der linken Seite
`hold`	:	Verzögerung der Auswertung
`index_val`	:	indizierter Zugriff ohne Auswertung
`last`, `%`	:	Zugriff auf bereits berechnete Ergebnisse
`level`	:	Auswertung mit angegebener Tiefe

14 Funktionen in Prozedurdefinitionen

`_procdef`	:	Prozedurdefinition
`args`	:	Zugriff auf Prozedurargumente

`error` : Ausgabe einer Fehlermeldung
`return` : Rückgabe von Werten
`testargs` : Kontrolle von Tests
`warning` : Ausgabe einer Warnung

15 Mathematische Funktionen und Algorithmen

`abs` : Absolutbetrag
`asympt` : asymptotische Reihenentwicklung
`binomial` : Binomialausdruck $\binom{m}{n}$
`ceil` : kleinste ganze Zahl $\geq x$
`changevar` : Variablensubstitution in Integralen
`conjugate` : komplexe Konjugation
`contfrac` : Kettenbruchentwicklung
`D` und `'` : Differentialoperator
`diff` : Differentiation von Ausdrücken
`discont` : Unstetigkeitsstellen einer Funktion
`div` : Division „modulo“
`Factor` : Faktorisierung von Polynomen, Ausdrücken und ganzen Zahlen
`fft` : Schnelle Fourier–Transformation
`float` : Konvertierung in Gleitkommazahlen
`floor` : Gaußklammer: größte ganze Zahl $\leq x$
`frac` : Nachkommastellen einer Zahl
`gcd` : größter gemeinsamer Teiler
`gcdex` : größter gemeinsamer Teiler (erweiterter Euklidischer Algorithmus)
`ifactor` : Primfaktorzerlegung
`ifft` : inverse Schnelle Fourier–Transformation
`igcd` : größter gemeinsamer Teiler ganzer Zahlen
`igcdex` : größter gemeinsamer Teiler ganzer Zahlen (erweitert)
`ilcm` : kleinstes gemeinsames Vielfaches ganzer Zahlen
`Im` : Imaginärteil
`int` : Integration
`intersect` : Schnitt zweier Mengen
`isprime` : Primzahltest
`isqrt` : ganzzahlige Approximation der Wurzel einer natürlichen Zahl
`iszero` : Test auf 0
`ithprime` : i–te Primzahl
`lagrange` : Polynominterpolation
`lcm` : kleinstes gemeinsames Vielfaches
`limit` : Grenzwertberechnung
`linsolve` : Lösung linearer Gleichungen
`lllint` : reduzierte Gitterbasis
`max` : Maximum

`min`	:	Minimum
`minus`	:	Differenz zweier Mengen
`mod`, `modp`, `mods`	:	Rest „modulo“
`nextprime`	:	nächste Primzahl $\geq x$
`norm`	:	Norm von Polynomen, Vektoren und Matrizen
`normal`	:	Normalisierung rationaler Ausdrücke
`not`	:	logische Verneinung
`or`	:	logisches „und“
`partfrac`	:	Partialbruchzerlegung
`pdioe`	:	Lösung diophantischer Gleichungen
`powermod`	:	Potenz „modulo“
`product`	:	Erzeuger eines Produktes
`random`	:	Zufallszahlengenerator
`Re`	:	Realteil
`rectform`	:	Zerlegung nach Real- und Imaginärteil
`revert`	:	Umkehrfunktion (für Reihen, Zeichenketten und Listen)
`round`	:	Rundung
`series`	:	Reihenentwickler
`sign`	:	(komplexes) Vorzeichen
`solve`	:	Gleichungslöser
`sort`	:	Sortieren von Listen
`spline`	:	kubische Spline–Interpolierende
`sqrfree`	:	quadratfreie Faktorisierung von Polynomen und Ausdrücken
`sum`	:	Erzeuger einer Summe
`taylor`	:	Taylor–Entwicklung
`trunc`	:	Abschneiden von Nachkommastellen
`union`	:	Vereinigung zweier Mengen

Zahlreiche weitere Funktionen zu speziellen mathematischen Themenkreisen sind in den Bibliotheken (Abschnitt 16) installiert.

16 Die Bibliotheken

Folgende Bibliotheken in MuPAD Version 1.4 enthalten eine große Anzahl weiterer Funktionen und Algorithmen zu speziellen mathematischen Themenkreisen. Die Bibliotheken befinden sich in ständiger Entwicklung: spätere MuPAD–Versionen stellen zusätzliche Bibliotheken und Funktionen bereit. Eine Übersicht über die aktuell installierten Funktionen erhält man jeweils mit `info` oder `?`. Hilfe zu den einzelnen Algorithmen ist in der Form `?Bibliotheksname::Funktionsname` anzufordern.

`Ax`	:	Erzeuger von Axiomen
`Cat`	:	Erzeuger von Kategorien
`combinat`	:	Kombinatorik
`Dom`	:	vorgefertigte Datenstrukturen: Körper, Ringe, Matrizen, etc.
`faclib`	:	Faktorisierung
`fp`	:	Funktionales Programmieren
`gcdlib`	:	Berechnung größter gemeinsamer Teiler

`generate`	:	Erzeuger von C–, Fortran– und TeX–Code
`groebner`	:	Berechnung mit Polynomidealen
`io`	:	Ein- und Ausgabe
`linalg`	:	Lineare Algebra
`linopt`	:	Lineare Optimierung
`misc`	:	Diverses
`module`	:	Verwaltung von Modulen
`Network`	:	Graphen
`numeric`	:	numerische Algorithmen
`numlib`	:	Zahlentheorie
`orthpoly`	:	Orthogonalpolynome
`plotlib`	:	Zeichenroutinen
`Pref`	:	Systemvoreinstellungen
`property`	:	Eigenschaften von Bezeichnern
`RGB`	:	Farbnamen (Rot–Grün–Blau–Werte)
`sharelib`	:	Diverses
`stats`	:	Statistikfunktionen
`stdlib`	:	Standardbibliothek
`string`	:	Manipulation von Zeichenketten
`student`	:	einige elementare Algorithmen
`transform`	:	Integraltransformationen
`Type`	:	Typenbezeichner

17 Kontrolle eigener Prozeduren

`debug`	:	Debuggen einer Prozedur
`profile`	:	Statistik über Laufzeiten etc.
`rtime`	:	Zeitangabe
`time`	:	Zeitangabe
`trace`	:	Protokoll des Durchlaufs einer Prozedur

18 Technisches

`built_in`	:	Funktionen des MuPAD–Kerns
`bytes`	:	Speicherverbrauch
`loadlib`	:	Einladen von Bibliotheken
`loadproc`	:	Einladen von Prozeduren
`memory`	:	Speicherinformationen
`patchlevel`	:	Information über MuPAD–Patches
`pathname`	:	Information über Pfadnamen
`reset`	:	Re–Initialisierung einer MuPAD–Sitzung
`sysname`	:	Name des Betriebssystems
`system`	:	Ausführen eines Kommandos des Betriebssystems
`version`	:	Information über die Versionsnummer MuPADs