Technology
Arts Sciences
TH Köln

# Bachelor Mathematik
# für Informatiker mit Maple

Unterlagen zu den
Vorlesungen und Praktika
Mathematik 1

**W. Konen, A. Schmitter**

**S. Bagheri, B. Breiderhoff, F. Giannakopoulos, E. Lau, I. Legler, B. Naujoks, P. Wagner**

26. Oktober 2017

Copyright (C) 2007-2010 T. Bartz-Beielstein. Es ist Ihnen zu den folgenden Bedingungen gestattet, das Werk zu vervielfältigen, verbreiten und öffentlich zugänglich machen.

- Namensnennung. Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen.
- Keine kommerzielle Nutzung. Dieses Werk darf nicht für kommerzielle Zwecke verwendet werden.
- Keine Bearbeitung. Dieses Werk darf nicht bearbeitet oder in anderer Weise verändert werden.

Im Falle einer Verbreitung müssen Sie anderen die Lizenzbedingungen, unter welche dieses Werk fällt, mitteilen. Am Einfachsten ist es, einen Link auf die Seite http://creativecommons.org/licenses/by-nc-nd/3.0/deed.de einzubinden. Jede der vorgenannten Bedingungen kann aufgehoben werden, sofern Sie die Einwilligung des Rechteinhabers dazu erhalten. Diese Lizenz lässt die Urheberpersönlichkeitsrechte unberührt.

## Arbeiten mit diesem Skript

Vor jedem Praktikumstermin ist es erforderlich, dass Sie

- das im Lernsystem ILIAS (ilias.fh-koeln.de) bereitstehenden Aufgabenblatt herunterladen
- die im Aufgabenblatt genannten Kapitel des Skriptes lesen
- die Aufgaben bearbeiten. Es handelt sich um kleine MAPLE-Programme und handschriftliche Berechnungen. Bringen Sie Ihre Lösungen bitte unbedingt zum Praktikumstermin mit.

Während eines Praktikumstermins sollen Sie

- Ihre Lösungen und die im Aufgabenblatt aufgelisteten MAPLE-Befehle erläutern
- Fragen zu den Aufgaben beantworten und die nächsten Abschnitte vorbereiten.

Wir wünschen Ihnen interessante Vorlesungen und Praktika

Wolfgang Konen<br>Ane Schmitter<br>Samineh Bagheri<br>Beate Breiderhoff<br>Fotios Giannakopoulos<br>Elmar Lau<br>Inga Legler<br>Boris Naujoks<br>Peter Wagner

# Inhaltsverzeichnis


I. Maple Grundlagen 1

1. MAPLE: Arbeitsblattumgebung 2
1.1. MAPLE: Start 2
1.2. MAPLE: Dateioperationen 3
1.3. MAPLE: Hilfe 3

2. MAPLE: Einfache Berechnungen 5
2.1. MAPLE: Variablen, Zuweisungen, Arithmetik 5
2.1.1. Numerische Berechnungen 5
2.1.2. Zuweisungen, Symbolische Ausdrücke 5
2.1.3. Rechenoperationen 5
2.1.4. Zahldarstellung, Auswertung 6
2.1.5. Lösen von Gleichungen 6

3. MAPLE: 2- und 3-dimensionale Grafiken, Animationen 7
3.1. MAPLE: Zweidimensionale Plots 7
3.2. MAPLE: Dreidimensionale Plots 9
3.3. MAPLE: Darstellung impliziter Funktionen 9
3.3.1. Laden des plots Paketes 9
3.3.2. Befehle des plots Paketes 10
3.4. MAPLE: Animationen 10
3.5. MAPLE: Zusammengesetzte Funktionen 11

4. MAPLE: Programmieren 13
4.1. Maple: Einfache Programme 13
4.1.1. MAPLE: for-Schleifen 13
4.1.2. MAPLE: while-Schleifen 13
4.1.3. MAPLE: if-Bedingungen 14
4.1.4. MAPLE: Funktionen 15
4.1.5. MAPLE: Prozeduren 15
4.1.6. Gültigkeitsbereiche, Sichtbarkeit von Variablen 17
4.1.7. Auswertungsregeln für Variablen 17
4.1.8. Löschen von Variablenwerten, Unterdrückung der Auswertung 18
4.1.9. Prozeduren für komplizierte mathematische Funktionen 18
4.1.10. Weitere Programmiertechniken 19

II. Analysis I: Zahlsysteme, Folgen und Funktionen 20

5. Aussagenlogik und Mengenlehre 21
5.1. MAPLE: Aussagenlogik 21
5.1.1. MAPLE: Verknüpfung von Aussagen 21
5.1.2. MAPLE: Vereinfachen von Ausdrücken 21
5.1.3. MAPLE: Wahrheitstabellen 21

5.1.4. MAPLE: Regeln für Aussagen . . . 23
5.2. MAPLE: Folgen, Listen und Mengen . . . 23
5.2.1. MAPLE: Folgen . . . 23
5.2.2. MAPLE: Listen . . . 24
5.2.3. MAPLE: Mengen . . . 26
5.2.4. MAPLE: Operationen auf Listen . . . 26
5.2.5. MAPLE: Operationen auf Mengen . . . 27

**6. Zahlsysteme** **28**
6.1. Potenzen, Wurzeln und Logarithmen reeller Zahlen . . . 28
6.2. Gleichungen und Ungleichungen . . . 28
6.2.1. MAPLE: Lösung von Gleichungen und Systemen von Gleichungen . . . 28
6.2.2. MAPLE: Lösen von Ungleichungen . . . 30
6.2.3. MAPLE: Lösen von Betragsgleichungen . . . 30
6.2.4. MAPLE: Faktorenzerlegung . . . 30
6.3. Modulare Arithmetik . . . 31
6.4. Binomialkoeffizient und Summenzeichen . . . 31

**7. Zahlenfolgen** **32**
7.1. Definition und Eigenschaften von Folgen . . . 32
7.1.1. MAPLE: Folgen . . . 32
7.1.2. MAPLE: Schaubilder von Zahlenfolgen . . . 32
7.1.3. MAPLE: Grenzwert einer Zahlenfolge . . . 32
7.1.4. MAPLE: Divergente Zahlenfolgen . . . 33
7.1.5. MAPLE: Teilfolgen . . . 33
7.1.6. MAPLE: Rekursive Folgen . . . 34

**8. Reelle Funktionen** **37**
8.1. Allgemeine Funktionseigenschaften . . . 37
8.1.1. MAPLE: Nullstellen einer Funktion . . . 37
8.1.2. MAPLE: Darstellung und Animation von Funktionen . . . 37
8.1.3. MAPLE: Umkehrfunktionen . . . 39
8.2. MAPLE: Grenzwert von Funktionen . . . 39
8.3. MAPLE: Kurvendiskussion . . . 40

**III. Analysis I: Differentialrechnung** **43**

**9. Differentialrechnung** **44**
9.1. Übersicht . . . 44
9.2. Differenzierbarkeit, Ableitung, Differential, Ableitungsregeln . . . 44
9.2.1. Der Ableitungsbegriff . . . 44
9.2.2. Numerische Differentiation . . . 44
9.2.3. Ableitungsoperatoren in Maple . . . 45
9.3. Mittelwertsatz . . . 46
9.3.1. Beispiel aus dem Mathematik Skript . . . 46
9.3.2. Tutoren . . . 47
9.4. Satz von Taylor . . . 47
9.4.1. Taylor-Entwicklung in MAPLE . . . 47
9.4.2. Entwicklung der $e$-Funktion . . . 48
9.4.3. Animation der Taylor-Approximation . . . 48
9.4.4. Taylor-Tutor . . . 49
9.4.5. Taylorpolynome . . . 49

9.5. Regel von L'Hospital . . . 51
9.5.1. Definition . . . 51
9.5.2. Untersuchung der Funktion $\sin^2 x/\tan(x^2)$ . . . 52
9.5.3. Direkte Berechnung der Grenzwerte . . . 53
9.5.4. Der Limit Tutor und die Regel von L'Hospital . . . 53
9.6. Eigenschaften differenzierbarer Funktionen, Kurvendiskussion . . . 54
9.6.1. Extremwerte . . . 54
9.6.2. Eine Bemerkung zur analytischen Berechnung der Extremstellen . . . 55
9.6.3. Wendepunkte . . . 55

**IV. Lineare Algebra und Integralrechnung** **57**

**10. Lineare Algebra und Analytische Geometrie** **58**
10.1. Übersicht . . . 58
10.2. Maple Pakete . . . 58
10.3. Vektoren . . . 58
10.3.1. Betrag eines Vektors . . . 59
10.3.2. Vektoraddition und -multiplikation mit Skalaren . . . 59
10.3.3. Skalarprodukt . . . 60
10.3.4. Vektorprodukt . . . 60
10.3.5. Winkel zwischen zwei Vektoren . . . 60
10.4. Matrizen . . . 60
10.4.1. Rechenregeln für Matrizen . . . 62
10.4.2. Determinanten . . . 63
10.4.3. Rang einer Matrix . . . 63
10.5. Arrays . . . 63
10.5.1. Arrays in Maple . . . 63
10.5.2. Das Paket ArrayTools . . . 64
10.6. Systeme von linearen Gleichungen . . . 65
10.6.1. Lösungen linearen Gleichungssysteme . . . 65
10.6.2. Gauß-Verfahren . . . 65
10.6.3. Tutoren . . . 66
10.6.4. Der Befehl LinearSolve . . . 66
10.7. Punkte, Geraden und Ebenen mit geom3d . . . 67
10.7.1. Punkte . . . 67
10.7.2. Geraden . . . 67
10.7.3. Ebenen . . . 68
10.7.4. Schnitte von Geraden und Ebenen . . . 69
10.7.5. Schnittwinkel . . . 70
10.7.6. Abstände . . . 70
10.8. Geraden und Ebenen ohne das Paket geom3d . . . 71
10.8.1. Geraden . . . 71
10.8.2. Ebenen . . . 72

**11. Integralrechnung** **74**
11.1. Übersicht . . . 74
11.2. Das bestimmte Integral . . . 74
11.3. Numerische Integration . . . 74
11.4. Volumen von Rotationskörpern . . . 74
Index . . . 76

# Teil I.

# Maple Grundlagen

# 1. MAPLE: Arbeitsblattumgebung

## 1.1. MAPLE: Start

Nach dem Start von MAPLE erscheint ein leeres Arbeitsblatt. Das Eingabesymbol > in der ersten Zeile des Arbeitsblatts signalisiert, dass MAPLE arbeitsbereit ist und Kommandos von Ihnen erwartet. Wir empfehlen Ihnen, bevor Sie fortfahren, folgende Einstellung vorzunehmen: Rufen Sie mittels `Tools -> Options` den Reiter `Display` auf. Wählen Sie als `Input Display` die `Maple Notation` (statt `2-D Math Notation`), siehe Abbildung 1.1. Bestätigen Sie Ihre Auswahl, indem Sie auf `Apply Globally` klicken. Nachdem Sie mit einem Klick auf [> (aus der Kopfleiste) eine neue Eingabezeile geöffnet haben, könnnen Sie Kommandos in der Maple-Notation eingeben. Ihre Eingaben werden dann in roter Schriftfarbe dargestellt.

Ein MAPLE Arbeitsblatt enthält vier verschiedene Komponenten.

**Text:** Genau das, was hier geschrieben steht.

**Maple-Eingabezeilen:** Hinter dem Eingabesymbol > können Sie Kommandos (gewöhnlich in roter Farbe) eingeben, z.B.:

```
a:=13; # Hier steht ein Kommentar
```

Hinter der Raute steht ein Kommentar in der MAPLE-Eingabezeile.

Anweisungen können mit einem Semikolon oder einem Doppelpunkt beendet werden. Verwendet man ein Semikolon, dann werden die Befehle nach Betätigen der `[RETURN]`-Taste ausgeführt und das Ergebnis angezeigt. Beendet man eine Zeile mit dem Doppelpunkt, so wird das Kommando beim Betätigen der `[RETURN]`-Taste nur ausgeführt, das Ergebnis aber nicht angezeigt.

**Maple-Ausgabezeile:** In der Ausgabezeile werden die Ergebnisse von MAPLE (gewöhnlich in blauer Farbe) dargestellt.

```
5+8;
    13
```

**Grafiken:** Grafiken können skaliert, gedreht, animiert und ausgeschnitten werden, z.B. um sie in andere Arbeitsblätter und Programme einzufügen.

Hinweise:

- Mit der `[RETURN]`-Taste wird das gesamte Arbeitsblatt nach und nach von oben nach unten bearbeitet. Der Cursor springt dabei automatisch zur jeweils nächsten Zeile bzw. zum nächsten Block.
- Um MAPLE-Kommandos in mehrere Zeilen zu schreiben, die mit einem einzigen `[RETURN]` ausgeführt werden, drückt man am Ende jeder Zeile gleichzeitig die Tastenkombination `[SHIFT][RETURN]`.
- Falls Sie das ganze Arbeitsblatt in einem Schritt ausführen lassen wollen, wählen Sie im Menü `Edit->Execute Worksheet`. Um alle Ausführungen (Ausgabezeilen) im Arbeitsblatt zu entfernen, wählen Sie im Menü unter `Edit->Remove Output->FromWorksheet`. Nur wenn alle Ausführungen entfernt sind, existiert eine Kompatibilität zwischen den einzelnen MAPLE Versionen.

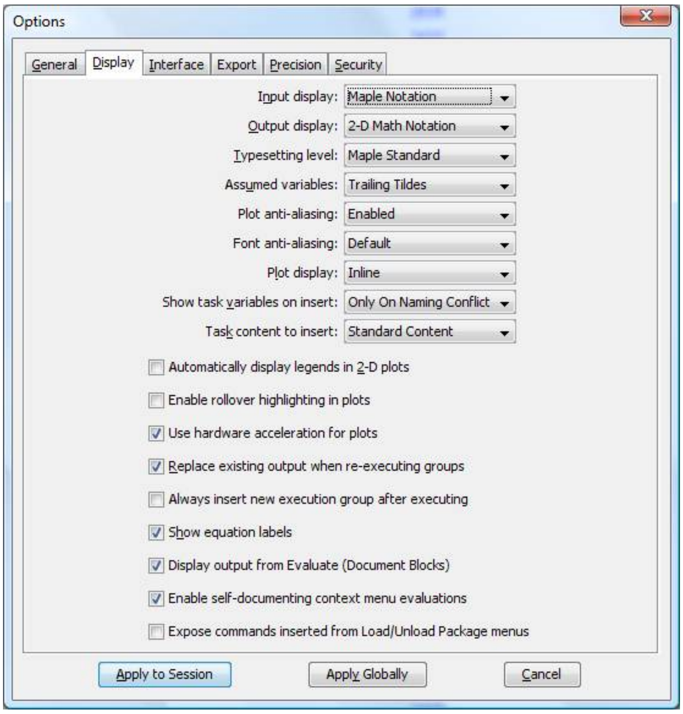


Abbildung 1.1.: Änderung des Eingabeformats

- Falls der Cursor sich irgendwo in dem Arbeitsblatt befindet, so befördern Sie ihn zurück zu einer `MAPLE`-Eingabezeile. Ausführungen von irgendwelchen Kommandos mitten im Arbeitsblatt können zu falschen Ergebnissen führen.
- Neue Blöcke mit einem `MAPLE`-Eingabesymbol > erhalten Sie durch das Anklicken des Symbols [> in der Kopfleiste.
- Jedes Arbeitsblatt (Worksheet) ist ein einzelnes Dokument. Die Dateien werden im .mw-Format (früher .mws) abgespeichert. `MAPLE` kann mehrere Arbeitsblätter gleichzeitig öffnen.
- Alle gleichzeitig geöffneten Worksheets können mit der gleichen `MAPLE`-Engine arbeiten. Das bedeutet, dass Zuweisungen, die in einem Worksheet getroffen wurden, im Anschluss in allen anderen Worksheets gültig sind. Es kann aber auch eine eigene `MAPLE`-Engine für jedes geöffnete Dokument gestartet werden. Sie können dies unter **`Tools->Options->General`** einstellen. **`Share one math engine among all documents`** verwendet die Variablen global, d.h. arbeitsblattübergreifend. **`New engine for each document`** benutzt entsprechend eine Engine für jedes Dokument.

## 1.2. `MAPLE`: Dateioperationen

`MAPLE` Arbeitsblätter können geladen und gespeichert werden. Zusätzlich gibt es diverse Import- und Exportmöglichkeiten. Da die `MAPLE` Ein- und Ausgabefunktionen wie in gängigen Textverarbeitungsprogrammen funktionieren, verzichten wir hier auf eine weiterführende Darstellung. Sie sollten in der Lage sein, bestehende `MAPLE` Arbeitsblätter zu laden und zu speichern.

## 1.3. `MAPLE`: Hilfe

`MAPLE` enthält ein komplettes interaktives Help-System. Man findet eine genaue Beschreibung der

Befehlssyntax mit detaillierter Liste der verschiedenen Optionen und Parameter, eine Erläuterung des Kommandos, einige Anwendungsbeispiele und weiterführende Informationen, Hilfen und verwandte Themen.

Um zum Beispiel Hilfe zu dem Kommando `plot3d` zu erhalten, kann man mit dem Cursor das Wort `plot3d` anklicken oder markieren, danach `Help` in der Kopfleiste anklicken und anschließend `Help on "plot3d"` auswählen. Eine andere Möglichkeit ist, in einer Eingabezeile das Kommando `?plot3d` einzugeben. Zurück zum aktuellen Arbeitsblatt kommen Sie durch Schließen des Fensters. Probieren Sie aus!

# 2. MAPLE: Einfache Berechnungen

## 2.1. MAPLE: Variablen, Zuweisungen, Arithmetik

### 2.1.1. Numerische Berechnungen

Mit MAPLE kann wie auf einem Taschenrechner gerechnet werden. Dazu gibt man hinter dem Eingabesymbol > den gewünschten, zu berechnenden Ausdruck ein, beendet diesen mit einem Semikolon und drückt anschließend die [RETURN]-Taste.

```
5+2;
```

Das Ergebnis wird direkt angezeigt. MAPLE besitzt zudem die folgenden Ausgabebefehle:

- `print`: Zentrierte Ausgabe
- `lprint`: linksbündige Ausgabe
- `printf`: formatierte Ausgabe.

Auch kann die Ausgabe vom Bildschirm mit dem Befehl `fprintf` in eine Datei umgeleitet werden.

### 2.1.2. Zuweisungen, Symbolische Ausdrücke

Möchte man einen Ausdruck einer Variablen zuweisen, so wird das Zeichen `:=` verwendet. Achtung, bei Variablennamen Groß- und Kleinbuchstaben beachten. Variablen werden während der ganzen Sitzung von MAPLE so lange behalten, bis sie durch das folgende Kommando gelöscht werden:

```
a:='a';
```

Mit dem `restart` Kommando werden alle Variablen gelöscht. Man hat anschließend den gleichen Zustand wie zu Beginn der MAPLE-Sitzung.

### 2.1.3. Rechenoperationen

Neben den Standardoperatoren wie `+`, `-`, `*` und `/` besitzt MAPLE viele verschiedene mathematische Funktionen, z. B. zur Berechnung der Fakultät:

```
50!;
```

$$30414093201713378043612608166064768844377641568960512000000000000$$

Um die lange Zahl nicht eingeben zu müssen, kann der letzte Ausdruck mit

```
%;
```

bezeichnet werden. Das MAPLE-Kommando für *hoch* ist `^`. Mit `a^n` wird die $n$-te Potenz der Zahl $a$ berechnet, `log[b](a)` berechnet den Logarithmus von $a$ zur Basis $b$ und `sqrt(a)` bestimmt die Quadratwurzel der Zahl $a$.

Mit Maple können endliche und unendliche Summen berechnet werden.

```
sum(i^2, i=1..10);
```

$$385$$

Viele weitere mathematische Funktionen wie z.B. sin, cos, mod oder auch der Ableitungsoperator `diff` stehen in MAPLE zur Verfügung, so dass praktisch alles, was Sie handschriftlich rechnen, in MAPLE berechnet werden kann.

### 2.1.4. Zahldarstellung, Auswertung

MAPLE unterscheidet Brüche (gebrochenrationale Zahlen wie 5/3) und Dezimalzahlen (wie 5./3.). MAPLE versucht Rundungsfehler zu vermeiden. Deshalb erhält man bei dem folgenden Bruch und der Wurzel das Ergebnis:
`(2^10/13)*sqrt(3);`

$$\frac{1024}{13}\sqrt{3}$$

Mittels `evalf(%,n)` ist auch eine Fließpunktdarstellung mit einer Näherung auf `n` Stellen möglich.
`evalf(%,3);`

```
136.
```

Wird kein Wert für `n` übergeben, so werden 10 Stellen dargestellt. Mit dem Befehl `Digits:=n` werden

- Darstellung und
- Genauigkeit der Rechnung für Dezimalzahlen (floats)

auf den Wert $n$ gesetzt. Ausdrücke wie
`(x+y)^3*(x+y)^2;`
werden in MAPLE sofort vereinfacht. Sie erhalten nach Drücken der `[RETURN]` Taste das Ergebnis:

$$(x+y)^5$$

Mit dem Kommando `simplify` kann MAPLE Ausdrücke vereinfachen, z.B. verwendet MAPLE die trigonometrischen Identitäten.
`simplify(cos(x)^5 + sin(x)^4 + 2*cos(x)^2 - 2*sin(x)^2 - cos(2*x));`

$$\cos(x)^4(\cos(x)+1)$$

Mit dem Befehl `expand(a)` werden Ausdrücke expandiert:
`expand((a+b)^2);` ergibt

$$a^2+2\,ab+b^2$$

### 2.1.5. Lösen von Gleichungen

#### Der Befehl solve

Mittels `solve(eq,var)` kann die Gleichung `eq` nach der Variablen `var` aufgelöst werden. So berechnet MAPLE nach Eingabe von
`solve(x^2+x-6=0,x);`
die Lösungen 2 und $-3$ der quadratischen Gleichung $x^2+x-6=0$. `solve` kann auch zum Lösen von Ungleichungen und Gleichungssystemen herangezogen werden. Die Lösungen werden als Liste zurückgegeben. Um anschließend die Lösungen einzelnen Variablen zuzuweisen, kann der Befehl `assign` benutzt werden.
`s := solve( {x+y=1, 2*x+y=3}, {x, y} );` ergibt

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

Anschließend können mittels
`assign(s);`
diese Werte den Variablen $x$ und $y$ zugewiesen werden.

Aufgabenstellungen zum Lösen von Gleichungen werden in Kapitel 6.2 behandelt.

# 3. MAPLE: 2- und 3-dimensionale Grafiken, Animationen

MAPLE unterstützt sowohl 2-dimensionale als auch 3-dimensionale Grafiken. Es lassen sich neben normalen Funktionen u.a. auch implizite Funktionen, parametrisierte Funktionen und Datenpunkte darstellen.

## 3.1. MAPLE: Zweidimensionale Plots

Funktionen einer Variablen können mit dem `plot` Befehl dargestellt werden. Abbildung 3.1 (links) zeigt den Graph einer Funktion:

```
restart;
plot(1/12*x^2-1, x=-6..6, title="Graph von 1/12*x^2-1");
```

Es können auch mehrere Funktionen, wenn diese als Liste, d.h. in eckigen Klammern (mehr zu Listen finden Sie Kapitel 5.2), an `plot` übergeben werden, dargestellt werden. Abbildung 3.1 (rechts) zeigt die Graphen der Funktionen $1/12x^2 - 1$ und $-1/12x^2 + 1$:

```
plot([1/12*x^2-1,-1/12*x^2+1] , x=-6..6,
 title="Graphen von 1/12*x^2-1 und -1/12*x^2+1", thickness =2);
```

Unter `?plot[options]` können Sie sich alle Einstellungen des `plot` Befehls anzeigen lassen. In den obigen Beispielen haben wir einen Titel eingefügt (`title`) und die Linienstärke (`thickness`) variiert. Grafiken können auch interaktiv verändert werden. Man klickt die Grafik an und wählt dann die Optionen aus dem Menü. So können auch Grafiken exportiert werden, z.B. im TIFF, GIF oder Postscript Format gespeichert werden.

Mit dem `display` Befehl können mehrere MAPLE Schaubilder in einem Schaubild dargestellt werden. Betrachten Sie das folgende Beispiel (der Befehl `display` benötigt die `plots` library, die zuerst geladen wird. Auf die `plots` library und das Laden von Paketen gehen wir in Kap. 3.3.1 ein):

```
with(plots):
F:=plot(cos(x), x=-Pi..Pi, y=-Pi..Pi, style=line):
G:=plot(tan(x), x=-Pi..Pi, y=-Pi..Pi, style=point):
display([F, G], axes=boxed, scaling=constrained, title="Kosinus und Tangens");
```

Das Resultat ist in Abbildung 3.2 dargestellt. Eine interessante Option für `display` ist `insequence`. Wählt man `insequence=false`, so werden die Abbildungen in einem Schaubild übereinander gezeichnet, `insequence=true` lässt die Bilder als Einzelbilder einer Animation ablaufen.

Manchmal ist eine spezielle Skalierung der $x$- oder $y$-Achsenbschriftung erwänscht. Mit dem Befehl `tickmarks` kann dies umgesetzt werden. Betrachten Sie dazu die Schaubilder der Sinusfunktion in Abbildung 3.3. Das linke Schaubild wurde mittels `plot(sin(x))` erzeugt. Das rechte Schaubild wurde mittels

```
with(plots):
plot(sin(x), x = -3*Pi .. 3*Pi, tickmarks = [spacing(Pi), default],
color = blue, thickness = 2);
```

erstellt.

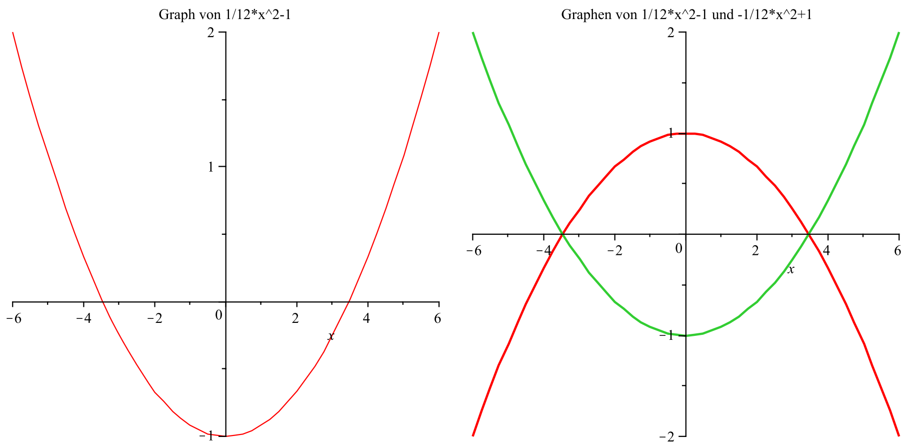


Abbildung 3.1.: Links: Graph einer Funktion. Rechts: Graphen der Funktionen $1/12x^2 - 1$ und $-1/12x^2 + 1$.

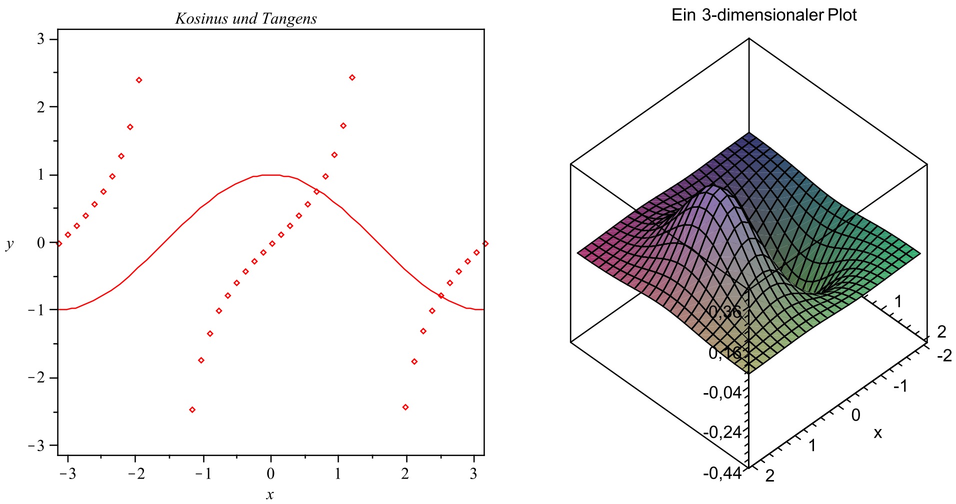


Abbildung 3.2.: *Links:* Mehrere Schaubilder in einem Schaubild mit dem display Befehl. *Rechts:* Graph einer dreidimensionalen Funktion.

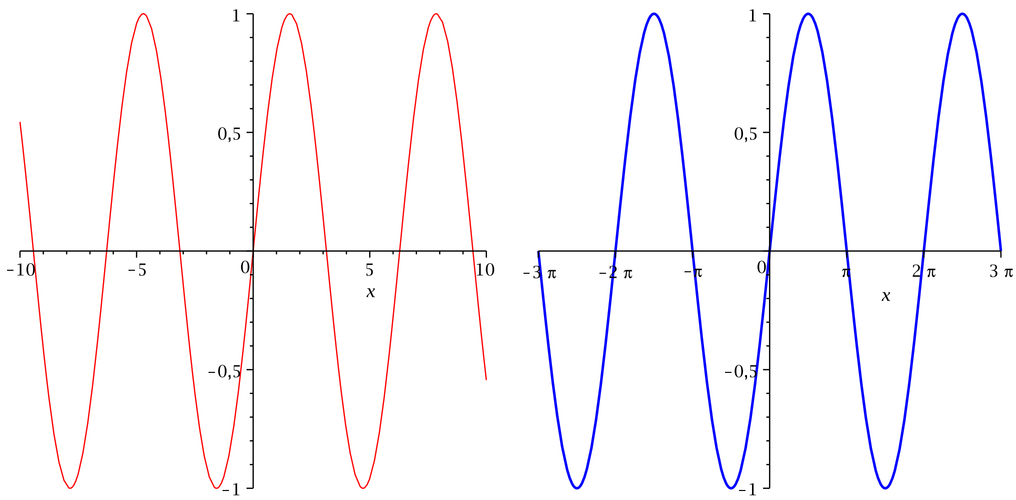


Abbildung 3.3.: Mit der Option `tickmarks` kann eine andere Achsenbeschriftung erzeugt werden.

## 3.2. MAPLE: Dreidimensionale Plots

Funktionen mit zwei Variablen können mit dem `plot3d` Befehl dargestellt werden. Abbildung 3.2 (rechts) zeigt den Graph von $x\exp(-x^2-y^2)$. Dabei ist mit `exp` die Exponentialfunktion bezeichnet.

```
plot3d(x*exp(-x^2-y^2), x=-2..2, y=-2..2,
axes=BOXED, title="Ein 3-dimensionaler Plot");
```

Durch die Option `axes=BOXED` erscheint das Koordinatensystem in Form eines Kastens. Die Grafik kann von verschiedenen Seiten betrachtet werden. Dazu klickt man die Grafik mit der linken Maustaste an und eine Box erscheint. Während die linke Maustaste gedrückt bleibt, kann die Grafik in die gewünschte Richtung gedreht werden und der Plot kann von verschiedenen Seiten betrachtet werden.

Es lassen sich mit `plot3d` auch mehrere Funktionen in einem Schaubild darstellen, wenn diese (wie beim `plot` Befehl) als Liste übergeben werden. Ebenso kann der Befehl `display3d` wie im Zweidimensionalen benutzt werden. Der Befehl `display` besitzt die Funktionalität des Befehls `display3d`, so dass einheitlich für zwei- und dreidimensionale Objekte der Befehl `display` benutzt werden kann.

## 3.3. MAPLE: Darstellung impliziter Funktionen

### 3.3.1. Laden des plots Paketes

Mit MAPLE lassen sich noch weitere komplexere Grafiken herstellen. Um diese Grafiken darzustellen, muss das `plots`-Paket geladen werden. Dies geschieht durch das Kommando `with(plots);`. Es erscheint eine Liste mit Kommandos aus dem `plots`-Paket. Diese Kommandos stehen nun zur Verfügung.

```
with(plots);
```

```
[ animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d,
conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot,
display, fieldplot, fieldplot3d, gradplot, gradplot3d, graphplot3d, implicitplot,
implicitplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot,
listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot,
```

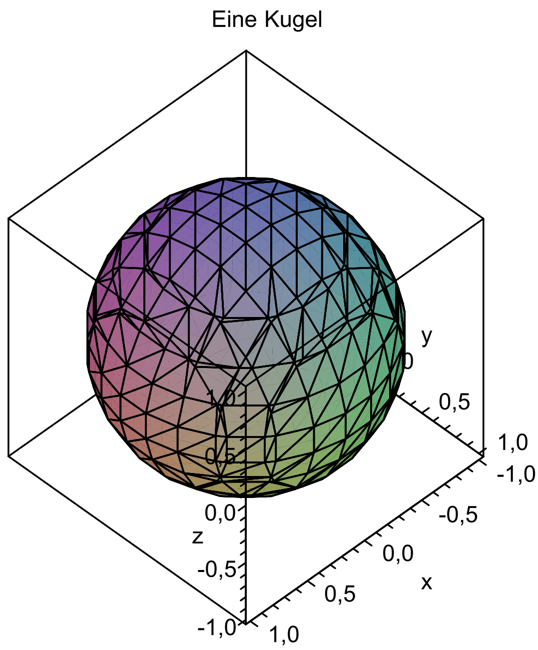


Abbildung 3.4.: Graph einer impliziten Funktion

```
multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot,
polygonplot3d, polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors,
setoptions, setoptions3d, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d,
tubeplot ]
```

Mit dem Befehl `with(Paketname)` können weitere Pakete geladen werden. Mittels `packages()` können die die bereits geladenen Pakete angezeigt werden. `unwith(Paketname)` löscht ein geladenes Paket.

### 3.3.2. Befehle des plots Paketes

Ein Kommando des `plots`-Pakets ist `implicitplot3d( )`. Mit diesem Kommando wird die Lösung einer impliziten Gleichung, z.B. `f(x,y,z)=0` dargestellt. Wir betrachten nun die Lösung der Gleichung $x^2 + y^2 + z^2 - 1 = 0$ und erhalten die in Abbildung 3.4 dargestellte Kugel.

```
implicitplot3d(x^2+y^2+z^2-1=0, x=-1..1, y=-1..1, z=-1..1,
scaling=constrained, axes=boxed, title="Eine Kugel");
```

Durch die Option `scaling=constrained` werden die Abstände auf allen Koordinatenachsen gleich groß.

## 3.4. MAPLE: Animationen

Das `plots`-Paket von MAPLE unterstützt somit auch 2-dimensionale und 3-dimensionale Animationen. Wir betrachten die Funktion

$$2\cos(tx) + \sin(t/2)^2 - 1.$$

Die Funktion soll im Bereich von $-\pi$ bis $\pi$ dargestellt werden.

```
animate(2*cos(t*x)+sin(t/2)^2-1, x=-Pi..Pi, t=-2..2);
```

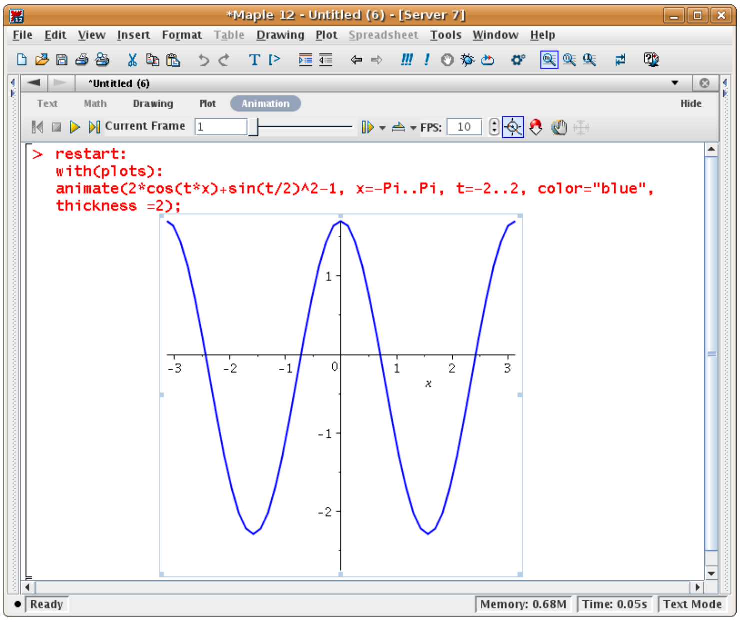


Abbildung 3.5.: Animationsmenü

Nach dem Anklicken der Grafik erscheint ein Extra-Rahmen um den Plot, siehe Abbildung 3.5. Wir drücken den Play-Knopf in der Kopfleiste und es beginnt eine Animation mit 16 Bildern. Bei dieser Animation wird die Variable $t$ in gleichen Abständen von -2 bis 2 durchlaufen.

## 3.5. MAPLE: Zusammengesetzte Funktionen

Zusammengesetzte Funktionen können mit dem `piecewise`-Befehl definiert werden. Wir betrachten die Funktion

$$f(x) = \begin{cases} \sin(2x), & \text{falls } x < \pi \\ \sin(20x), & \text{falls } \pi \leq x < 2\pi \\ \sin(x), & \text{sonst.} \end{cases}$$

Die Definition mittels `piecewise` lautet dann:

```
f := x-> piecewise(x < Pi, sin(2*x), x<2*Pi, sin(20*x),sin(x) );
plot(f(x), x=-2*Pi..4*Pi, thickness= 2,color = blue, font=[HELVETICA,14]);
```

Abbildung 3.6 zeigt das Schaubild dieser stückweise definierten Funktion.

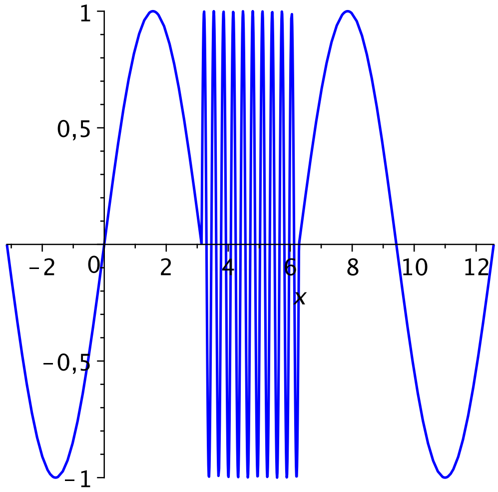


Abbildung 3.6.: Schaubild einer zusammengesetzten Funktion.

# 4. MAPLE: Programmieren

Dieses Kapitel gibt einen kurzen Einblick in das Programmieren mit `MAPLE`. Wir betrachten einfache Schleifenbildungen mit `for` und `while`, Verzweigungen mit `if` und Unterprogramme, die mittels `proc` erstellt werden.

## 4.1. Maple: Einfache Programme

`MAPLE` besitzt entsprechend zu anderen Programmiersprachen ein paar einfache Programmstrukturen, wie Schleifen, Verzweigungen, Prozeduren.

### 4.1.1. MAPLE: for-Schleifen

Bei einer `for`-Schleife wird der Schleifenkopf durch `for` eingeleitet, gefolgt von der Schleifenvariablen. Der Startwert der Variablen steht hinter `from`, der Endwert hinter `to`, das Inkrement ist optional und kann hinter `by` angegeben werden. Standardmäßig wird um eins hochgezählt. Der Schleifenkörper beginnt mit `do` und endet mit `end do` oder alternativ mit `od`.

Eine `for`-Schleife zur Berechnung der Summe der Zahlen von 1 bis 10:

```
iz:= 0;
for i from 1 to 10 by 1 do
   iz:= iz + i;
od;
```

$$0$$
$$1$$
$$3$$
$$6$$
$$10$$
$$15$$
$$21$$
$$28$$
$$36$$
$$45$$
$$55$$

### 4.1.2. MAPLE: while-Schleifen

Eine weitere Schleife ist die `while`-Schleife. Die Anweisungen im Schleifenkörper werden so lange ausgeführt, solange die Bedingung den Wert `true` hat. Ergibt die Bedingung den Wert `false`, bricht die Schleife ab.

Das obige Beispiel mit einer `while`-Schleife:

```
iz:=0;
i:=1;
while i <= 10 do
   iz:= iz+i:
   i:= i+1
od;
```

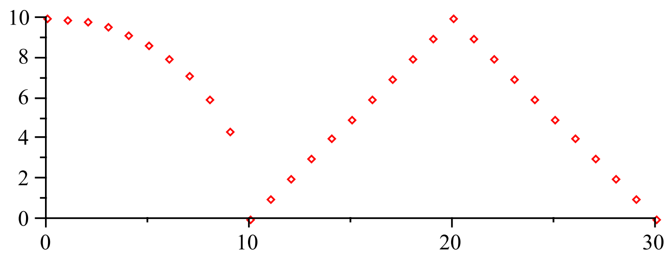


Abbildung 4.1.: Graph einer Folge

### 4.1.3. MAPLE: if-Bedingungen

Verzweigungen werden mit den Schlüsselworten `if`, `then`, `elif`, `else` und `fi` gebildet. Sie können beliebig verschachtelt werden. `if` und `fi` bilden Anfang und Ende der Verzweigungsanweisung, `then` leitet den Zweig ein, der ausgeführt wird, wenn die Bedingung erfüllt ist. Ist die Bedingung nicht erfüllt, werden die auf `else` folgenden Anweisungen berücksichtigt. `elif` ist eine Abkürzung für `elseif` und hat dieselbe Bedeutung. Zur Beendigung einer `elif`-Verzweigung ist kein `fi` nötig. Beispiel: Die Folgenglieder `fm[0], fm[1], fm[2], ..., fm[30]` werden definiert. Mit der Funktion `sqrt(.)` wird die Quadratwurzel berechnet.

```
for i from 0 to 30 do
   if i < 10
      then fm[i]:= sqrt(100-i^2)
      elif i < 20
         then fm[i]:= i-10
         else fm[i]:= 30-i
   fi
od;
```

Und zur grafischen Darstellung:
`data2:= [seq([i,fm[i]], i=0..30)];`

$$data2 := [[0, 10], [1, 3\sqrt{11}], [2, 4\sqrt{6}], ...usw.$$

`plot(data2, style=point, scaling=constrained);`
Durch die Option `scaling=constrained` werden die Abstände auf allen Koordinatenachsen gleich groß, siehe Abbildung 4.1.

### 4.1.4. MAPLE: Funktionen

Funktionen sind sehr wichtige Elemente in der Mathematik. Standardfunktionen wie sin oder tan sind schon in MAPLE implementiert.

Für das Definieren von Funktionen kennt MAPLE mehrere Möglichkeiten. Eine Möglichkeit ist, den Pfeil-Operator zu benutzten. Diese ist der bekannten mathematischen Notation am ähnlichsten. Der Pfeil besteht aus dem Minus-Zeichen gefolgt von dem Größer-Zeichen.

```
f:= x -> x^2+1/2;
```

$$x \mapsto x^2 + 1/2$$

Funktionswerte können nun berechnet werden.

```
f(2);
```

$$\frac{9}{2}$$

Genauso funktioniert

```
f(u+v);
```

$$(u+v)^2 + 1/2$$

### 4.1.5. MAPLE: Prozeduren

Eine Prozedur wird durch die Zuweisung auf eine Variable definiert. Die Definition der Prozedur beginnt mit dem Schlüsselwort `proc` und endet mit dem Schlüsselwort `end`. Die Parameterliste wird in Klammern hinter dem Schlüsselwort `proc` angegeben. Es kann ein Deklarationsteil mit lokalen Variablen folgen. Im Anschluss befinden sich die Anweisungen der Prozedur.

Ein einfaches Beispiel ist eine Prozedur, die die Fläche eines Kreises zu einem vorgegebenen Radius berechnet.

```
flaeche := proc(radius)
Pi*radius^2
end proc:
```

Rufen Sie nun `flaeche(2)` auf, so erhalten Sie

$$4\ \pi.$$

Worin unterscheiden sich nun Funktionen und Prozeduren? Prozeduren gleichen Funktionen, wenn die Prozeduren nur aus einer Anweisung bestehen. Dieser Ausdruck darf dabei keine Zuweisung, Schleife oder `if`-Bedingung sein. Dies trifft auf das Beispiel zur Flächenberechnung zu. Die Flächenberechnung hätte daher auch als Funktion implementiert werden können:

```
flaeche2 := radius -> Pi*radius^2;
```

$$flaeche2 := radius- > \pi radius^2.$$

Rufen Sie nun `flaeche2(2)` auf, so erhalten Sie

$$4\ \pi.$$

## Prozedurparameter

Es können auch Variablen an Prozeduren übergeben werden oder Prozeduren, die keine Parameter besitzen, definiert werden. Die Angabe des Datentypen der Parameter ist optional. Datentypen können hinter zwei Doppelpunkten angegeben werden. Die Angabe der Datentypen erleichtert die Fehlersuche bei komplexen Programmen. `MAPLE` bricht mit einer Fehlermeldung ab, falls ein falscher Datentyp übergeben wird.

Als weiteres Beispiel wird die Teilbarkeit von natürlichen Zahlen betrachtet. Die folgende Prozedur prüft, ob die eingegebene Zahl $n$ eine Primzahl ist. $n$ ist hier eine nichtnegative Integerzahl (Datentyp `nonnegint`). Primzahlen spielen in der Kryptographie in wichtige Rolle. Mit der Funktion `sqrt(.)` wird die Quadratwurzel berechnet. Durch die Funktion `floor( )` werden die Nachkommastellen abgeschnitten, man erhält ein ganze Zahl (Bsp.: `floor(12.345)`=12). Die Funktion `mod` heißt modulo, durch $n \mod i$ wird der Rest berechnet, welcher bei der ganzzahligen Division von $n$ durch $i$ entsteht (Bsp.: $12 \mod 5 = 2$). Die Ausführung des Kommandos `return` bewirkt ein Verlassen der Prozedur, bei `return` (false) erhält `fprim(n)` aus der folgenden `for`-Schleife den Wert `false`, bei `return` (true) den Wert `true`.

```
fprim:= proc(n::nonnegint)
   local i:
   if (n=1) # 1 ist keine Primzahl
      then return(false)
   fi;
   for i from 2 to floor(sqrt(n)) do
      if (n mod i) = 0
         then return(false)
      fi
   od:
   return true;
end;
```

Wir prüfen die Prozedur für einige Zahlen:

```
for n from 2 to 20 do
   n, ist_eine_Primzahl, fprim(n);
od;
```

$2, ist\_eine\_Primzahl, true$
$3, ist\_eine\_Primzahl, true$
$4, ist\_eine\_Primzahl, false$
$5, ist\_eine\_Primzahl, true$
$6, ist\_eine\_Primzahl, false$
$7, ist\_eine\_Primzahl, true$
$8, ist\_eine\_Primzahl, false$
$9, ist\_eine\_Primzahl, false$
$10, ist\_eine\_Primzahl, false$
$11, ist\_eine\_Primzahl, true$
$12, ist\_eine\_Primzahl, false$
$13, ist\_eine\_Primzahl, true$
$14, ist\_eine\_Primzahl, false$
$15, ist\_eine\_Primzahl, false$
$16, ist\_eine\_Primzahl, false$
$17, ist\_eine\_Primzahl, true$
$18, ist\_eine\_Primzahl, false$
$19, ist\_eine\_Primzahl, true$
$20, ist\_eine\_Primzahl, false$

#### Rückgabewerte von Prozeduren

MAPLE Prozeduren liefern den Wert des letzten Ausdrucks zurück. Hierbei gelten `od`, `end do` und `end if` nicht als Ausdrücke. Um ein Ergebnis zurückzugeben, das nicht im letzten Ausdruck berechnet wurde, kann der gewünschte Rückgabewert in der letzten Zeile vor `end proc` explizit angegeben werden.

Mit dem Befehl `return` kann ein anderer, als der letzte ausgewertete Ausdruck zurückgegeben werden. `return` kann an einer beliebigen Position benutzt werden, z.B. auch um `if` Klauseln zu verlassen.

Es können auch mehrere Werte zurückgegeben werden. Dazu trägt man mehrere Werte in die letzte Zeile (oder hinter `return`). Auch können die Rückgabewerte als Liste zurückgeben werden. Listen betrachten wir eingehend in Kapitel 5.2.

### 4.1.6. Gültigkeitsbereiche, Sichtbarkeit von Variablen

MAPLE unterscheidet globale und lokale Variablen. Parameter, die an Prozeduren übergeben werden, dürfen nicht deklariert werden. Betrachten Sie dazu das folgende Beispiel:

```
restart:
flaeche := proc(radius)
local radius;
Pi*radius^2
end proc:
```

führt zu folgender Fehlermeldung:

```
Error, parameter and local ‘radius‘ have the same name in procedure flaeche
```

Lokale Variablen sind außerhalb der Prozedur nicht sichtbar. Sie sind nur innerhalb der Prozedur gültig. Globale Variablen sind hingegen in der gesamten Arbeitsumgebung gültig. Globale Variablen sollten nur selten eingesetzt werden, da unerwünschte Seiteneffekte auftreten können.

### 4.1.7. Auswertungsregeln für Variablen

Es gibt wichtige Unterschiede bei der Auswertung verschiedener Bezeichner in MAPLE.

- Die in der interaktiven Ebene, d.h. im Arbeitsblatt, definierten Variablen werden auch in Prozeduren vollständig ausgewertet.
- Lokale Variablen werden nur auf einer Ebene ausgewertet.
- Parameter werden nur einmal ausgewertet.

Betrachten Sie dazu folgendes Beispiel:

```
restart:
a:=b;
b:=c;
c:=1;
```

```
a:= b
b:= c
c:= 1
```

```
a,b,c;
```

```
1, 1, 1
```

Die Variable $c$ wird einmal ausgewertet, $b$ und $c$ werden mehrfach ausgewertet, da wir uns in der inaktiven Ebene befinden. $a$ und $b$ nehmen somit den Wert von $c$ an. Dieser Wert wird ihnen aber nicht zugewiesen, wie durch die folgende Zeile deutlich wird:

```
b:=2:
a,b,c;
```

```
2, 2, 1
```

Nun besitzt $a$ den Wert 2.

### 4.1.8. Löschen von Variablenwerten, Unterdrückung der Auswertung

Wird eine Variable, die zuvor mit einem Wert belegt wurde, wie z.B.
`restart;`
`x:=1;`
wieder als Unbestimmte benötigt, so kann sie durch
`x:='x';`
zurückgesetzt werden. Die einfachen Anführungszeichen („Apostrophe“) verhindern die Auswertung eines Ausdrucks. So kann auch die Auswertung eines kompletten Ausdrucks durch `MAPLE` verhindert werden. Setzen wir den Term `a`
`a:=x+1;`

$$a := x + 1$$

in die Gleichung `g` in einfachen Anführungszeichen ein
`g:=('a'+b)^2=0;`

$$g := (a + b)^2 = 0$$

so unterbleibt zunächst die Auswertung. Wird die Gleichung `g` erneut aufgerufen, so wird `a` ausgewertet:
`g;`

$$(x + 1 + b)^2 = 0$$

Dieses Verfahren kann wiederholt benutzt werden: Wird ein Ausdruck $n$-mal in einfache Anführungszeichen gesetzt, so wird seine Auswertung $n$-mal unterbunden.

### 4.1.9. Prozeduren für komplizierte mathematische Funktionen

Um kompliziertere mathematische Funktionen zu definieren, kann man auch Prozeduren benutzen. Als Beispiel betrachten wir eine Funktion $f$, die stückweise definiert ist:

$$f(x) = \begin{cases} x + 1, & \text{falls } x > 0 \\ 1, & \text{sonst.} \end{cases} \tag{4.1}$$

Diese Funktion kann mit dem `if`-Befehl wie folgt definiert werden:

```
fplus:= proc(x)
   if x > 0 then x+1
            else 1
   fi;
end;
```

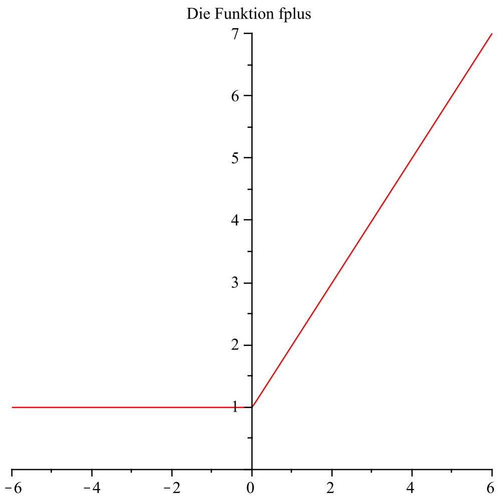


Abbildung 4.2.: Graph einer Prozedur

`MAPLE` berechnet:
`fplus(-1);`

1

Allerdings führt der Aufruf
`plot(fplus(x), x= -6..6, title="Die Funktion fplus");`
zu der folgenden Fehlermeldung:

`Error, (in fplus) cannot determine if this expression is true or false:`$0 < x$

Dies hängt mit den in Abschnitt 4.1.8 besprochenen Auswertungsregeln zusammen. Setzt man den Funktionsaufruf in einfache Anführungszeichen, dann wird die gewünschte Abbildung ohne Fehlermeldungen erzeugt, siehe Abbildung 4.2. Der entsprechende Aufruf des `plot`-Befehls für die selbst definierte Prozedur lautet daher:
`plot('fplus(x)', x= -6..6, title="Die Funktion fplus");`
Alternativ kann die Abbildung auch durch den Befehl
`plot(fplus, -6..6, 0..7, title="Die Funktion fplus");` erzeugt werden. Zudem kann die Funktionsdefinition der in (4.1) dargestellten Funktion auch mit dem `piecewise`-Befehl erfolgen.

### 4.1.10. Weitere Programmiertechniken

In diesem Kapitel haben wir Ihnen nur einen kurzen Einblick in die Möglichkeiten, die `MAPLE` als Programmiersprache bietet, geben können. So können Funktionen an Prozeduren übergeben und von ihnen zurückgegeben werden, Optionen als Argumente angegeben werden oder Fehlerbehandlungsroutinen programmiert werden. Walz (2002) gibt sehr viele Beispiele an.

# Teil II.

# Analysis I: Zahlsysteme, Folgen und Funktionen

# 5. Aussagenlogik und Mengenlehre

## 5.1. MAPLE: Aussagenlogik

### 5.1.1. MAPLE: Verknüpfung von Aussagen

Nach dem Kommando
`with(Logic);`
stehen die logischen Operatoren und Funktionen zur Verfügung. Wir stellen in Tabelle 5.1 die wichtigsten logischen Verknüpfungen von Aussagen und die entsprechenden MAPLE Befehle zusammen.

Tabelle 5.1.: Verknüpfung von Aussagen

| **Bezeichnung** | **Beschreibung** | **Maple Schreibweise** |
|---|---|---|
| Konjunktion | "und" | `&and` |
| Disjunktion | "oder" | `&or` |
| Negation | "nicht" | `&not` |
| Implikation | "wenn - dann" | `&implies` |
| Äquivalenz | "genau dann, wenn" | `Equivalent` |

### 5.1.2. MAPLE: Vereinfachen von Ausdrücken

Ausdrücke können folgendermaßen vereinfacht werden:

```
BooleanSimplify(&not(p &implies q) &or q);
```

```
p &or q
```

### 5.1.3. MAPLE: Wahrheitstabellen

Eine gute Übersicht über Aussagen erhält man mit Wahrheitstabellen. Der MAPLE-Befehl `TruthTable` liefert folgende Ausgabe.

```
TruthTable(&not(A &or B),[A,B]);
```

```
table([(false, true) = false, (true, true) = false, (false, false) = true,
(true, false) = false])
```

Die in Abbildung 5.1 dargestellte MAPLE-Prozedur WWT berechnet zu jedem gültigen logischen Ausdruck die Wahrheitstabelle in bekannter Form. Wir testen die Prozedur an einem Beispiel:

```
WWT(&not(A &or B),{A,B});
```

$$\begin{bmatrix} A & B & WW \\ true & true & false \\ true & false & false \\ false & true & false \\ false & false & true \end{bmatrix}$$

```
WWT:=proc(x,y) local i,j,n,v,a,z;
if not type(x,function)then
   ERROR(`var1 should be a boolean expression`)
fi:
if not type(y,set)then
   ERROR(`incorrect type in var2`)
fi:
n:=nops(y);
v:=array(1..n);
a:=x:
z:=array(1..2^n+1,1..n+1,sparse);
for i from 1 to n do
    z[1,i]:=op(i,y)
od:
z[1,n+1]:='WW';
for i from 1 to n do
    for j from 1 to 2^n do
        if trunc((j-1)/2^(n-i)) mod 2 = 0 then
           z[j+1,i]:= true
 else
 z[j+1,i]:=false
        fi :
    od:
od:
for i from 1 to 2^n do
  a:=x:
    for j from 1 to n do
       v[j]:=z[i+1,j];
       a:=subs(op(j,y)=v[j],a):
    od:
        if (Tautology(a)) then
           z[i+1,n+1]:= true
          else z[i+1,n+1]:=false
        fi:
     od:
print(z):
end:
```

Abbildung 5.1.: Die Prozedur `WWT`.

### 5.1.4. MAPLE: Regeln für Aussagen

Die aussagenlogische Regeln können mit MAPLE überprüft werden. Als Beispiel zeigen wir die Gültigkeit der de Morgan'schen Gesetze:

**Beispiel 5.1 (Die De Morgan'sche Regel 1)**
*Equivalent(&not(A &or B), (&not A) &and (&not B));*

*true*

□

**Beispiel 5.2 (Die De Morgan'sche Regel 2)**
*Die zweite De Morgan'sche Regel mit Wahrheitstafeln:*

```
s:=&not(A &and  B):
t:=(&not A) &or  (&not B):
WWT(s,{A,B});
WWT(t,{A,B});
```

$$\begin{bmatrix} A & B & WW \\ true & true & false \\ true & false & true \\ false & true & true \\ false & false & true \end{bmatrix}$$

$$\begin{bmatrix} A & B & WW \\ true & true & false \\ true & false & true \\ false & true & true \\ false & false & true \end{bmatrix}$$

□

## 5.2. MAPLE: Folgen, Listen und Mengen

Eine geordnete Ansammlung eines oder mehrerer MAPLE Ausdrücke ist eine Folge (engl. sequence, seq). In MAPLE entstehen Listen durch Einschluss einer Folge in eckige Klammern, Mengen entstehen durch Einschluss einer Folge in geschweifte Klammern. Wir betrachten zunächst Folgen, da aus ihnen Listen und Mengen gebildet werden.

### 5.2.1. MAPLE: Folgen

Folgen können aus unterschiedlichen Datentypen bestehen.

```
folge := 1, 2 ,3 ,a , b, c, Pi;
```

folge := 1, 2, 3, a, b, c, $\pi$

Über den Index kann auf einzelne Elemente einer Folge zugegriffen werden.

```
folge[5];
```

b

Sie können auch auf Bereiche zugreifen `folge[2 .. 4]` oder den Index vom letzten Element an rückwärts zählen lassen.

```
folge[-2 .. -1];
```

$c, \pi$

Folgen können einfach mit dem Befehl `seq` erzeugt werden.

```
seq(i, i=-10..10);
```

$-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$

Statt eines Bereichs kann auch ein Ausdruck an `seq` übergeben werden, der dann gliedweise abgearbeitet wird.

```
seq(i/2, i = [2,4,6]);
```

$1, 2, 3$

Folgen können durch Voranstellung oder Hintenanfügung weiterer Elemente (oder Folgen) ergänzt werden:

```
folge2 := folge, 4;
```

$folge2 := 1, 2, 3, a, b, c, \pi, 4$

Lösungen, die mit dem Befehl `solve` ermittelt werden, werden als Folge zurückgegeben:

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

$0, -1, -1$

Auf die zweite Lösung kann dann z.B. mittels

```
%[2];
```

$-1$

zugegriffen werden.

### 5.2.2. MAPLE: Listen

Schließt man eine Folge in eckige Klammern ein, so entsteht eine Liste. Listen stellen eine geordnete Sammlung von Objekten dar.

```
liste := [1,2,3,1,2,4];
```

$liste := [1, 2, 3, 1, 2, 4]$

Der Zugriff auf einzelne Listenelemente erfolgt über den Index

```
liste[3];
```

$3$

Der Befehl `op` gibt eine bestimmte Auswahl von Listenelementen zurück: `op(m, liste)` gibt das $m$-te Element zurück. `op(m..n, liste)` gibt die Teilfolge vom m-ten bis zum $n$-ten Element zurück.

```
liste := [2,4,6,8,10]:

op(5,liste);
```

$$10$$

`op(0, ausdruck)` gibt den Datentyp zurück:

```
op(0,liste);
```

$$list$$

Mit `nops` kann die Anzahl der Elemente einer Liste (oder auch einer Menge) ermittelt werden.

```
nops(liste);
```

$$5$$

`member(element,liste)` überprüft, ob das Element `element` in der Liste (oder auch Menge) enthalten ist.

```
member(2,liste);
```

$$true$$

Um ein Element `x` am Ende einer Liste `L` anzuhängen, benutzt man den Befehl `[op(L), x];`

```
liste := [1,2,3,1,2,4];
liste := [op(liste), a];
```

$$[1, 2, 3, 1, 2, 4]$$
$$[1, 2, 3, 1, 2, 4, a]$$

### MAPLE: Listen von Listen

Listen von Listen sind bei der Erstellung eines Schaubildes mit `plot` hilfreich. Um eine Liste `pts` zu erzeugen, die laufend die Punktkoordinaten $(x, y)$ aufnimmt, muss die Variable zuerst mit `NULL` initialisiert werden.

```
pts := NULL:
for x from 0 to 1 by 1/plots[setoptions](numpoints) do
    pts := pts, [x, sqrt(abs(2-x^2))]
end do:
```

MAPLE verwendet den Befehl `abs()`, um den Betrag zu berechnen. Es gilt:

$$|x| = \begin{cases} -x, & \text{falls } x < 0 \\ x, & \text{sonst.} \end{cases} \tag{5.1}$$

Zur Darstellung der Punkte mit `plot` muss die Folge `pts` in eine Liste umgewandelt werden, so dass eine Liste von Listen entsteht.

```
pts := [pts]:
plot(pts);
```

ergibt das in Abbildung 5.2 dargestellte Kreissegment.

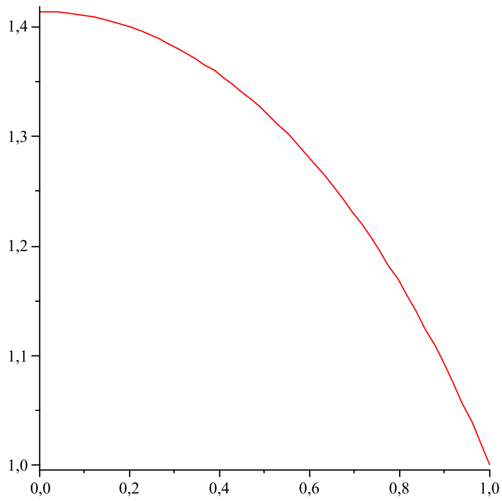


Abbildung 5.2.: Kreissegment

### 5.2.3. MAPLE: Mengen

Mengen enthalten im Gegensatz zu Listen keine Elemente mehrfach. Sie werden ebenfalls aus Folgen gebildet. Diese werden in geschweifte Klammern gesetzt. Die Reihenfolge der Elemente kann von Sitzung zu Sitzung variieren. Daher sind Indexzugriffe, obwohl möglich, nicht sinnvoll.

```
menge := {1,2,3,1,2,4};
```

$menge := 1, 2, 3, 4$

Eine Menge kann mit dem Befehl `op` in eine Folge umgewandelt werden:

```
op(menge);
```

$1, 2, 3, 4$

Die leere Menge wird mit `leer := {}` erzeugt. `{NULL}` ist gleichbedeutend mit `{}`.

### 5.2.4. MAPLE: Operationen auf Listen

#### Austausch von Elementen

Zum Austausch einzelner Listenelemente wird der Befehl `subsop` benutzt.

```
liste := [1,2,3,4];
```

$liste := [1, 2, 3, 4]$

```
liste:=subsop(4=5,liste);
```

$liste := [1, 2, 3, 5]$

Somit können auch Elemente aus Listen gelöscht werden, indem die Konstante `NULL` benutzt wird:

```
liste:=subsop(2=NULL,liste);
```

$liste := [1, 3, 5]$

#### Sortieren

Listen können mit dem Befehl `sort` sortiert werden.
`liste:=[3,4,1,2]: sort(liste);`

$$[1, 2, 3, 4]$$

`sort(liste, ‘>‘);`
gibt eine absteigend sortierte Liste aus. Welche Anführungszeichen müssen Sie verwenden, um das gewünschte Resultat zu erhalten?

### 5.2.5. MAPLE: Operationen auf Mengen

Als Beispiele betrachten wir die folgenden Mengen:
`M1 := {1,2,3};`
und
`M2 := {a,b};`
Folgende Operationen stehen für Mengen zur Verfügung.

**Vereinigung:** `union`, z.B.
```
M1 union M2;
```

$$\{1, 2, 3, b, a\}$$

**Schnitt:** `intersect` z.B.
```
M1 intersect (M1 union M2);
```

$$\{1, 2, 3\}$$

**Komplementbildung:** `minus`, z.B.
```
(M1 union M2) minus M1;
```

$$\{b, a\}$$

# 6. Zahlsysteme

Wir setzen voraus, dass Sie die Definitionen für natürliche, ganze, rationale und reelle Zahlen kennen und wenden uns den Rechenoperationen für reelle Zahlen zu

## 6.1. Potenzen, Wurzeln und Logarithmen reeller Zahlen

Wir stellen in Tabelle 6.1 die wichtigsten mathematischen Funktionen und die entsprechenden `MAPLE` Befehle zusammen.

## 6.2. Gleichungen und Ungleichungen

### 6.2.1. `MAPLE`: Lösung von Gleichungen und Systemen von Gleichungen

Die einfache Gleichung $x^2 = 2$ wird mit dem Kommando `solve` gelöst.
`solve(x^2=2);`

$$\sqrt{2}, -\sqrt{2}$$

Als nächstes wird eine Gleichung mit dem Namen `gl` betrachtet.
`gl:= x^3-1/2*x^2*a+13/3*x^2 = 13/6*x*a+10/3*x-5/3*a;`

$$x^3 - 1/2\,x^2a + 13/3\,x^2 = \tfrac{13}{6}\,xa + 10/3\,x - 5/3\,a$$

Durch Angabe der Variablen `x` im Kommando `solve` wird die Gleichung `gl` nach `x` gelöst.
`solve(gl,x);`

$$-5, 2/3, (1/2) * a$$

`MAPLE` kann auch Systeme von Gleichungen lösen. Als Beispiel sind die folgenden drei Gleichungen mit den Namen `eqn1`, `eqn2` und `eqn3` und den drei Variablen $x$, $y$ und $z$ gegeben.

```
eqn1:= x+2*y+3*z=41;
eqn2:= 5*x+5*y+4*z=20;
eqn3:= 3*y+4*z=125;
```

Nun wird das System in den drei Variablen gelöst.
`s1:= solve({eqn1,eqn2,eqn3}, {x,y,z});`

$$s1 := \left\{x = -\tfrac{527}{13}, z = -\tfrac{70}{13}, y = \tfrac{635}{13}\right\}$$

Auch Gleichungssysteme, welche mehr Variablen als Gleichungen besitzen, können gelöst werden. Im folgenden Beispiel betrachten wir nur die ersten beiden Gleichung `eqn1` und `eqn2`. `MAPLE` soll nun die beiden Gleichungen nur in den Variablen $x$ und $y$ lösen.
`s2:= solve({eqn1,eqn2}, {x,y});`
Die Lösungen für die gesuchten Variablen $x$ und $y$ hängen von $z$ ab:

$$s2 := \left\{y = -\tfrac{11}{5}\,z + 37, x = 7/5\,z - 33\right\}$$

Tabelle 6.1.: Mathematische Funktionen

| Mathematische Funktion | Maple Schreibweise |
|---|---|
| $a^x$ | `a^x` |
| $e^x$ | `exp(x)` |
| $\sqrt{x}$ | `sqrt(x)` |
| $\sqrt[n]{x}$ | `surd(x,n)` |
| $\log_a x$ | `log[a](x)` |
| $\ln x$ | `ln(x)` |
| $\sin x$ | `sin(x)` |
| $\cos x$ | `cos(x)` |
| $\tan x$ | `tan(x)` |
| $\cot x$ | `cot(x)` |
| $\arcsin x$ | `arcsin(x)` |
| $\arccos x$ | `arccos(x)` |
| $\arctan x$ | `arctan(x)` |
| $\lvert x\rvert$ | `abs(x)` |
| $n!$ | `n!` oder `factorial(n)` |
| $\binom{a}{b}$ | `binomial(a,b)` |
| $\sum_{k=1}^{n} x_k$ | `sum(x, k=1..n)` |
| $a \mod b$ | `a mod b` |
| $\lfloor x\rfloor$ | `floor(x)` |
| $\lceil x\rceil$ | `ceil(x)` |

### Die RootOf-Darstellung

Im folgenden Beispiel möchten wir das Gleichungssystem

$$\begin{aligned} x^2+y^2+z^2 &= 1 \\ -3/2+4z &= y \\ 3-5z &= x \end{aligned}$$

lösen. Dazu geben wir folgende `MAPLE`-Befehle ein:

```
s:=solve({x^2+y^2+z^2 = 1, y = -3/2+4*z, x = 3-5*z},{x,y,z});
```

und erhalten die folgende `MAPLE`-Ausgabe:

$$\begin{aligned} &s := \{z = 1/2\, RootOf\left(41 - 84\_Z + 42\_Z^2, label = \_L2\right), \\ &y = -3/2 + 2\, RootOf\left(41 - 84\_Z + 42\_Z^2, label = \_L2\right), \\ &x = 3 - 5/2\, RootOf\left(41 - 84\_Z + 42\_Z^2, label = \_L2\right)\} \end{aligned}$$

Dies ist die sog. `RootOf`-Darstellung. Setzt man die Umgebungsvariable `_EnvExplicit` auf den Wert `true`, dann erhält man die gewohnte algebraische Darstellung:

```
_EnvExplicit := true;
s:=solve({x^2+y^2+z^2 = 1, y = -3/2+4*z, x = 3-5*z},{x,y,z});
```

ergibt:

$$\begin{aligned} &s := \left\{z = 1/2 + \tfrac{1}{84}\sqrt{42}, y = 1/2 + 1/21\sqrt{42}, x = 1/2 - \tfrac{5}{84}\sqrt{42}\right\}, \\ &\left\{z = 1/2 - \tfrac{1}{84}\sqrt{42}, y = 1/2 - 1/21\sqrt{42}, x = 1/2 + \tfrac{5}{84}\sqrt{42}\right\}. \end{aligned}$$

### 6.2.2. MAPLE: Lösen von Ungleichungen

Mit `MAPLE` können nicht nur Lösungen von Gleichungen, sondern auch Lösungen von Ungleichungen bestimmt werden.

```
solve(3*x >= 12);
```

$$\mathit{RealRange}\,(4, \infty)$$

Oder:

```
solve(3*x > 12);
```

$$\mathit{RealRange}\,(\mathit{Open}\,(4)\,, \infty)$$

Was besagt die Ausgabe von `MAPLE`?

### 6.2.3. MAPLE: Lösen von Betragsgleichungen

Auch Betragsgleichungen können mit `MAPLE` gelöst werden.

```
solve(abs(x+1) = 12,x);
```

$$11, -13$$

### 6.2.4. MAPLE: Faktorenzerlegung

Wir betrachten zunächst das Ausmultiplizieren von Polynomen mit dem Befehl `expand`.

```
expand( (a+b) * (a-b));
```

$$a^2 - b^2$$

Sollen einzelne Teilausdrücke nicht ausmultipliziert werden, so müssen sie als weitere Argumente des `expand`-Befehls angegeben werden:

```
expand( (a+b) * (a-b), a+b );
```

$$(a + b)\, a - (a + b)\, b$$

Mittels `factor` werden polynomiale Ausdrücke in Linearfaktoren zerlegt:

```
factor( a^2 - b^2 );
```

$$(a + b)\,(a - b)$$

Das Polynom $x^2 - 2$ lässt sich nicht direkt mit dem `factor` Befehl in Linearfaktoren zerlegen.

```
factor( x^2 - 2 );
```

$$x^2 - 2$$

Hintergrund: Die Koeffizienten des zu zerlegenden Polynoms bestimmen bei der Anwendung des `factor` Befehls den Zahlenbereich der in den Linearfaktoren enthaltenen Nullstellen. Beachten Sie das Ergebnis, wenn wir `2` durch `2.` ersetzen:

```
factor( x^2 - 2. );
```

$$(x + 1.414213562)\,(x - 1.414213562)$$

Ist zudem eine Nullstelle bekannt, so kann diese als weiteres Argument an `factor` übergeben werden. Damit funktioniert die Linearfaktorzerlegung.

```
factor( x^2 - 2, sqrt(2) );
```

$$-\left(x+\sqrt{2}\right)\left(-x+\sqrt{2}\right)$$

Kennen Sie die Nullstellen nicht, können Sie eine Linearfaktorzerlegung durchführen, indem Sie die Option `real` angeben:

```
factor( x^2 - 2, real );
```

$$(x+1.414213562)\,(x-1.414213562)$$

Sie erhalten das Ergebnis als (ungenaue) Fließkommazahlen. Daher empfiehlt sich eine Berechnung der Nullstellen mit dem `solve` Befehl, bevor `factor` angewendet wird.

## 6.3. Modulare Arithmetik

Modulare Arithmetik kann auch mit `MAPLE` durchgeführt werden. Das Beispiel aus dem Skript (Konen, 2005) kann direkt in `MAPLE` eingegeben werden:

```
(117 * 76 + 303 ) mod 5;
```

$$0$$

## 6.4. Binomialkoeffizient und Summenzeichen

Auch Summen können einfach in `MAPLE` mit dem `sum` Befehl berechnet werden. Wir betrachten wieder das Beispiel aus Konen (2005):

$$s := \sum_{k=1}^{50} k^4$$

In `MAPLE` kann die Berechnung wie folgt durchgeführt werden:

```
s:= sum(k^4, k=1..50);
```

$$s := 65666665$$

Summen können mit `combine` zusammengefasst werden.

```
s := sum(a(k), k=m..n) + sum(a(k), k=n+1 .. p);
```

$$\sum_{k=m}^{n} a\left(k\right)+\sum_{k=n+1}^{p} a\left(k\right)$$

```
combine(s);
```

$$\sum_{k=m}^{p} a\left(k\right)$$

Fakultäten können mit dem Operator `!` oder mit der Funktion `factorial` berechnet werden. Der Binomialkoeffizient $\binom{a}{b}$ kann mittels `binomial` berechnet werden.

```
binomial(49,6) ;
```

$$13983816$$

# 7. Zahlenfolgen

## 7.1. Definition und Eigenschaften von Folgen

### 7.1.1. MAPLE: Folgen

Eine Folge (Sequenz) ist eine geordnete Menge von Ausdrücken. Wir werden zunächst nur Zahlenfolgen betrachten. Wie bereits in Abschnitt 5.2 dargestellt wurde, wird in `MAPLE` eine Folge mit dem Kommando `seq` gebildet, z.B.:

```
fn:= seq(i^2, i=1..10);
```

```
fn := 1, 4, 9, 16, 25, 36, 49, 64, 81, 100
```

### 7.1.2. MAPLE: Schaubilder von Zahlenfolgen

Um die Folge grafisch darzustellen, wird eine Liste mit den Datenpunkten definiert.

```
data1:= [seq([i,fn[i]], i=1..10)];
```

```
data1 := [[1, 1], [2, 4], [3, 9], [4, 16], [5, 25], [6, 36], [7, 49], [8,
64], [9, 81], [10, 100]]
```

Diese Liste mit dem Namen `data` kann analog zu den Funktionen dargestellt werden. Als Darstellungsart wird `style=point` gewählt, siehe Abbildung 7.1.

```
plot(data1, style=point);
```

Folgen können in `MAPLE` auch explizit durch Angabe einer Funktionsvorschrift definiert werden:

```
a := n -> n/(n+1);
```

$$n \mapsto \frac{n}{n+1}$$

Dann können einfach die Werte der $k$-ten Glieder ermittelt werden:

```
a(1); a(100);
```

$$1/2, 100/101$$

Wenden wir auf `a(n)` den `seq` Befehl an, so erhalten wir die Teilfolge:

```
b := seq( a(n), n=1 .. 5);
```

$$b := 1/2, 2/3, 3/4, 4/5, 5/6$$

### 7.1.3. MAPLE: Grenzwert einer Zahlenfolge

Grenzwerte von Folgen werden mit dem Kommando `limit` bestimmt. Geben Sie die folgenden Kommandos in Maple ein:

```
folge:= (3*n+1)/(n+1);
ersten_Glieder:= seq(folge,n=1..10);
Grenzwert:= limit(folge, n=infinity):
print("Grenzwert für n->oo: ", Grenzwert);
```

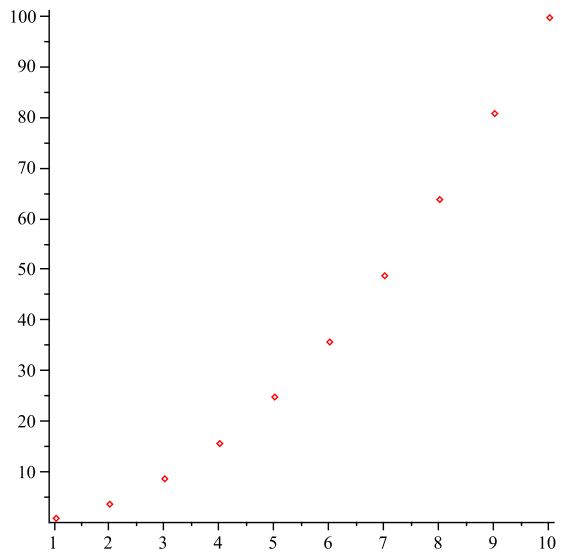


Abbildung 7.1.: Schaubild einer Folge von Datenpunkten

Für den Grenzwert erhalten Sie dann das Ergebnis:

```
Grenzwert für n->oo:, 3
```

### 7.1.4. MAPLE: Divergente Zahlenfolgen

Die Folge

```
a:= n -> (-1)^n*n/(n+2);
```

$$a := n \mapsto \frac{(-1)^n n}{n+2}$$

besitzt keinen Grenzwert, aber zwei Häufungspunkte: $\pm 1$.

```
limit (a(n), n=infinity);
```

$$-1..1$$

Das Schaubild (siehe Abb. 7.2) erstellen wir folgendermaßen:

```
pts := seq([n, a(n)], n=1..20): n='n':
plot([pts], style = point, color = red);
```

### 7.1.5. MAPLE: Teilfolgen

Mit dem Befehl `unapply` wird aus einem Ausdruck eine *anonyme Funktion* gebildet. Anonyme Funktionen haben im Gegensatz zu normalen Funktionen (wie z.B. $f(x)$, $g(x)$) keinen Namen.

```
unapply(x+2,x);
```

$$x \mapsto x + 2$$

Der anonymen Funktion kann ein Name zugewiesen werden, z.B. wie folgt:

```
f := %;
```

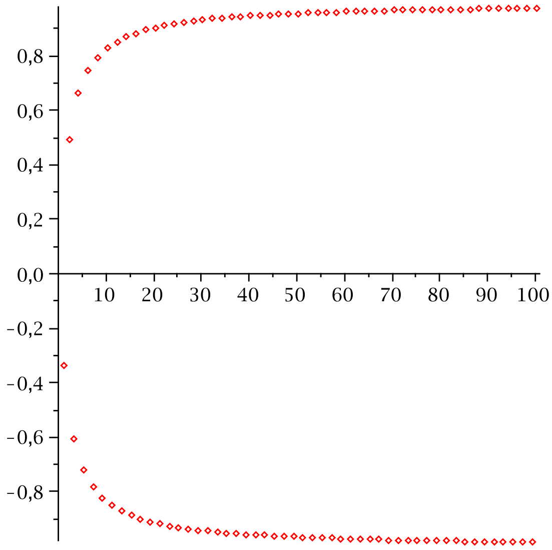


Abbildung 7.2.: Schaubild einer divergenten Folge von Datenpunkten

$$f := x \mapsto x + 2$$

Mit `unapply` erzeugen wir nun eine Teilfolge der Folge $a_n$:

```
a:= n -> (-1)^n*n/(n+2);
a2n := unapply (a(2*n), n);
```

$$a := n \mapsto \frac{(-1)^n n}{n+2}$$
$$a2n := n \mapsto 2\,\frac{(-1)^{2\,n} n}{2\,n+2}$$

Das Schaubild (siehe Abb. 7.3) erstellen wir folgendermaßen:

```
pts := seq([n, a2n(n)], n=1..100): n='n':
plot([pts], style = point, color = red);
```

## 7.1.6. MAPLE: Rekursive Folgen

### Fibonacci-Folge

Wir betrachten nun rekursive Folgen. Eines der bekanntesten Beispiele für rekursive Folgen sind die Fibonacci-Zahlen. Die Fibonacci-Folge wird über zwei Vorgängerwerte definiert:
`a[0]:=0; a[1]:=1;`
Die weiteren Folgenglieder werden rekursiv definiert:

```
for n from 1 to 10 do
   a[n+1]:= evalf (a[n]+a[n-1]);
od;
```

Die folgende Prozedur berechnet die Fibonacci-Zahlen ebenfalls. Dabei wird die Prozedur rekursiv aufgerufen.

```
f:=proc(x::nonnegint)
   option remember;
   if x=0 then 0
```

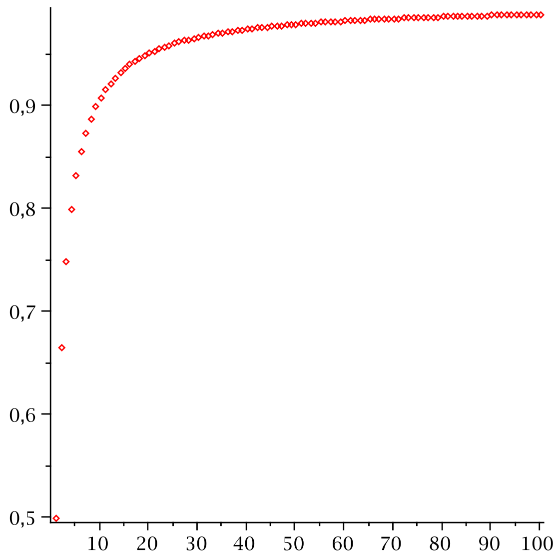


Abbildung 7.3.: Schaubild einer Teilfolge

```
          elif x=1 then 1
                   else f(x-1)+f(x-2)
   fi
end;
```

Die Option `remember` dient nur zur schnelleren Berechnung, die wiederholt benötigten Zwischenergebnisse werden zwischengespeichert.
`f(50)`

```
12586269025
```

### Quadratwurzel

Als weiteres Beispiel für eine Folge betrachten wir die Berechnung einer Quadratwurzel. Die Quadratwurzel einer vorgegebenen Zahl $p$ kann durch eine rekursive Folge mit der Vorschrift

$$a_n := \frac{1}{2}\,a_{n-1} + \frac{1}{2}\,\frac{p}{a_{n-1}}$$

ermittelt werden. Dabei ist $a_0$ ein vorgegebener Startwert, z.B. 1, und die weiteren Iterierten $a_n$ werden nacheinander berechnet.

Man teste für verschiedene $p$ mit dem folgenden Programm, dass die Folge `a[n]` gegen die Wurzel von $p$ konvergiert.

```
a[0]:=1;
p:= 100;
for n from 1 to 10 do
   a[n]:= evalf ( 1/2 * (a[n-1]+p/a[n-1]) );
od;
print("Vergleich",sqrt(100),a[10]);
```

Warum konvergiert die Folge gegen die Quadratwurzel von p? Wir wenden auf beide Seiten der Definitionsgleichung einen Grenzübergang (lim) an und bezeichnen den Grenzwert mit $\alpha$, also $\alpha = \lim(a[n])$. Dann folgt die Beziehung:

```
p:='p': gl:= alpha = 1/2 * (alpha+p/alpha);
```
Diese Gleichung hat die folgenden Lösungen:
```
solve(gl,alpha);
```

$$\sqrt{p}, -\sqrt{p}$$

### Träge Formen

Wird der Anfangsbuchstabe eines Ausdrucks groß geschrieben, dann liegt eine sog. *träge Form* vor. Das Kommando `Limit` ist z.B. eine *träge Form* des Ausdrucks `limit` und kann zur besseren Darstellung benutzt werden. Liegt ein Ausdruck als träge Form vor, so wird er nicht ausgewertet, sondern nur formatiert dargestellt. Viele Kommandos besitzen eine träge Form, z.B. auch `sum`, `diff` etc.
```
Limit(folge, n=infinity) = limit(folge, n=infinity);
```
Zum Beispiel giltfür die oben definiterte divergente Folge $a_n$:
```
Limit(a(n), n=infinity) = limit(a(n), n=infinity);
```

$$\lim_{n\to\infty} \frac{(-1)^n n}{n+2} = -1 - I \ldots 1 + I$$

# 8. Reelle Funktionen

## 8.1. Allgemeine Funktionseigenschaften

### 8.1.1. MAPLE: Nullstellen einer Funktion

Gesucht sind die Nullstellen der Funktion $f(x)$:

$$f(x) = 0.$$

MAPLE bietet hierzu die Befehle `solve` und `fsolve` an. `solve` berechnet die Lösungen algebraisch, d.h. durch Umformungen. Es werden dann u.U. mehrere Lösungen ausgegeben. So besitzt die Gleichung $y = x^2$ zwei Lösungen. Manche Gleichungen sind jedoch nicht algebraisch lösbar. Dann greift man auf numerische Verfahren zurück. Um numerisch die Nullstellen der Funktion $f(x) = \sqrt{x} - 4x^2$ mit MAPLE zu bestimmen, geht man folgendermaßen vor:

```
f(x) := sqrt(x) - 4*x^2;
fsolve(f(x) = 0, x);
```

$$0$$

`fsolve` berechnet im Gegensatz zu `solve` allerdings nur eine Lösung. Um weitere Nullstellen zu berechnen, kann ein Intervall angegeben werden, in dem die Nullstellen näherungsweise berechnet werden sollen:

```
fsolve(f(x) = 0, x, x=0.1..2);
```

$$0.3968502630$$

### 8.1.2. MAPLE: Darstellung und Animation von Funktionen

MAPLE ist sehr geeignet um Funktionen grafisch darzustellen. Dazu folgen einige Beispiele.

- Ein Polynom vom Grad 5 mit 5 reellen Nullstellen, siehe Abbildung 8.1 (links):
  ```
  plot(x^5-5*x^3+4*x, x=-3..3, y=-6..6, title="x^5-5*x^3+4*x");
  ```

- Quotient zweier Polynome, der Zähler ist das Polynom aus Abbildung 8.1, der Zählergrad ist kleiner als der Nennergrad (Abbildung 8.1, rechts):
  ```
  plot((x^5-5*x^3+4*x)/(x^8+1), x=-3..3, y=-2..2, title="(x^5-5*x^3+4*x)/(x^8+1)");
  ```

- Quotient zweier Polynome, der Zähler ist das Polynom aus Abbildung 8.1, der Zählergrad ist größer als der Nennergrad, das Polynom im Nenner hat reelle Nullstellen Die Option `discont=true` dient zur Überbrückung der Unstetigkeiten.
  ```
  plot((x^5-5*x^3+4*x)/(x^4-1.5^4),x=-5..5,y=-4..4, discont=true);
  ```

Nun betrachten wir noch eine Animation. Nach Eingabe der folgenden Kommandos in MAPLE können Sie die Grafik anklicken und die Animation starten. Der Parameter $a$ läuft dabei von 1 nach 25. Durch die Option `frames=25` werden 25 Einzelbilder erzeugt (standardmäßig 16).

```
with(plots):
animate (sin(a*x)/a, x=0..2*Pi, a=1..25, thickness=3,
numpoints=100, frames=25, title="sin(a*x)/a");
```

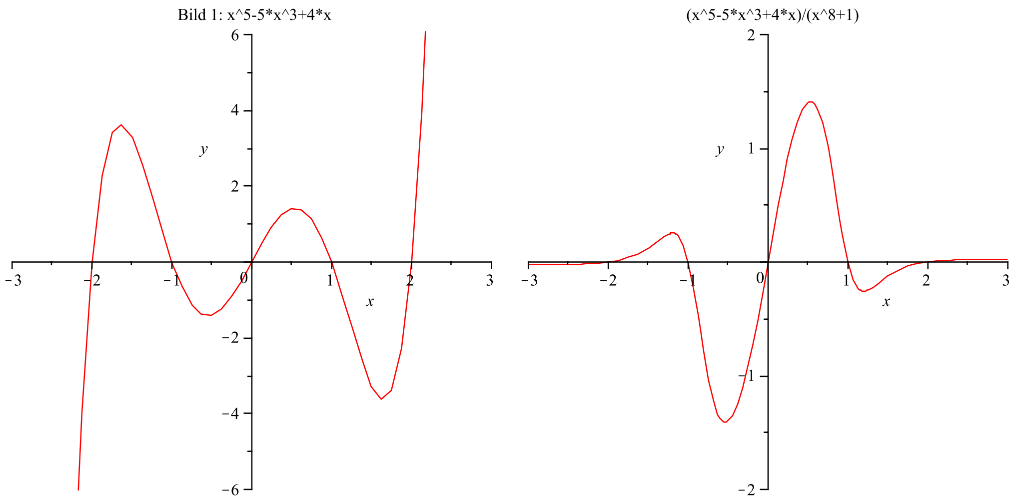


Abbildung 8.1.: Links: Graph eines Polynoms. Rechts: Graph des Quotienten zweier Polynome

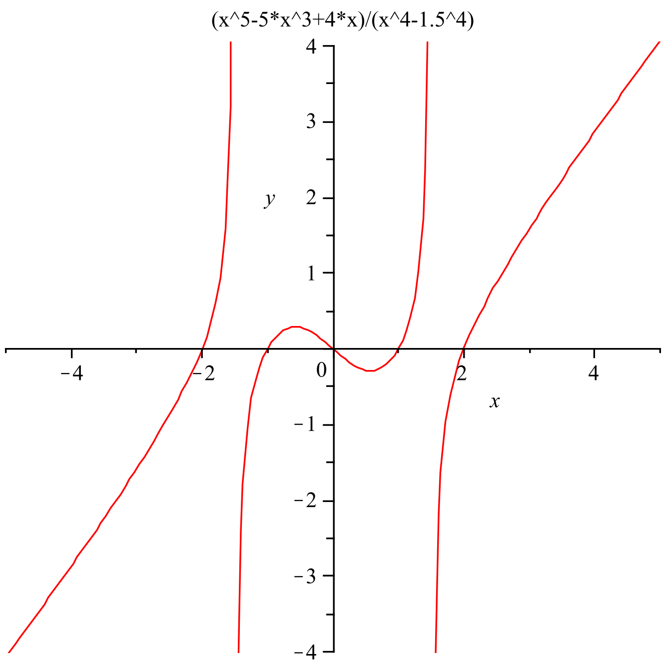


Abbildung 8.2.: Graph des Quotienten zweier Polynome

### 8.1.3. MAPLE: Umkehrfunktionen

Mit MAPLE lassen sich Umkehrfunktionen bestimmen, z.B. von der Funktion

$$y(x) := (1 - e^x)/(1 + e^x).$$

```
e:=exp(1);
gl:= y=(1-e^x)/(1+e^x);
solve(gl,x);
```

$$\ln\left(-\frac{y-1}{y+1}\right)$$

In dieser Ausgabe sind die Variablen $x$ und $y$ noch nicht vertauscht. Beachten Sie, dass unter Umständen keine eindeutige Lösung der Gleichung gl existiert. Das folgende Skript berücksichtigt einige Probleme, die bei der Bestimmung der Umkehrfunktion auftreten können. Es erzeugt zudem die Schaubilder von $f$, der Umkehrfunktion $g = f^{-1}$ und der ersten Winkelhalbierenden. Wir betrachten dazu die Berechnung der Umkehrfunktion von $f(x) = \exp(x^2)$:

```
restart:
f := x -> exp(x^2);
gl := y = f(x):
s1:= [solve(gl,x)];
if nops(s1) > 1 then
printf("Warnung. Es gibt %d Loesungen.
  Es wird die erste Loesung aus der obigen
  Liste s1 gezeichnet.\n", nops(s1) );
end if;
s1 := s1[1]:
subs(y=x, s1):  y:= %:
printf("Die Umkehrfunktion g von f lautet:");
g := unapply(y,x);
x := 'x':
y := 'y':
plot([x, f(x) ,g(x)], x=-1..5, y=-1..5,thickness = 2,
  color = ["blue", "gray", "black"], legend=["id","f","g"],
  font=[HELVETICA, 14], legendstyle=[font= [HELVETICA,14]]);
```

$f := x \to \exp(x^2)$
s1 := $\left[\ln(y^{1/2}), -\ln(y^{1/2})\right]$
Warnung. Es gibt 2 Loesungen. Es wird die erste Loesung aus der obigen Liste s1 gezeichnet. Die Umkehrfunktion g von f lautet: $x \mapsto \sqrt{\ln(x)}$.

Die Schaubilder sind in Abbildung 8.3 dargestellt. Außerdem bietet MAPLE die Möglichkeit, den Befehl InversePlot aus dem Paket Student[Calculus1] zum Zeichnen des Schaubilds der Umkehrfunktion an. Das Ergebnis des Aufrufs von

```
with(Student[Calculus1]);
InversePlot(exp(x^2), -1 .. 1);
```

ist in Abbildung 8.3 dargestellt.

## 8.2. MAPLE: Grenzwert von Funktionen

Auch können in MAPLE Grenzwerte von Funktionen berechnet werden:

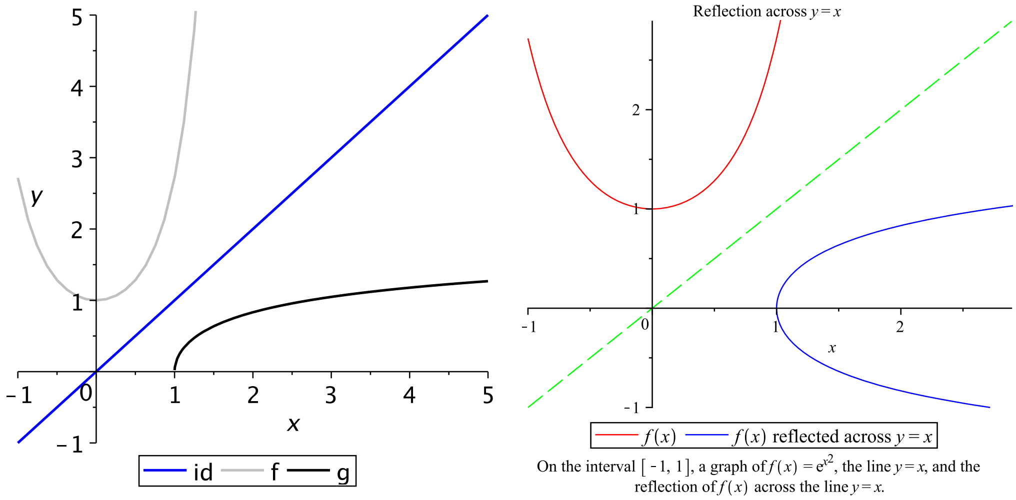


Abbildung 8.3.: *Links:* Die Funktion $f(x) = \exp(x^2)$ und ihre Umkehrfunktion $g(x) = \sqrt{\ln(x)}$. *Rechts:* Die mittels `InversePlot` erzeugte Grafik.

```
f:= x->(2*x+3)/(7*x+5);
Grenzwert:= limit(f(x), x=infinity):
print("Grenzwert für x->oo: ", Grenzwert);
bzw.
Limit(f(x), x=infinity) = limit(f(x), x=infinity);
```

$f := x \mapsto \frac{2\,x+3}{7\,x+5}$

`Grenzwert für x->oo:, 2/7`

$\lim_{x\to\infty} \frac{2\,x+3}{7\,x+5} = 2/7$

**MAPLE: Rechts- und linksseitige Grenzwerte**

Auch links- und rechtsseitige Grenzwerte können bestimmt werden. Wir betrachten die Grenzwerte der Tangensfunktion an der Stelle $\pi/2$.

```
f:= x->tan(x);
limit(f(x), x=Pi/2, left);
limit(f(x), x=Pi/2, right);
```

$f := x \mapsto \tan(x)$

$\infty$

$-\infty$

## 8.3. MAPLE: Kurvendiskussion

Anhand eines Beispiels führen wir eine Kurvendiskussion mit MAPLE durch. Dazu zeichnen wir das Schaubild, untersuchen die Symmetrie, Nullstellen und das Verhalten im Unendlichen. Die Bestimmung der Extrema und Wendepunkte besprechen wir, wenn die Differentialrechnung eingeführt wurde.

**Beispiel 8.1 (Gebrochenrationale Funktion)**
*Gegeben sei die gebrochenrationale Funktion*

$$f(x) = \frac{x^3}{x^3 + 1}$$

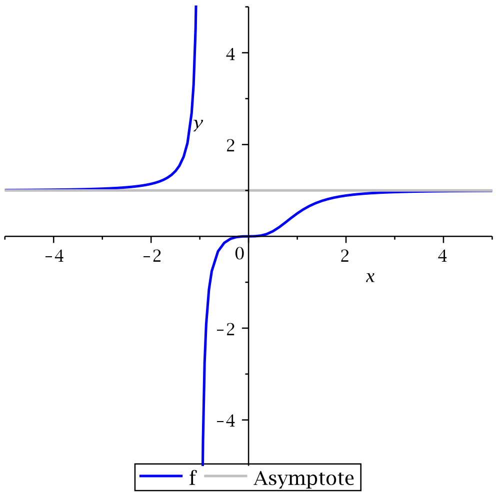


Abbildung 8.4.: Schaubild und Asymptote der Funktion $f(x) = x^3/(x^3+1)$.

***Definition*** *In* `MAPLE` *definieren wir:*
`f:= x -> x^3/(x^3+1):`

***Nullstellen*** *Die Nullstellen ermitteln wir mit dem* `solve` *Befehl.*
`solve(f(x) = 0,x);`

***Symmetrie*** *Hier sind die Gleichungen $f(x) = f(-x)$ und $f(x) = -f(-x)$ zu untersuchen. Hierzu benutzen wir den* `simplify` *Befehl, der den entsprechenden Quotienten vereinfacht:*
`simplify(f(x)/f(-x), symbolic);.`

***Asymptotisches Verhalten*** *Das Verhalten für große $x$-Werte bestimmen wir mit dem* `asympt` *Befehl. Dieser Befehl wird mit den folgenden Parametern aufgerufen:*
`asympt(f(x), x, n);`
*wobei* `f(x)` *die Funktion,* `x` *die unabhängige Variable der Funktion und* `n` *die Entwicklung nach den Termen $\frac{1}{x}$ bis $\left(\frac{1}{x}\right)^n$ darstellt. Soll die asymptotische Gerade an eine Funktion berechnet werden, so setzt man $n = 1$. Die* `MAPLE`*-Standardeinstellung $n = 6$ ist für uns eher nicht so sehr brauchbar.*

***Schaubild*** *Der* `MAPLE` *Befehl* `plot( f(x) );` *liefert das Schaubild.*

*Zusammenfassend sind die* `MAPLE`*-Befehle in Abbildung 8.5 dargestellt. Es wird damit die folgende Ausgabe erzeugt:*

$x \mapsto \frac{x^3}{x^3+1}$

```
Nullstelle bei x= 0
Nullstelle bei x= 0
Nullstelle bei x= 0
Die Funktion ist nicht symmetrisch zum Ursprung oder zur y-Achse.
Asymptotengleichung: y= 1
```

*Das Schaubild ist in Abbildung 8.4 dargestellt.* □

```
f := x -> x^3/(x^3+1);
# Nullstellen:
h:= x->numer(f(x)):
wurzel := [solve(h(x)=0,x)]:
nwurzel := nops(wurzel):
for i to nwurzel do
print("Nullstelle bei x=", wurzel[i]);
od;
#Symmetrie:
if (simplify(-f(x)/f(-x)) = 1)
then print("Die Funktion ist symmetrisch zum Ursprung.")
else
if (simplify(f(x)/f(-x)) = 1)
then print("Die Funktion ist symmetrisch zur y-Achse.")
else
print("Die Funktion ist nicht symmetrisch zum Ursprung oder zur y-Achse.");
end if;
end if;
#Asymptoten:
asym:=asympt(f(x), x,1):
a2:= convert(asym,polynom):
print("Asymptotengleichung: y=",a2);
plot([f(x) ,a2(x)], x=-5..5, y=-5..5, thickness = 2, color = ["blue", "gray"],
legend=["f","Asymptote"],  discont = true);
```

Abbildung 8.5.: Maple-Aufrufe zur Kurvendiskussion einer gebrochenrationalen Funktion. Der Befehl `numer` stellt den Nenner $h$ der gebrochenrationalen Funktion $f = g/h$ dar, `nops` berechnet die Länge einer Liste (hier die Liste der Nullstellen), `convert` verwandelt einen Ausdruck in ein Polynom und `discont` versucht, Unstetigkeitsstellen (Polstellen) beim Zeichnen zu berücksichtigen.

# Teil III.

# Analysis I: Differentialrechnung

# 9. Differentialrechnung

## 9.1. Übersicht

Dieser Teil führt in die folgenden Bereiche ein:

- Differentialrechnung
- Mittelwertsatz
- Taylorpolynome
- L'Hospital
- Vollständige Kurvendiskussion

## 9.2. Differenzierbarkeit, Ableitung, Differential, Ableitungsregeln

### 9.2.1. Der Ableitungsbegriff

Wir betrachten die Funktion

$$f(x) = x^2 + 1 \tag{9.1}$$

und ihre Tangente im Punkt $P(0|1)$ sowie einige Sekanten durch den Punkt $P$ und einen durch $h$ festgelegten weiteren Punkt $Q(h|h^2+1)$. `MAPLE` stellt mit dem **`Student`** Paket umfangreiche Hilfestellungen und Visualisierungen zur Differentialrechnung zur Verfügung, die wir im Folgenden betrachten. Geben Sie dazu die folgenden Befehle ein:

```
with(Student[Calculus1]):
f := x -> x^2+1;
```

Nun können durch Eingabe des Befehls `NewtonQuotient` Sekanten für unterschiedliche $h$-Werte gezeichnet werden: Starten Sie mit $h = 1$, indem Sie

```
NewtonQuotient(f(x), x=0, 'h'=1, -1..1, output=plot, title="Sekante fuer h=1");
```

eingeben (siehe Abbildung 9.1 links). Für $h = 0.5$ erhalten Sie das in Abbildung 9.1 rechts gezeigte Schaubild. Für kleiner werdende $h$-Werte nähern sich die Sekanten an die Tangente im Punkte $P$ an. Diesen Vorgang können Sie in `MAPLE` auch als Animation darstellen. Geben Sie dazu den Befehl

```
NewtonQuotient(f(x), x=0, 'h'=1, -1..1, output=animation,
   title="Sekanten fuer versch. h");
```

ein.

### 9.2.2. Numerische Differentiation

Eine Näherung der Ableitung eines Ausdrucks $f(x)$ an einer Stelle $x_0$ kann wie folgt numerisch berechnet werden:

$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0-h)}{2h}. \tag{9.2}$$

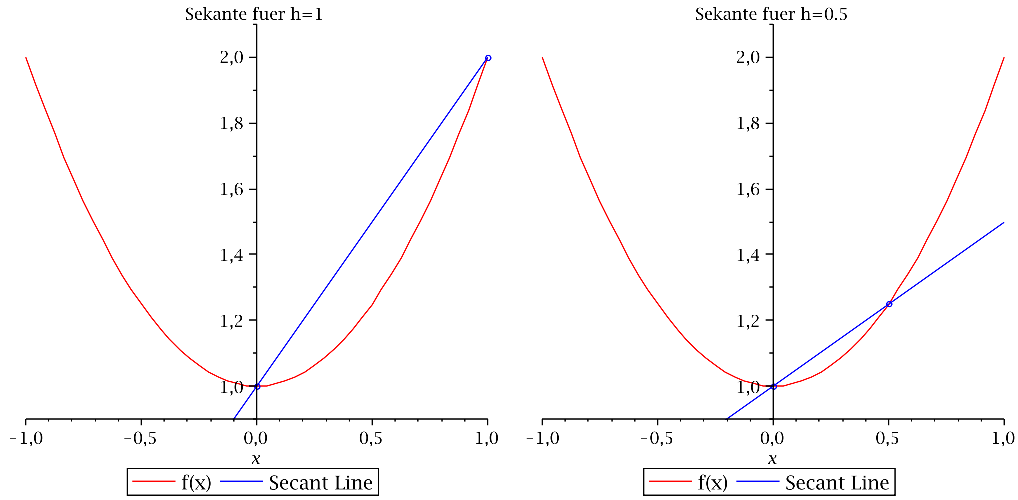


Abbildung 9.1.: Sekante durch $P(0|1)$ und $P(h|h^2+1)$ für $h=1$ (links) und $h=0.5$ (rechts).

Wir betrachten die Funktion $f(x)=x^3$ an der Stelle $x_0=2$. Die Schrittweite sei $h=0.1$

```
f := x -> x^3: x0 := 2: h := 0.1:
```

Dann erhalten wir den Näherungswert

```
(f(x0+h)-f(x0-h)) / (2*h);
```

$$12.010000000$$

Der exakte Wert lautet 12.

### 9.2.3. Ableitungsoperatoren in Maple

`MAPLE` unterscheidet zwischen Ausdrücken und Funktionen. Entsprechend gibt es zwei verschiedene Ableitungsoperatoren:

**diff** leitet einen Ausdruck ab und liefert als Ergebnis einen Ausdruck

**D** leitet eine Funktion ab und liefert als Ergebnis wieder eine Funktion.

#### Ableitung eines Ausdrucks mit diff

Mit `MAPLE` können Ableitungen bestimmt werden. Wir bestimmen die Ableitung eines Ausdrucks $f(x)$ in einer Variablen $x$, also

$$\frac{d}{dx}f(x),$$

indem der Befehl `diff(f(x),x)` eingeben wird. Um den Ausdruck $f(x)=x^2+1$ abzuleiten, geben wir die Befehle

```
f(x) := x^2+1:
Diff(f(x),x)=diff(f(x),x);
```

ein und erhalten

$$\frac{d}{dx}\left(x^2+1\right)=2\,x$$

Höhere Ableitungen können mit dem `$` Operator berechnet werden. So können die zweiten und dritten Ableitungen mittels `diff(f(x), x$2)`, `diff(f(x), x$3)` usw. berechnet werden:

```
Diff(f(x),x$2) = diff(f(x),x$2);
```

ergibt:

$$\frac{d^2}{dx^2}\left(x^2+1\right)=2$$

`Diff(f(x),x$3) = diff(f(x),x$3);`
ergibt:

$$\frac{d^3}{dx^3}\left(x^2+1\right)=0$$

#### Ableitung einer Funktion mit D

Wir betrachten nun die Ableitung einer Funktion mit `D`. Um die Funktion `g:=x->x^2+1` abzuleiten, geben wir den Befehl
`D(g);`
ein und erhalten

$$x \mapsto 2\,x$$

Die Ableitung der Funktion `g` an der Stelle $x = 2$ kann mit
`D(g)(2);`
berechnet werden:

$$4$$

Die zweite Ableitung der Funktion `g` nach `x` kann wie folgt berechnet werden:

```
(D@@2)(g);
```

$$2$$

Somit können wir z.B. $h(x) = \exp(2x)$ einhundert Mal nach $x$ ableiten:

```
h:=x->exp(2*x);
(D@@100)(h);
```

$$x \mapsto 1267650600228229401496703205376\,e^{2\,x}$$

Zur Kontrolle können wir $2^{100}$ berechnen.

$$1267650600228229401496703205376$$

#### Zusammenhang von D und diff

Es gilt
`D(f)(x) = diff(f(x),x)`
`D` und `diff` liefern hier dasselbe Ergebnis, da die Funktion, die durch den `D`-Operator zurückgeliefert wird, an der Stelle `x` ausgewertet wird. Der `D`-Operator kann somit den `diff`-Operator simulieren.

## 9.3. Mittelwertsatz

### 9.3.1. Beispiel aus dem Mathematik Skript

Wir betrachten eine auf einem Intervall $\mathcal{I} := [a; b]$ stetige und im Inneren von $\mathcal{I}$ differenzierbare Funktion $f$. Der Mittelwertsatz der Differentialrechnung (engl. Mean Value Theorem) besagt, dass die Steigung der Sekante von $a$ nach $b$, die sog. mittlere Steigung, an mindestens einer Stelle $u \in ]a; b[$ von der Ableitung angenommen wird.

`MAPLE` bietet verschiedene Tools an, die wichtige Sachverhalte aus der Analysis grafisch aufbereiten und veranschaulichen. Diese befinden sich in den `Student` Paketen. Wir betrachten das Paket zur Erläuterung des Mittelwertsatzes. Nach Eingabe von

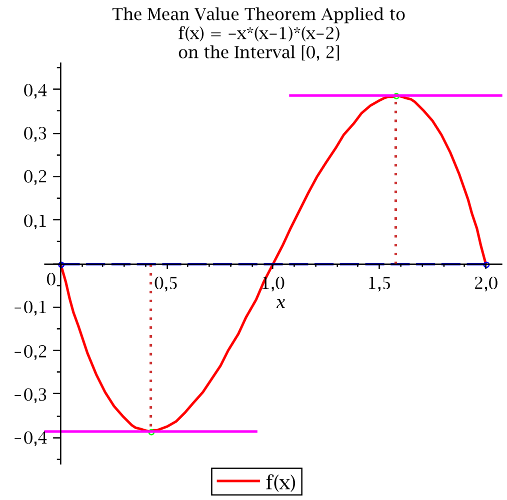


Abbildung 9.2.: Schaubild der Funktion $-x(x-1)(x-2)$ zur Erläuterung des Mittelwertsatzes.

```
restart:
with(Student[Calculus1]):
```

stehen die entsprechenden Befehle zur Verfügung. Konen (2005) betrachtet z.B. die Funktion $f(x) = -x(x-1)(x-2)$.

```
MeanValueTheorem(-x * (x-1)*(x-2), x=0..2, output = points);
```

ergibt

$$[1 - (1/3) * \sqrt{(}3), 1 + (1/3) * \sqrt{(}3)]$$

bzw. mittels `evalf` die Werte

$$[.4226497307, 1.577350269]$$

Dies sind die Stellen, an denen die Ableitung der Funktion $f$ verschwindet. Mit dem Befehl

```
MeanValueTheorem(-x*(x-1)*(x-2), x=0..2);
```

kann eine ähnliche Abbildung wie im Mathematik-Skript (Konen, 2005) erzeugt werden, siehe Abbildung 9.2.

### 9.3.2. Tutoren

Weitere interaktive Beispiele, sog. MAPLE-Tutoren, erhalten Sie durch folgende Eingabe:

```
with(Student[Calculus1]):
MeanValueTheoremTutor(x*sin(x));
MeanValueTheoremTutor(x*sin(x), -2*Pi..2*Pi );
MeanValueTheoremTutor(x*sin(x), x=-2*Pi..2*Pi );
```

## 9.4. Satz von Taylor

### 9.4.1. Taylor-Entwicklung in MAPLE

Wir betrachten die Taylor-Entwicklung der Ordnung $N$ für eine Funktion $f(x)$ in einer Variablen $x$:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \ldots + \frac{1}{N!} f^{(N)}(x_0)(x - x_0)^{(N)}, \tag{9.3}$$

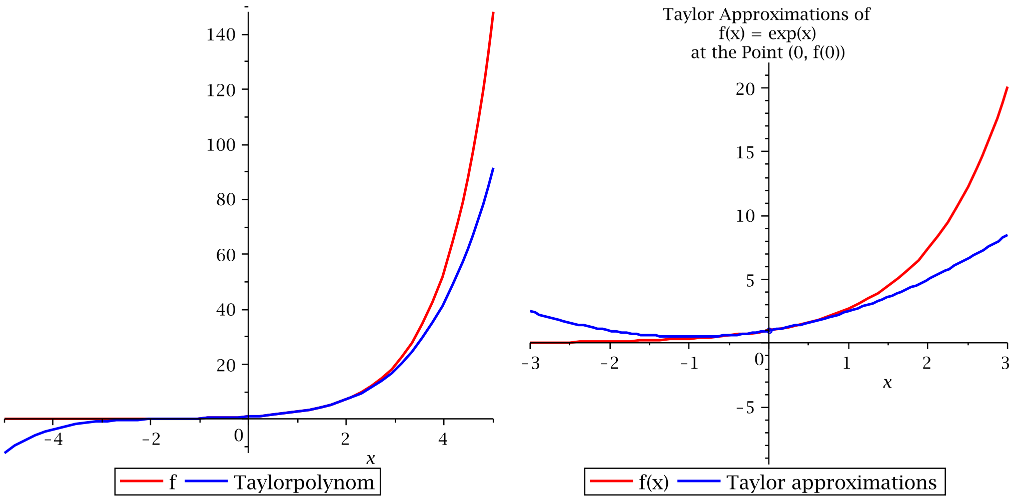


Abbildung 9.3.: Links: Taylor-Entwicklung einer Funktion. Rechts: Ausgabe, die mit dem Befehl `TaylorApproximation` aus dem `Student` Paket erzeugt wurde.

die mittels
`taylor(f(x), x=x0, N+1)`
in `MAPLE` berechnet werden kann. `f(x)` bezeichnet den Funktionsausdruck, `x=x0` ist der Entwicklungspunkt, `N` gibt die Ordnung der Taylor-Entwicklung an.

### 9.4.2. Entwicklung der $e$-Funktion

Wir betrachten die Entwicklung der Funktion $f(x) = \exp(x)$ an der Stelle $x_0 = 0$ bis zur Ordnung $N = 5$.

```
restart:
f:=x->exp(x):
taylor(f(x), x=0, 6); # Achtung: Ordnung 5, aber N+1=6
```

$$1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5 + O\left(x^6\right)$$

$O\left(x^6\right)$ bedeutet, dass Terme ab der Ordnung 6 vernachlässigt werden. Um das Taylor-Polynom zeichnen zu können (siehe Abbildung 9.3, links), wird der Ausdruck $1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5 + O\left(x^6\right)$ mit dem `MAPLE`-Befehl `convert` in ein Polynom umgewandelt.

```
p:=convert(%,polynom);
plot([f(x), p], x=-2..4, color = [red,blue]);
```

### 9.4.3. Animation der Taylor-Approximation

`MAPLE` bietet vielfache Visualisierungsmöglichkeiten für Taylor-Polynome an. So erhalten Sie z.B. durch Eingabe der folgenden Befehle

```
restart: with(Student[Calculus1]):
TaylorApproximation(exp(x), x=0, order=1..10, output=animation,
view=[-5..3, DEFAULT]);
```

eine animierte Darstellung der Entwicklung der Funktion $f(x) = \exp(x)$ an der Stelle $x_0 = 0$ bis zur Ordnung $N = 10$, siehe Abbildung 9.3 (rechts).

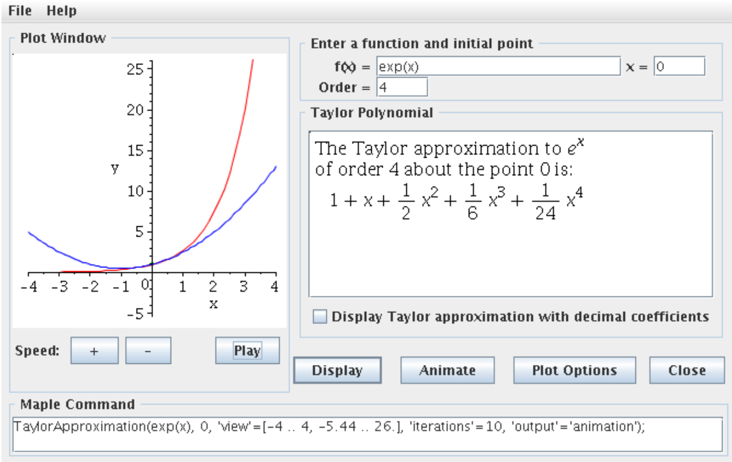


Abbildung 9.4.: Taylor-Entwicklung mit dem Taylor-Tutor aus dem `Student` Paket.

### 9.4.4. Taylor-Tutor

Sie können auch den `MAPLE`-Tutor heranziehen, um die Funktionsapproximation interaktiv zu visualisieren (siehe Abbildung 9.4). Geben Sie dazu die folgenden Befehle ein:

```
restart: with(Student[Calculus1]):
TaylorApproximationTutor(exp(x),x=0);
```

### 9.4.5. Taylorpolynome

Nachdem wir einige der von `MAPLE` bereitgestellten Hilfsmittel zur Berechnung von Taylorpolynomen vorgestellt haben, führen wir noch eine elementare Berechnung der Taylorpolynome durch. Es sollen die Taylorpolynome ersten bis bis fünften Grades von der Funktion

$$f(x) = \sin((x-1)\ln(x+1))$$

berechnet und graphisch dargestellt werden. Dazu geben wir die folgenden Befehle ein und erhalten das in Abbildung 9.5 gezeigte Schaubild.

```
restart: with(plots):
f[0]:= x -> sin((x-1) * ln(x+1));
pf:= plot(f[0](x), x=-0.9..5, -1..2, color=red, thickness=2):
display(pf);
```

Zur Berechnung der ersten fünf Ableitungen gehen wir folgendermaßen vor:

```
for i from 1 to 5 do
   f[i]:= D(f[i-1]):
od:
```

Wir erhalten z.B. für die erste Ableitung `f[1]`:

$$x \mapsto \cos\left((-1+x)\ln(1+x)\right)\left(\ln(1+x) + \frac{-1+x}{1+x}\right).$$

Die 5.Ableitung sieht schon relativ kompliziert aus:

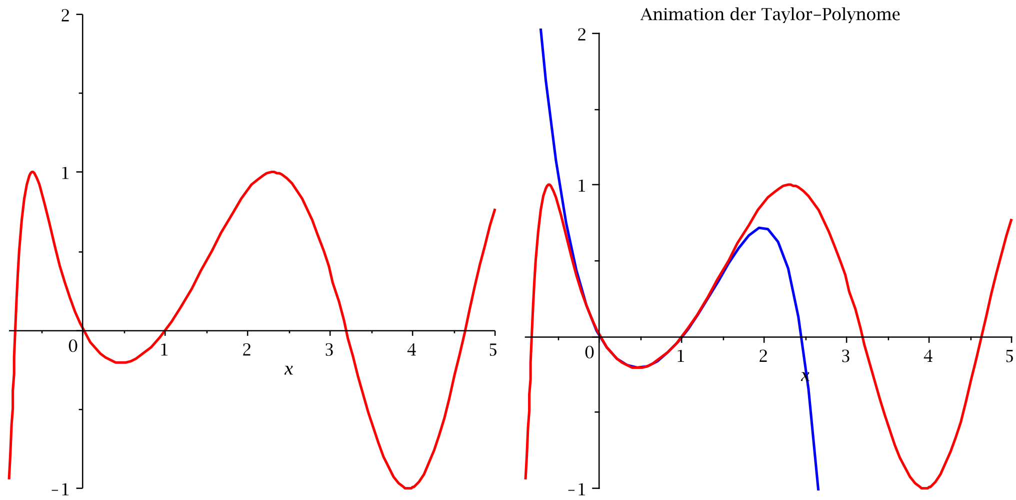


Abbildung 9.5.: Links: Funktion $f(x) = \sin((x-1)\ln(x+1))$. Rechts: Taylor-Entwicklung.

$$
\begin{aligned}
&x \mapsto \cos\left((-1+x)\ln(1+x)\right)\left(\ln(1+x)+\frac{-1+x}{1+x}\right)^5 \\
&+10\sin\left((-1+x)\ln(1+x)\right)\left(\ln(1+x)+\frac{-1+x}{1+x}\right)^3\left(2\,(1+x)^{-1}-\frac{-1+x}{(1+x)^2}\right) \\
&-15\cos\left((-1+x)\ln(1+x)\right)\left(\ln(1+x)+\frac{-1+x}{1+x}\right)\left(2\,(1+x)^{-1}-\frac{-1+x}{(1+x)^2}\right)^2 \\
&-10\cos\left((-1+x)\ln(1+x)\right)\left(\ln(1+x)+\frac{-1+x}{1+x}\right)^2\left(-3\,(1+x)^{-2}+2\,\frac{-1+x}{(1+x)^3}\right) \\
&-10\sin\left((-1+x)\ln(1+x)\right)\left(2\,(1+x)^{-1}-\frac{-1+x}{(1+x)^2}\right)\left(-3\,(1+x)^{-2}+2\,\frac{-1+x}{(1+x)^3}\right) \\
&-5\sin\left((-1+x)\ln(1+x)\right)\left(\ln(1+x)+\frac{-1+x}{1+x}\right)\left(8\,(1+x)^{-3}-6\,\frac{-1+x}{(1+x)^4}\right) \\
&+\cos\left((-1+x)\ln(1+x)\right)\left(-30\,(1+x)^{-4}+24\,\frac{-1+x}{(1+x)^5}\right)
\end{aligned}
$$

Die Taylorpolynome um den Entwicklungspunkt $x_0$ werden mittels

```
Taylor:= x-> sum((1/n!) * f[n](x0) * (x-x0)^n, n=0..m);
```

berechnet:

$$Taylor := x \mapsto \sum_{n=0}^{m} \frac{f_n(x0)(x-x0)^n}{n!}$$

Ist z.B. der Entwicklungspunkt $x_0 = 0.5$, dann lauten die Taylorpolynome ersten bis fünften Grades:

```
x0:= 0.5;
for m from 1 to 5 do
   evalf(Taylor(x));
od;
```

$$-0.2366739302 + 0.07065451928\,x$$

$$-0.04608074858 - 0.6917182072\,x + 0.7623727265\,x^2$$

$$0.01422978599 - 0.8828239827\,x + 1.144584278\,x^2 - 0.2548077007\,x^3$$

$$-0.003235865531 - 0.9707753464\,x + 1.408438368\,x^2 - 0.6066131553\,x^3 + 0.1759027273\,x^4$$

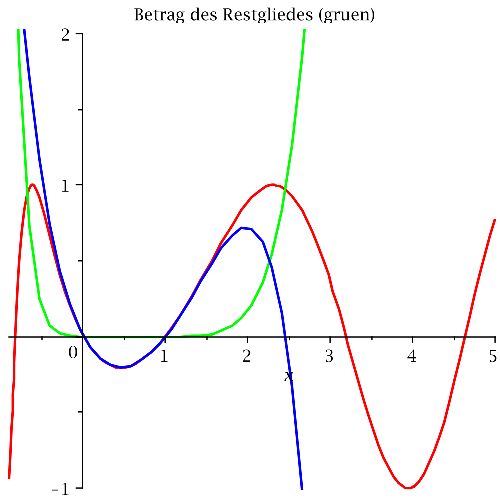


Abbildung 9.6.: Taylor-Approximation und Restglied.

$$0.0005985090688-1.009119092\,x+1.561813352\,x^2-0.9133631233\,x^3+0.4826526953\,x^4-0.1226999872\,x^5$$

Auch die Animation können wir selbst programmieren. Der Animationsparameter $m$ läuft von 1 bis 5, deshalb wird `frames=5` gesetzt:

```
with(plots):
m:='m':
animation:= animate(Taylor(x), x=-0.9..5, m=1..5, color=blue, thickness=2, frames=5):
display([animation,pf], title="Animation der Taylor-Polynome");
```

Wir erhalten die in Abbildung 9.5 (rechts) dargestellte Visualisierung.

#### Berechnung des Restglieds

Das Restglied, d.h. die Differenz zwischen der Funktion und dem Taylorpolynom vom Grad $m = 5$ ist:
`m:=5: Restglied:= x -> f[0](x) - Taylor(x);`
Zur Visualisierung benutzen wir die folgenden Befehle:

```
pd:= plot(abs(Restglied(x)), x=-0.9..5, color=green, thickness=2):
pt:= plot(Taylor(x), x=-0.9..5, color=blue, thickness=2):
display(pf,pd,pt, title="Betrag des Restgliedes (gruen)");
```

Die entsprechenden Schaubilder sind in Abbildung 9.6 dargestellt.

## 9.5. Regel von L'Hospital

### 9.5.1. Definition

Eine Variante der Regel von L'Hospital besagt, dass aus

$$\lim_{x\to a} f(x) = 0$$

und

$$\lim_{x\to a} g(x) = 0$$

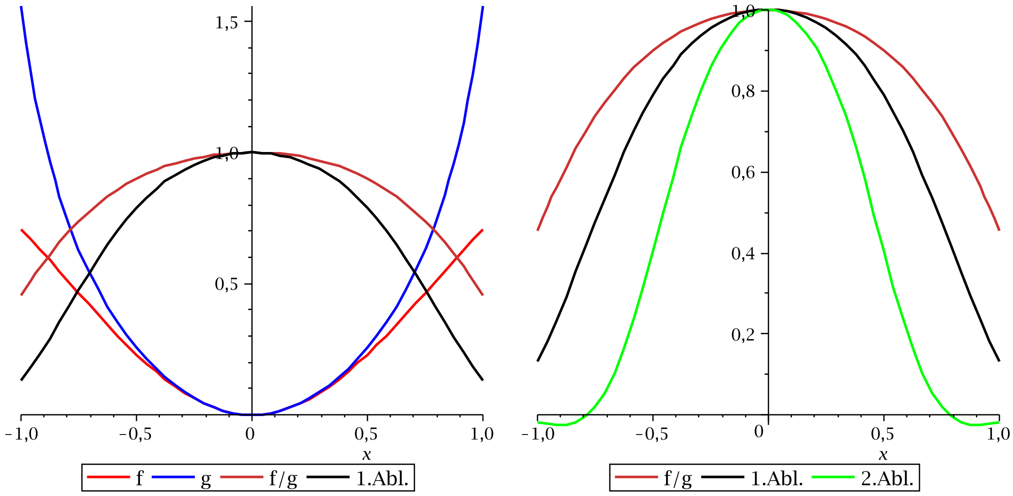


Abbildung 9.7.: Links: Die Funktionen $f$, $g$, $f/g$ und $f'/g'$ in der Umgebung der Stelle $a = 0$. Rechts: Die Quotienten $f/g$, $f'/g'$, $f''/g''$.

die Aussage

$$\lim_{x\to a}\frac{f(x)}{g(x)} = \lim_{x\to a}\frac{f'(x)}{g'(x)}$$

folgt.

### 9.5.2. Untersuchung der Funktion $\sin^2 x/\tan(x^2)$

Wir untersuchen die Funktion

$$h(x) = \frac{\sin^2 x}{\tan(x^2)}$$

an der Stelle $a = 0$. Der Funktionswert ist an dieser Stelle nicht definiert, da Zähler und Nenner verschwinden. Für die folgende Untersuchung betrachten wir Zähler und Nenner getrennt: $f(x) = \sin^2 x$ und $g(x) = \tan(x^2)$, und definieren in `MAPLE` entsprechend

```
f:=x->sin(x)^2; g:=x->tan(x^2);
```

dann gilt:

```
limit(f(x),x=0);limit(g(x),x=0);
```

$$0$$
$$0$$

Beide Funktionen $f$ und $g$ haben in $a = 0$ den Grenzwert 0. Abbildung 9.7 zeigt die zugehörigen Schaubilder der Funktionen und wurde mittels

```
plot([f,g,f(x)/g(x),D(f)(x)/D(g)(x)],x=-1..1,color=[red,blue,orange,black]);
```

erzeugt. Abbildung 9.7 legt die Vermutung nahe, dass $f/g$ und $f'/g'$ an der Stelle $a = 0$ den Wert 1 annehmen. Diese Vermutung werden wir weiter untersuchen. Algebraisch erhalten wir mit `MAPLE`:

```
D(f)(x)/D(g)(x);
```

$$\frac{\sin(x)\cos(x)}{\left(1 + (\tan(x^2))^2\right)x}$$

Auch dieser Bruch ist an der Stelle Null nicht definiert, da Zähler und Nenner wieder verschwinden. Somit erfüllt $f'/g'$ genauso wie bereits $f/g$ die Bedingung für den Satz von l'Hospital. Diesen können wir daher ein zweites Mal anwenden:
`plot([f(x)/g(x),D(f)(x)/D(g)(x),(D2)(f)(x)/(D2)(g)(x)],x=-1..1);`
Algebraisch erhalten wir den Quotienten der zweiten Ableitungen mit `MAPLE` wie folgt:
`simplify((D2)(f)(x)/(D2)(g)(x));`

$$\frac{\left(2\,(\cos(x))^2-1\right)\left(\cos\left(x^2\right)\right)^3}{4\,\sin\left(x^2\right)x^2+\cos\left(x^2\right)} \tag{9.4}$$

Wegen $\sin(0) = 0$ und $\cos(0) = 1$ folgt, dass Zähler und Nenner in Gleichung (9.4) jeweils den Wert 1 annehmen und der Quotient $f''/g''$ somit den Wert 1 besitzt. Wir haben folgendes Ergebnis erhalten:

$$\lim_{x\to 0}\frac{\sin^2(x)}{\tan(x^2)} = 1.$$

□

### 9.5.3. Direkte Berechnung der Grenzwerte

Die Grenzwerte können mit `MAPLE` natürlich auch direkt ausgerechnet werden:
```
Limit(f(x)/g(x),x=0) =  limit(f(x)/g(x),x=0);
```

$$\lim_{x\to 0}\frac{(\sin(x))^2}{\tan(x^2)} = 1$$

bzw.
```
Limit(D(f)(x)/D(g)(x),x=0) =limit(D(f)(x)/D(g)(x),x=0);
```

$$\lim_{x\to 0}\frac{\sin(x)\cos(x)}{\left(1+\left(\tan\left(x^2\right)\right)^2\right)x} = 1$$

und
```
Limit((D2)(f)(x)/(D2)(g)(x),x=0) = limit((D2)(f)(x)/(D2)(g)(x),x=0);
```

$$\lim_{x\to 0}\frac{2\,(\cos(x))^2-2\,(\sin(x))^2}{8\,\tan\left(x^2\right)\left(1+\left(\tan\left(x^2\right)\right)^2\right)x^2+2+2\,\left(\tan\left(x^2\right)\right)^2} = 1$$

### 9.5.4. Der Limit Tutor und die Regel von L'Hospital

Die Regel von l'Hospital kann im `LimitTutor` wie folgt aufgerufen werden:

```
restart: with(Student[Calculus1]):
LimitTutor((sin(x))^2/tan(x^2), x=0);
```

**Warnung:** Die im `MAPLE`-Tutor vorgeschlagenen Schritte sind nicht identisch mit den Schritten, die wir in Abschnitt 9.5.2 hergeleitet haben.

## 9.6. Eigenschaften differenzierbarer Funktionen, Kurvendiskussion

Wir führen die in Abschnitt 8.3 begonnene Kurvendiskussion fort. Schaubild, Symmetrie, Nullstellen und asymptotisches Verhalten hatten wir bereits untersucht. Nun betrachten wir noch lokale Extrema und Wendepunkte. Gegeben sei die Funktion $f$ aus Abschnitt 8.3

$$f(x) = \frac{x}{\sqrt{x^4+2}}$$

In `MAPLE` definieren wir:
`f:= x -> x/sqrt(x^4+2):`

### 9.6.1. Extremwerte

Zur Untersuchung der lokalen Extrema (und anschließend der Wendepunkte) bestimmen wir die ersten drei Ableitungen mit dem `D` Befehl.
`df1:=D(f)(x);`
ergibt

$$df1 := \frac{1}{\sqrt{x^4+2}} - 2\,\frac{x^4}{(x^4+2)^{3/2}}.$$

Dieser Ausdruck lässt sich noch vereinfachen, indem wir den `simplify` Befehl anwenden:
`df1:=simplify(D(f)(x));`

$$df1 := -\frac{x^4-2}{(x^4+2)^{3/2}}$$

Beachten Sie, dass wir einen Ausdruck (und keine Funktion) als Ergebnis erhalten haben, weil wir `df1:=D(f)(x);` und nicht `df1:=D(f);` berechnet haben. Der so erhaltene Ausdruck wird im Folgenden gleich Null gesetzt, um die Bedingungen für das Vorliegen von Extremstellen zu prüfen.
`s1:=solve( df1 = 0,x);`

$$s1 := 2^{(1/4)}, I * 2^{(1/4)}, -2^{(1/4)}, -I * 2^{(1/4)}$$

`evalf(%);`
ergibt:

$$1.189207115, 1.189207115 * I, -1.189207115, -1.189207115 * I,$$

so dass wir zwei reelle und zwei komplexe Lösungen erhalten. `MAPLE` rechnet standardmäßig mit komplexen Zahlen. Da wir normalerweise nur an den reellen Lösungen interessiert sind, können wir das Paket `RealDomain` laden:

```
use RealDomain in;
s1:=solve( df1 = 0,x);
end use;
```

Dann erhalten wir mit

$$s1 := 2^{(4^{(-1)})}, -2^{(4^{(-1)})}$$

bzw. nach Anwendung von `evalf` die zugehörigen Fließkommadarstellungen

$$1.189207115, -1.189207115.$$

Mit der zweiten Ableitung überprüfen wir, ob diese Werte auch Extrema darstellen. Die Ergebnisse müssen wir nicht von Hand in die zweite Ableitung einsetzen, dafür ist der `MAPLE`-Befehl `subs` hilfreich:

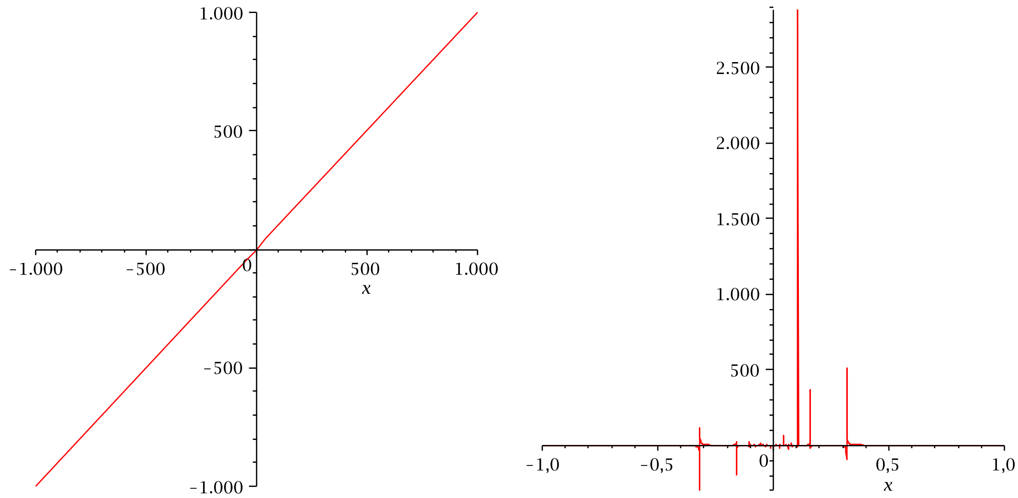


Abbildung 9.8.: $f(x) = \frac{1}{\sin \frac{1}{x}}$ im Bereich von $x = -1000$ bis 1000 (links) und $x = -1$ bis 1 (rechts).

```
df2:=simplify((D@@2)(f)(x));
evalf(subs(x=s1[1], df2));
```

ergibt:

$-.8408964155$

Da die zweite Ableitung negativ ist, liegt ein lokales Maximum vor. Erneut können wir den `subs`-Befehl anwenden, um den Funktionswert an der Stelle $x = \sqrt[4]{2} \approx 1.189207115$ zu berechnen:
`evalf(f(s1[1]));`

$0.5946035575$

Somit ist $E_1(1.189207115, 0.5946035575)$ ein lokales Maximum. Entsprechend können wir zeigen, dass $E_2(-1.189207115, -.5946035575)$ ein lokales Minimum ist.

### 9.6.2. Eine Bemerkung zur analytischen Berechnung der Extremstellen

Sie können sich an dieser Stelle fragen, warum wir die Extremstellen explizit mit `MAPLE` berechnen und nicht einfach ein Schaubild (`plot`) betrachten. Das Problem bei den Schaubildern ist, dass immer nur eine endliche Punktmenge gezeichnet werden kann. Der Definitionsbereich besteht aber i.d.R. aus unendlich vielen Punkten, so dass bei der grafischen Darstellung eine oder gar mehrere Extremstellen übersehen werden können. Betrachten Sie z.B. die Funktion

$$f(x) = \frac{1}{\sin \frac{1}{x}}$$

Das Schaubild für das Intervall von $x = -1000$ bis 1000 ist in Abbildung 9.8 (links) dargestellt. Rechts in Abbildung 9.8 finden Sie das Schaubild für den Bereich von $x = -1$ bis 1.

### 9.6.3. Wendepunkte

Wir überprüfen die Bedingungen $f''(x) = 0$ und $f'''(x) \neq 0$:
`ws:= evalf([solve(df2=0,x)]);`
Es gibt drei Kandidaten (eine dreifache Nullstelle): $0, 1.778279410, -1.778279410$. Nun wenden wir die dritte Ableitung an:

```
df3:=simplify((D@@3)(f)(x));
evalf(subs(x=ws[1], df3));
```

ergibt

$-0.$

, so dass an der Stelle $x = 0$ keine Wendestelle vorliegt. An den Stellen $x_2 = 1.778279410$ und $x_4 = -1.778279410$ nimmt die dritte Ableitung jedoch Werte ungleich Null an, so dass hier Wendepunkte vorliegen. Wir ermitteln noch die zugehörigen Funktionswerte mittels

```
evalf(subs(x=ws[2], f(x)));
```

$0.5133450480$

und

```
evalf(subs(x=ws[4], f(x)));
```

$-0.5133450480$

Beachten Sie, dass wir `f(x)` benutzt haben, da `f` im Gegensatz zu `df1`, `df2` und `df3` eine Funktion und kein Ausdruck ist. Die Wendepunkte lauten somit $W_1(1.778279410, 0.5133450480)$ und $W_3(-1.778279410, -0.5133450480)$.

## Teil IV.

# Lineare Algebra und Integralrechnung

# 10. Lineare Algebra und Analytische Geometrie

## 10.1. Übersicht

In diesem Kapitel wird die Darstellung von Vektoren und Matrizen in MAPLE eingeführt und einfache Berechnungen durchgeführt. Zur Addition von Vektoren und Matrizen verwendet MAPLE die bereits bekannten Operatoren $+$ und $-$. Gilt für die Multiplikation $a \cdot b = b \cdot a$, wobei $a$ und $b$ zwei Zahlen darstellen, dann ist die Multiplikation kommutativ. Der bereits bekannte Operator $*$ bezeichnet die kommutative Multiplikation in MAPLE. Da die Multiplikation für Vektoren und Matrizen nicht kommutativ ist, verwendet MAPLE den Punkt-Operator `.`.

Konen (2005) führt zunächst Matrizen und Determinanten ein und betrachtet dann Vektoren als Spezialfälle von Matrizen. Wir wählen den entgegengesetzten Weg, so dass wir Matrizen als Verallgemeinerungen von Vektoren betrachten.

## 10.2. Maple Pakete

*Wichtig:* Das `linalg` Paket wurde durch die Pakete `LinearAlgebra` und `VectorCalculus` ersetzt. Sie finden in älteren Lehrbüchern Beispiele aus dem `linalg` Paket. Wir beschäftigen uns im Folgenden nur mit den aktuellen Paketen.

Das `geom3d` Paket beinhaltet Befehle zur Analytischen Geometrie des Raumes (z.B. Darstellung und Berechnung von Punkten, Geraden und Ebenen im Raum)

## 10.3. Vektoren

Das Paket `LinearAlgebra` enthält die meisten Befehle für Vektoren. Wir setzten voraus, dass es geladen wurde:

```
with(LinearAlgebra);
```

$$[\&x, Add, Adjoint, BackwardSubstitute, BandMatrix, \ldots ZeroVector, Zip]$$

```
Vector(2);
```

erzeugt einen zweidimensionalen Vektor:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Um einen Spaltenvektor mit vordefinierten Werten zu erzeugen, gibt man

```
v:= Vector([1,2,3]);
```

ein.

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

`v:= Vector[row]([1,2,3]);` erzeugt einen Zeilenvektor (engl. row = Zeile).

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

Es gibt auch Kurzschreibweisen für die Erzeugung von Vektoren.

```
v:= <1,2,3>;
```

erzeugt einen Spaltenvektor,

```
v:= <1|2|3>;
```

einen Zeilenvektor. Um auf die einzelnen Komponenten eines Vektors zuzugreifen, geht man wie bei den Listen vor: Mit `v[i]` wird auf die $i$-te Komponente des Vektors $v$ zugegriffen:

```
v[3];
```

$$3$$

und ebenfalls, falls Sie an der letzten Position mit der Nummerierung beginnen:

```
v[-1];
```

$$3$$

### 10.3.1. Betrag eines Vektors

Mit dem Befehl `Norm` kann der Betrag eines Vektors bestimmt werden.

```
v:=<1,2,3>;
```

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

```
Norm(v,1);
Norm(v,2);
Norm(v);
```

ergibt dann

$$6$$
$$\sqrt{14}$$
$$3$$

`Norm(v,1)` ist die Betragssummennorm $||\cdot||_1$, also

$$v := (v_1, v_2, \ldots, v_n) \Rightarrow ||v||_1 := |v_1| + |v_2| + \ldots + |v_n|$$

`Norm(v,2)` ist die euklidische Norm $||\cdot||_2$, also

$$v := (v_1, v_2, \ldots, v_n) \Rightarrow ||v||_2 := \sqrt{v_1^2 + v_2^2 + \ldots + v_n^2}$$

`Norm(v)` ist die Maximumsnorm $||\cdot||_\infty$, also

$$v := (v_1, v_2, \ldots, v_n) \Rightarrow ||v||_\infty := \max\{|v_1|, |v_2| \ldots, |v_n|\}$$

### 10.3.2. Vektoraddition und -multiplikation mit Skalaren

Addition und Subtraktion werden für Vektoren komponentenweise durchgeführt. In `MAPLE` werden dazu `+` und `-` verwendet. Ein Vektor wird mit einem Skalar $s$ multipliziert,indem jede Komponente mit $s$ multipliziert wird. Diese Multiplikation ist in `MAPLE` mit dem Operator `*` möglich.

```
a := <1,2,3>: b := <4,5,z>:
a + 2*b;
```

$$\begin{bmatrix} 9 \\ 12 \\ 3 + 2\,z \end{bmatrix}$$

### 10.3.3. Skalarprodukt

Wir betrachten wieder zwei Vektoren `a := <1,2,3>: b := <4,5,z>:`. Der Befehl
```
DotProduct(a,b);
```
berechnet das Skalarprodukt der Vektoren $a$ und $b$.

$$14 + 3\,z$$

### 10.3.4. Vektorprodukt

Erneut seien zwei Vektoren `a := <1,2,3>: b := <4,5,z>:` gegeben. Der Befehl
```
CrossProduct(a,b);
```
berechnet das Vektorprodukt (Kreuzprodukt) der Vektoren $a$ und $b$. Beachten Sie, dass das Vektorprodukt nur für dreidimensionale Vektoren definiert ist.

$$\begin{bmatrix} 2\,z - 15 \\ 12 - z \\ -3 \end{bmatrix}$$

### 10.3.5. Winkel zwischen zwei Vektoren

Der Befehl
```
VectorAngle(a,b)
```
berechnet den Winkel $\alpha$, den zwei Vektoren $a$ und $b$ miteinander einschließen. Um den Winkel $\alpha$ in Gradmaß zu erhalten, sind weitere Umrechnungen erforderlich:

```
a  := <2,0,0>: b := <0,3,0>:
alpha := VectorAngle(a,b);
convert(alpha, degrees);
```

$$a := 1/2\,\pi$$
$$90\;degrees$$

Der Winkel kann auch für $n$-dimensionale Vektoren berechnet werden.

## 10.4. Matrizen

Matrizen sind in `MAPLE` zweidimensionale Felder `A[i,j]`, auf die mit einem Zeilen- und Spaltenindex zugegriffen werden kann. Matrizen werden zeilenweise definiert. Dabei wird jede Zeile als Liste angegeben.

```
with(LinearAlgebra):
A := Matrix( [ [1,2,3], [4,5,6], [7,8,9] ]);
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Für Matrizen gibt es, ähnlich wie für Vektoren, eine Kurzschreibweise:
```
B:= < <1|2|3>,<4|5|6>, <7|8|9> >;
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Beachten Sie, dass die Elemente auch spaltenweise definiert werden können. So ergibt

```
B:= < <1,2,3>|<4,5,6>| <7,8,9> >;
```

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

und

```
B:= < <1,2,3>,<4,5,6>,<7,8,9> >;
```

führt zu

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{bmatrix}$$

Der Befehl

```
Matrix(m,n)
```

erzeugt eine Nullmatrix mit $m$ Zeilen und $n$ Spalten.

```
Matrix(3,4);
```

führt zu

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

### Diagonalmatrix

Eine Diagonalmatrix wird mit dem Befehl `DiagonalMatrix` erzeugt:

```
DiagonalMatrix([1,2,3]);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

### Einheitsmatrix

Eine Einheitsmatrix wird mit dem Befehl `IdentityMatrix` erzeugt:

```
IdentityMatrix(4);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### 10.4.1. Rechenregeln für Matrizen

#### Addition und Subtraktion

Wir betrachten die Matrizen $A$ und $B$, die wie folgt definiert sind.

```
A := Matrix( [ [1,2,3], [4,5,6], [7,8,9] ]);
B := Matrix( [ [1,1,1], [2,2,2], [3,3,3] ]);
```

Dann folgt für die Addition:

```
A + B;
```

$$\begin{bmatrix} 2 & 3 & 4 \\ 6 & 7 & 8 \\ 10 & 11 & 12 \end{bmatrix}$$

und entsprechend für die Subtraktion

```
A - B;
```

$$\begin{bmatrix} 0 & 1 & 2 \\ 2 & 3 & 4 \\ 4 & 5 & 6 \end{bmatrix}$$

#### Multiplikation

Beachten Sie, dass für die nicht-kommutative Matrixmultiplikation das Zeichen `.` verwendet werden muss:

```
A := Matrix( [ [1,2,3], [4,5,6], [7,8,9] ]);
B := Matrix( [ [1,1,1], [2,2,2], [3,3,3] ]);
```

Dann folgt für die Multiplikation:

```
A . B;
```

$$\begin{bmatrix} 14 & 14 & 14 \\ 32 & 32 & 32 \\ 50 & 50 & 50 \end{bmatrix},$$

wohingegen

```
A * B;
```

den Fehler:

```
Error, (in rtable/Product) invalid arguments
```

erzeugt.

#### Transponieren

Zum Transponieren einer Matrix wird der Befehl `Transpose` verwendet:

```
A := Matrix( [ [1,2,3], [4,5,6], [7,8,9] ]);
B := Transpose(A);
```

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} B := \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

#### Invertieren

Zum Invertieren einer Matrix wird der Befehl `MatrixInverse` benutzt:

```
A := Matrix( [ [2,0,0], [0,2,0] , [0,0,2] ]);
B := MatrixInverse(A);
```

$$A := \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} B := \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/2 \end{bmatrix}$$

### 10.4.2. Determinanten

Mit dem Befehl `Determinant` wird die Determinante einer quadratischen Matrix berechnet.

```
A := Matrix( [ [2,0,0], [0,2,0] , [0,0,2] ]):
Det(A) = Determinant(A);
```

$$Det\left(\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}\right) = 8$$

`Det` ist eine träge Form und dient zur Darstellung der Determinante.

### 10.4.3. Rang einer Matrix

Den Rang einer Matrix, der die Anzahl der linear unabhängigen Zeilen (Spalten) einer $m \times n$ Matrix angibt, berechnet man in `MAPLE` mit dem Kommando `Rank`.

```
A := Matrix( [ [2,0,0], [0,2,0] , [0,0,2] ]):
Rang(A) = Rank(A);
```

$$Rang\left(\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}\right) = 3$$

Beachten Sie, dass `Rang` nicht eine träge Form darstellt, sondern ein einfacher `MAPLE` Ausdruck ist.

## 10.5. Arrays

### 10.5.1. Arrays in Maple

Beachten Sie, dass der Befehl `array` durch den Befehl `Array` ersetzt wurde. Ein Array kann wie folgt in `MAPLE` erzeugt werden:

```
A:=Array([[1,2,3],[4,5,6],[7,8,9]]);
```

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

### 10.5.2. Das Paket ArrayTools

Das `ArrayTools` Paket enthält wichtige Befehle zur Speicherung und Manipulation von Matrizen, Vektoren und Arrays. Mathematische Operationen für Matrizen und Vektoren werden durch die Pakete `LinearAlgebra` und `VectorCalculus` zur Verfügung gestellt.

Wir betrachten die folgenden Befehle aus dem Paket `ArrayTools`:

**Concatenate:** Zunächst erzeugen wir zwei Zeilenvektoren $v_1$ und $v_2$:

```
with(ArrayTools):
v1 := Vector[row]([a, b, c]);
v2 := Vector[row]([d, e, f]);
```

$$v1 := \begin{bmatrix} a & b & c \end{bmatrix}$$
$$v2 := \begin{bmatrix} d & e & f \end{bmatrix}$$

Diese werden dann mittels `Concatenate(1,v1,v2);` verknüpft:

$$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}$$

Ruft man `Concatenate` mit dem Parameter 2 auf, dann werden die Elemente in der Zeile verknüpft: `Concatenate(2,v1,v2)` ergibt somit:

$$\begin{bmatrix} a & b & c & d & e & f \end{bmatrix}$$

**Copy:** Der Befehl `Copy` ermöglicht das Kopieren eines Vektors oder einer Matrix in ein anderes Objekt gleichen Typs und gleicher Dimension. Wir erzeugen eine $(10 \times 2)$-Matrix mit zufälligen, ganzzahligen Einträgen aus dem Bereich von $-100$ bis 100:

```
A := LinearAlgebra:-RandomMatrix(10,2, generator = -100 .. 100,
 outputoptions = [datatype = integer]);
```

$$\begin{bmatrix} 28 & -41 \\ 18 & -53 \\ -23 & 97 \\ -83 & -69 \\ -82 & -22 \\ 17 & 56 \\ 65 & 92 \\ 59 & -92 \\ -4 & 21 \\ -80 & 30 \end{bmatrix}$$

Anschließend erzeugen wir eine $(10 \times 2)$-Matrix $B$, deren Einträge alle Null sind:

```
B := Matrix(10, 2, datatype = integer);
```

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Mittels `Copy` kopieren wir nun die Einträge von $A$ nach $B$:
`Copy(A, B); B;`
Die Matrix $B$ besitzt nun dieselben Einträge wie die Matrix $A$.

**Reshape:** Mit dem Befehl `Reshape` kann die Dimension von Matrizen verändert werden. So kann auch eine Matrix in einen Vektor (oder umgekehrt) verwandelt werden. Hier zwei Beispiele:

```
Reshape(Vector([1, 2, 3, 4, 5, 6]), 3, 2);
```

ergibt:

$$\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

```
Reshape(Vector([1, 2, 3, 4, 5, 6]), 1, 6);
```

ergibt:

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$$

## 10.6. Systeme von linearen Gleichungen

### 10.6.1. Lösungen linearen Gleichungssysteme

Bereits in Abschnitt 6.2 haben wir dargestellt, wie `MAPLE` lineare Gleichungssysteme mit dem `solve`-Befehl lösen kann.

### 10.6.2. Gauß-Verfahren

Ziel des Gauß-Verfahrens ist es, durch äquivalente Umformungen ein *lineares Gleichungssystem* (LGS) auf Treppenform bzw Zeilen-Stufen-Form zu bringen. Wir zeigen nun, wie ein Beispiel aus Konen (2005) mit `MAPLE` berechnet werden kann:

```
restart:with(LinearAlgebra):
A := <<1|1|1|2>,<2|3|1|-1>,<3|1|4|13>>;
```

$$A := \begin{bmatrix} 1 & 1 & 1 & 2 \\ 2 & 3 & 1 & -1 \\ 3 & 1 & 4 & 13 \end{bmatrix}$$

Der Befehl, um ein LGS mit dem Gauß-Verfahren in `MAPLE` auf Treppenform bzw. auf Zeilenstufenform zu bringen, lautet `GaussianElimination`.

```
GaussianElimination(A);
```

ergibt

$$\begin{bmatrix} 1 & 1 & 1 & 2 \\ 0 & 1 & -1 & -5 \\ 0 & 0 & -1 & -3 \end{bmatrix}$$

Mit dem Befehl ReducedRowEcholonForm wird die reduzierte Stufenform einer Matrix durch den Gauß-Jordan-Algorithmus berechnet.

```
ReducedRowEchelonForm(A);
```

### 10.6.3. Tutoren

Ein interaktives Tutorensystem zum Gauß-Verfahren erhalten Sie durch folgende Eingabe:

```
Student[LinearAlgebra][GaussianEliminationTutor]();
```

### 10.6.4. Der Befehl LinearSolve

Die Lösung eines linearen Gleichungssystems der Gestalt $A\vec{x} = \vec{b}$ wird mit dem Befehl `LinearSolve(A,b)` berechnet. Beachten Sie den freien Parameter in der Lösung!

```
A := <<1|1|1|2>,<2|3|1|-1>,<3|1|4|13>>;
 b:=<1,0,3>;
 x:=LinearSolve(A,b);
```

$$\texttt{A:=} \begin{bmatrix} 1 & 1 & 1 & 2 \\ 2 & 3 & 1 & -1 \\ 3 & 1 & 4 & 13 \end{bmatrix}$$

$$\texttt{b:=} \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix}$$

$$\texttt{x:=} \begin{bmatrix} -5 - \_t3_4 \\ 2 + 2\_t3_4 \\ 4 - 3\_t3_4 \\ \_t3_4 \end{bmatrix}$$

Die Probe:

```
A.x;
```

$$\begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix}$$

## 10.7. Punkte, Geraden und Ebenen mit geom3d

Hinweis: Wir verwenden im Folgenden das Paket `geom3d`. Dieses stellt Funktionen zur Konstruktion der folgenden geometrischen Objekte zur Verfügung:

- geom3d[point]
- geom3d[segment]
- geom3d[dsegment] (for directed line segment)
- geom3d[line]
- geom3d[triangle]
- geom3d[plane]
- geom3d[sphere]
- geom3d[polyhedra]

### 10.7.1. Punkte

Um einen Punkt in `MAPLE` darzustellen, wird der Befehl `point` benutzt.

```
restart: with(geom3d):
point(P1, [1,2,3]):
detail(P1);
```

ergibt:

```
name of the object P1,
form of the object point3d,
coordinates of the point [1,2,3]
```

Mittels
`draw(P1, axes=normal);`
kann der Punkt gezeichnet werden.

### 10.7.2. Geraden

Eine Gerade kann durch zwei Punkte festgelegt werden. Dies geschieht in `MAPLE` mit dem Befehl `line`.

```
point(P1, [1,2,3]):
point(P2, [-1,1,2]):
line(g1, [P1,P2]):
detail(g1);
```

`detail` gibt jetzt allerdings einige Warnmeldungen aus, u.a. weil eine Bezeichnung für den Parameter der Geradengleichung fehlt:

```
Warning, assume that the parameter in the parametric equations is _t
Warning, assuming that the names of the axes are _x, _y, and _z
name of the object g1, form of the object line3d,
equation of the line, _x = 1 - 2 _t, _y = 2 - _t, _z = 3 - _t
```

Wir wählen als Parameter $\lambda$ und rufen die Geradengleichung wie folgt auf:
`Equation(g1,lambda)`

$[1 - 2\lambda, 2 - \lambda, 3 - \lambda]$

Dies ist die Darstellung der Punkt-Richtungsform der Geraden

$$g_1 := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + \lambda \begin{pmatrix} -2 \\ -1 \\ -1 \end{pmatrix}$$

in `MAPLE`.

## 10.7.3. Ebenen

### Ebene durch drei Punkte

Eine Ebene kann durch drei Punkte festgelegt werden. Dies geschieht in `MAPLE` mit dem Befehl `plane`.

```
point(P1, [-1,1,1]):
point(P2, [1,1,0]):
point(P3, [1,-1,2]):
plane(E1, [P1,P2,P3]):
Equation(E1, [x,y,z]);
```

Ähnlich wie bei der Geraden geben wir bei der Ebene die Namen der Koordinatenachsen an. Wir erhalten folgende Ausgabe:

$6 - 2x - 4y - 4z = 0$

für die Ebenengleichung.

### Punkt-Richtungsform der Ebenengleichung

`MAPLE` bietet weitere Möglichkeiten an, Ebenen und Geradengleichungen anzugeben. Wir betrachten als Beispiel die Definition einer Ebene durch einen Punkt $P$ und zwei Richtungen $a_1$ und $a_2$, die sog. *Punkt-Richtungsform einer Ebene*

```
point(P,[-1,1,1]):
a1 := [1,1,0]:
a2 := [1,-1,2]:
line(g1,[P,a1]):
line(g2,[P,a2]):
plane(E2,[P,g1,g2]):
Equation(E2,[x,y,z]);
```

ergibt die Gleichung der Ebene $E_2$ in Koordinatendarstellung.

$6 + 2x - 2y - 2z = 0$

Mittels
`draw(E2,axes=boxed);`
erhalten wir die in Abbildung 10.1 gezeigte Darstellung.

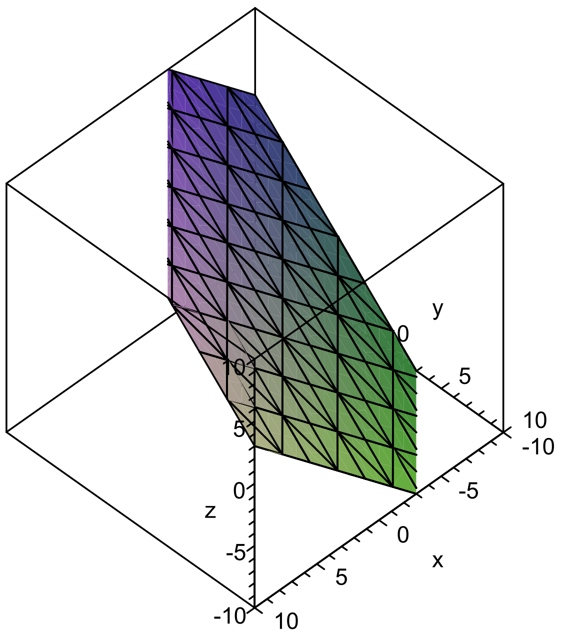


Abbildung 10.1.: Ebenendarstellung mit draw.

#### Normalenform

Eine Ebene kann auch durch einen Punkt und einen Normalenvektor festgelegt werden. Der Aufruf in `MAPLE` lautet dann z.B.:

```
with(geom3d):
with(LinearAlgebra):
point(B, [0,0,6]);
n := Vector([1,2,3]);
plane(p,[B,n],[x,y,z]);
```

### 10.7.4. Schnitte von Geraden und Ebenen

Zur Bestimmung des Schnittpunktes zweier Geraden bietet `MAPLE` den Befehl `intersection` an. Wir betrachten zwei Geraden $g_1$ und $g_2$:

```
point(P,[-1,1,1]):
a1 := [1,1,0]:
a2 := [1,-1,2]:
line(g1,[P,a1]):
line(g2,[P,a2]):
```

Mittels

```
intersection(S, g1, g2):
oordinates(S);
```

können wir den (bereits bekannten) Schnittpunkt $S$ berechnen.

$$[-1, 1, 1]$$

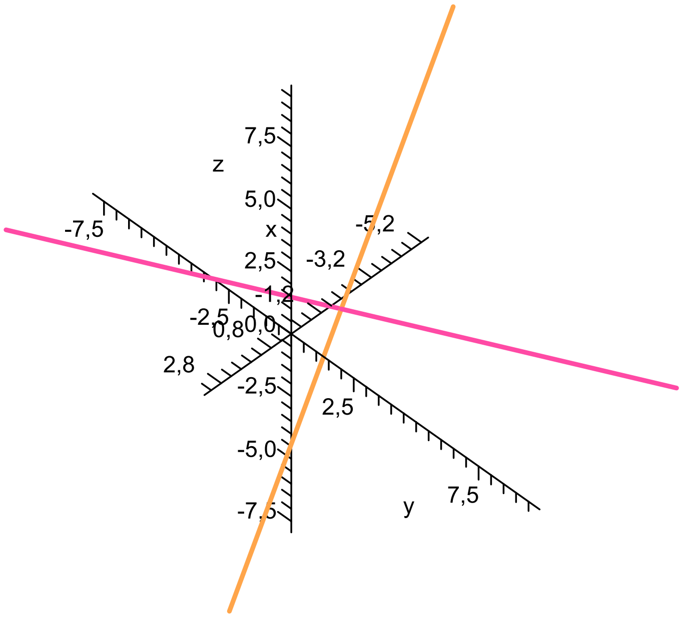


Abbildung 10.2.: Darstellung zweier Geraden mit draw.

## 10.7.5. Schnittwinkel

Wir berechnen den Schnittwinkel $\alpha$ der Geraden $g_1$ und $g_2$:

```
point(P,[-1,1,1]):
line(g1, [P,[-1,2,0]]):
line(g2, [P,[-1,0,2]]):
```

Mittels

```
alpha:= FindAngle(g1, g2):
evalf(alpha);
```

können wir den Schnittwinkel $\alpha$ berechnen.

$\alpha := \arccos(1/5)$
$1.369438406$

`draw` zeichnet die entsprechenden Objekte (s. Abbildung 10.2):

```
draw([g1,g2], axes=normal, thickness=2, labels=["x","y","z"]);
```

## 10.7.6. Abstände

Zur Berechnung des Abstands von Punkten, Geraden und Ebenen steht der Befehl `distance` zur Verfügung.

```
point(P1,[-1,1,1]):
point(P2,[-2,1,2]):
line(g1, [P1,[-1,2,0]]):
line(g2, [P2,[-1,0,2]]):
```

Mittels

```
d1 := distance(P1, P2);
```

$$d1 := \sqrt{(}2)$$

kann der Abstand der Punkte $P_1$ und $P_2$ berechnet werden. Entsprechend kann auch der Abstand zwischen einem Punkt und einer Geraden berechnet werden:

```
d2 := distance(P1, g2);
```

$$d2 := 1/5\,\sqrt{5}$$

## 10.8. Geraden und Ebenen ohne das Paket geom3d

### 10.8.1. Geraden

Eine Gerade läßt sich durch einen Ortsvektor, der vom Ursprung auf einen Punkt der Geraden zeigt und einen Richtungsvektor beschreiben. Wir betrachten die beiden Geraden $g_1$ und $g_2$:

```
restart:
 with(LinearAlgebra):
 a1:=<1,2,3>:
 r1:=<2,1,4>:
 g1:=lambda1->a1+lambda1*r1;
 a2:=<2,4,2>:
 r2:=<1,0,3>:
 g2:=lambda2->a2+lambda2*r2;
```

$$\texttt{g1:=}\lambda 1 \mapsto a1 + \lambda 1\ r1$$
$$\texttt{g2:=}\lambda 2 \mapsto a2 + \lambda 2\ r2$$

Für jeden Wert des Parameters $\lambda_1$ ($\lambda_2$ entsprechend) erhält man einen Punkt der Geraden, z.B.:

```
g1(3);
```

$$\begin{bmatrix} 7 \\ 5 \\ 15 \end{bmatrix}$$

#### Schnittpunkte

Um einen Schnittpunkt der beiden Geraden $g_1$ und $g_2$ zu berechnen, stellen wir mit dem Befehl `equate` ein lineares Gleichungssystem mit 3 Gleichungen auf. Dieses wird gelöst und anschließend werden den Parametern die gefundenen Lösungen zugewiesen.

```
LGS:=student[equate](g1(lambda1),g2(lambda2));
 lsg:=solve(LGS);
 assign(lsg):
```

$$\texttt{LGS:=}\{1 + 2\,\lambda 1 = 2 + \lambda 2, 2 + \lambda 1 = 4, 3 + 4\,\lambda 1 = 2 + 3\,\lambda 2\}$$
$$\texttt{lsg:=}\{\lambda 1 = 2, \lambda 2 = 3\}$$

Als Schnittpunkt erhalten wir:

```
g1(lambda1)=g2(lambda2);
lambda1:='lambda1': lambda2:='lambda2':
```

$$\begin{bmatrix} 5 \\ 4 \\ 11 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \\ 11 \end{bmatrix}$$

Hätten die beiden Geraden keinen gemeinsamen Schnittpunkt, hätten wir mit Maple keine Lösungen für $\lambda_1$ und $\lambda_2$ erhalten. Wären die beiden Geraden identisch, hätten wir unendlich viele Lösungen, d.h. einen freien Parameter erhalten.

### Winkelberechnung

Der Winkel zwischen den beiden sich schneidenden Geraden entspricht dem Winkel zwischen den Richtungsvektoren der beiden Geraden. Wir berechnen den Winkel mit dem Befehl `VectorAngle` (siehe Abschnitt 10.3.5) zunächst in Bogenmaß und konvertieren ihn anschließend in Gradmaß:

```
alpha:=VectorAngle(r1,r2);
evalf(convert(alpha,degrees));
```

$$\arccos\left(1/15\,\sqrt{21}\sqrt{10}\right)$$
$$14.96321736\ \mathit{degrees}$$

## 10.8.2. Ebenen

Eine Ebene läßt sich durch einen Ortsvektor, der vom Ursprung auf einen Punkt der Ebene zeigt und zwei Richtungsvektoren, die in der Ebene liegen, beschreiben. Dabei dürfen die Richtungsvektoren kein Vielfaches voneinander sein. Die Ebenengleichung für eine Ebene $E$ in der Parameterform lautet dann z.B.:

```
a:=<-1,3,2>:
r:=<4,1,1>:
s:=<-2,-1,1>:
E:=(lambda,mu)->a+lambda*r+mu*s;
```

$$\mathtt{E:=}(\lambda,\mu) \mapsto a + \lambda\, r + \mu\, s$$

### Schnittpunkte

Um einen Schnittpunkt der Ebene $E$ mit der Geraden $g_1$ zu berechnen, stellen wir wieder mit dem Befehl equate ein lineares Gleichungssystem mit 3 Gleichungen auf, lösen dieses und weisen den Parametern die gefundene Lösung zu.

```
LGS:=student[equate](E(lambda,mu),g1(lambda1));
lsg:=solve(LGS);
assign(lsg):
```

$$\mathtt{LGS:=}\{-1+4\,\lambda-2\,\mu=\lambda 1, 2+\lambda+\mu=\lambda 1, 3+\lambda-\mu=\lambda 1\}$$
$$\mathtt{lsg:=}\{\mu=\tfrac{11}{5}, \lambda=2, \lambda 1=4/5\}$$

Als Schnittpunkt erhalten wir:

```
E(lambda,mu)=g1(lambda1);
lambda:='lambda': mu:='mu': lambda1:='lambda1':
```

$$\begin{bmatrix} \frac{13}{5} \\ \frac{14}{5} \\ \frac{31}{5} \end{bmatrix} = \begin{bmatrix} \frac{13}{5} \\ \frac{14}{5} \\ \frac{31}{5} \end{bmatrix}$$

Würde die Gerade parallel zur Ebene liegen, hätten wir mit Maple keine Lösung erhalten. Würde die Gerade in der Ebene liegen, hätten wir unendlich viele Lösungen, d.h. einen freien Parameter erhalten. Entsprechend können auch Schnitte zwischen 2 Ebenen berechnet werden.

### Abstandsberechnung

Der (minimale) Abstand d zwischen einem Punkt $Q$ und einer Ebene $E$ kann auch mit der Formel $d = \frac{|\vec{n} \cdot (\vec{r}_Q - \vec{a})|}{|\vec{n}|}$ berechnet werden. Dabei ist $\vec{r}_Q$ der Ortsvektor vom Punkt $Q$, $\vec{n}$ der Normalenvektor der Ebene $E$ und $\vec{a}$ ein Ortsvektor der Ebene $E$. Mit den Befehlen `CrossProduct` für das Kreuzprodukt, `DotProduct` für das Skalarprodukt und `Norm( ,2)` für die euklidische Norm erhält man für den Abstand vom Punkt $Q = [4, 5, 6]$ zur o.g. Ebene $E$:

```
rQ:=<4,5,6>;
n:=CrossProduct(r,s);
d:=abs(DotProduct(n,rQ-a))/Norm(n,2);
```

$$\mathtt{rQ:=}\begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

$$\mathtt{n:=}\begin{bmatrix} 2 \\ -6 \\ -2 \end{bmatrix}$$

$$\mathtt{d:=}\frac{5}{11}\sqrt{1}$$

# 11. Integralrechnung

## 11.1. Übersicht

`MAPLE` kann bestimmte, unbestimmte und uneigentliche Integrale mit dem Befehl `int` berechnen.

## 11.2. Das bestimmte Integral

Zur Berechnung des bestimmten Integrals

$$\int_a^b f(x)\,dx, \tag{11.1}$$

z.B. für die Funktion $f(x) = x^2 + \ln(x)$ in den Grenzen von $a = 1$ bis $b = 2$ geht man folgendermaßen vor:

```
f(x) := x^2 + ln(x):
Int(f(x), x=1..2) = int(f(x), x=1..2);
```

$$\int_1^2 x^2 + \ln(x)\,dx = 4/3 + 2\,\ln(2)$$

`Int` bezeichnet wieder die träge Form des Integrals. Mittels `value` kann eine träge Form später ausgewertet werden. Es gilt also:

```
Int(f(x), x=1..2) = value( Int(f(x), x=1..2));
```

$$\int_1^2 x^2 + \ln(x)\,dx = 4/3 + 2\,\ln(2)$$

## 11.3. Numerische Integration

Zur numerischen (näherungsweisen) Bestimmung des Integral geht man in `MAPLE` folgendermaßen vor. Zunächst verwendet man die träge Form (`Int`), die mittels `evalf` ausgewertet wird.

```
I1 := Int (f(x), x=1..2);
evalf(I1);
```

liefert

$$I1 := \int_1^2 x^2 + \ln(x)\,dx$$
$$2.719627694$$

## 11.4. Volumen von Rotationskörpern

Rotiert der Funktionsgraph der Funktion $f(x)$ um die $x$-Achse, so entsteht ein Körper mit dem Volumen

$$V_x = \pi \int_a^b f^2(x)\,dx. \tag{11.2}$$

Wir betrachten die Funktion $f(x) = x^2$ im Bereich von $a = 0$ bis $b = 1$. Der bei Rotation um die $x$-Achse entstehende Körper besitzt das Volumen

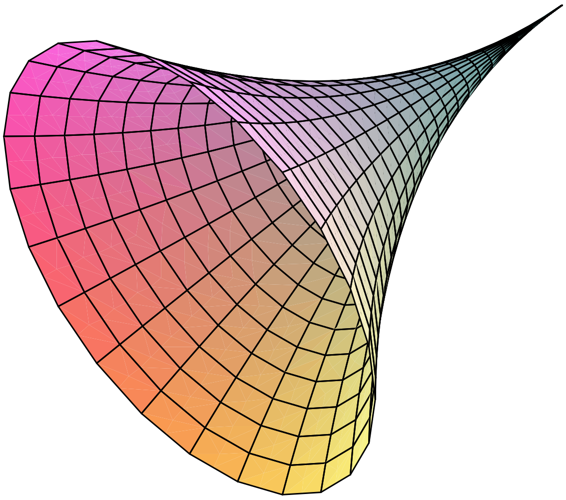

Abbildung 11.1.: Rotationskörper.

```
f(x) := x^2:
Vx := Pi * int(f(x)^2, x=0..1);
```

$$Vx := 1/5\,\pi$$

Zur Darstellung des Rotationskörpers kann der `plot3d`-Befehl benutzt werden (s. Abbildung 11.1):

```
f:= x -> x^2:
plot3d( [x, f(x) * cos(t), f(x) *sin(t)], x=0..1, t=0..2*Pi);
```

# Literaturverzeichnis


Bartz-Beielstein, T. (2009). Skript Mathematik 1, SS 2009. FH Köln.

Konen, W. (2005). Skript Mathematik 1, WS 2005/06. FH Köln.

Walz, A. (2002). *Maple 7 – Rechnen und Programmieren.* Oldenbourg, München, 2. edition.