Datenanalyse in der Physik

Vorlesung 1

Prof. Dr. J. Mnich

DESY und Universität Hamburg

Vorlesung 1

Heutige Vorlesung:

Computeralgebra-Programm Maple

Programme der Computeralgebra wie Maple oder Mathematica können eingesetzt werden für:

- numerische Rechnungen (erweiterter „Taschenrechner“)
- algebraische Rechnungen
- Lösung von Gleichungen und Differentialgleichungen
- graphische Darstellung der Lösungsfunktionen
- Neu: Editieren mathematischer Texte

Maple bietet eine sehr grosse Zahl an Möglichkeiten

Im Rahmen dieser Vorlesung und Übung kann nur eine Einführung gegeben werden

Achtung:
Auf den Rechnern sind neuere Maple Versionen installiert, die Beispiele und Screenshots der Vorlesung stammen aus Version 7.
Funktionalität ist aber gleich

Dokumentation

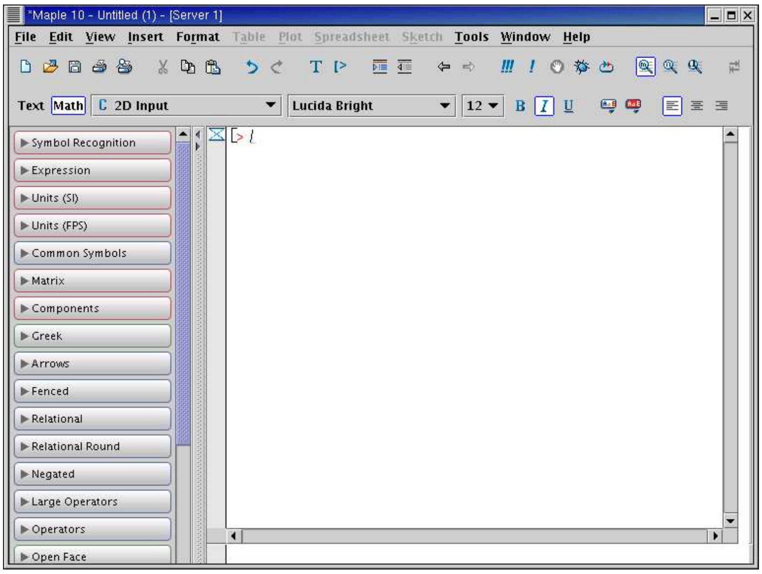
- Maple-Einführungkurse im Internet:
  - A) `Einführungskurs in Maple` oder
  - B) `Einführungskurs in Maple`

Maple auf den DESY Rechnern

- Aufruf von Maple mit graphischer Benutzeroberfläche durch den Befehl
  `xmaple`

  Geeignet zum interaktiven Arbeiten

Maple Benutzeroberfläche (Version 10):



Maple Arbeitsblatt und Hilfe

Arbeitsblatt

- Im Arbeitsblatt werden Maple-Kommandos eingegeben und Ergebnisse angezeigt
- Arbeitsblätter können in Dateien gespeichert werden und in einer späteren Maple-Sitzung wieder ausgeführt oder geändert werden
  Dateiendung ist `.mws` für Maple Worksheet

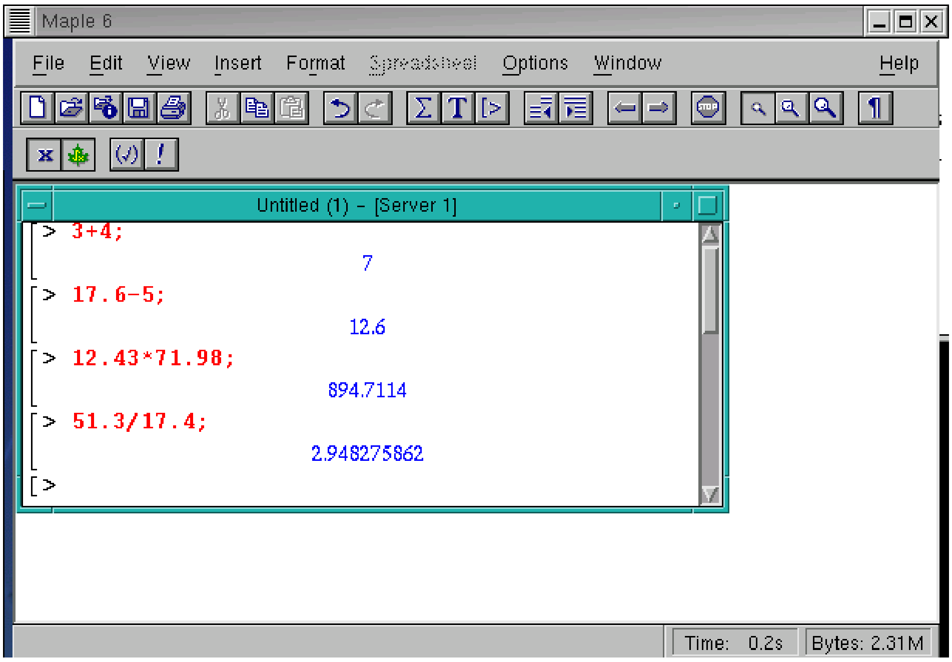
Online-Hilfe:
Maple bietet eine sehr umfangreiche Online-Hilfe

- `Help`-Schalter in der Menuleiste
- oder `?` im Arbeitsblatt
  Benutzung `?`*Befehlsname*, z.B.: `?plot`
- Einstieg in Maple durch `?intro`

Maple als Taschenrechner

In dem Maple-Arbeitsblatt können ganz normale arithmetische Rechnungen wie auf einem Taschenrechner eingegeben werden:

- Eingaben werden mit einem Semikolon `;` abgeschlossen (wie in C)
- oder mit einem Doppelpunkt `:` $\Longrightarrow$ keine Ergebnisausgabe



Konstanten und Funktionen

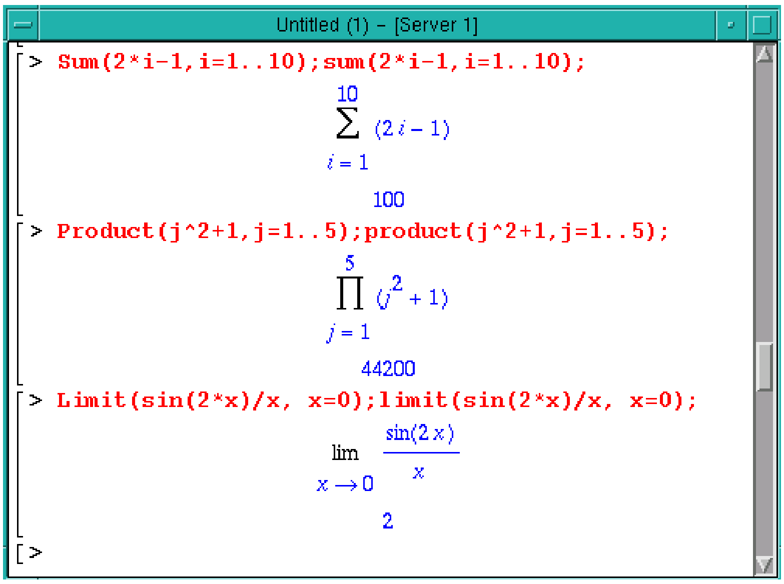
- Maple kennt alle mathematischen Konstanten und Funktionen, z.B.

Symbol	Wert	Eingabe
π	≈ 3.141592654	`Pi`
e	≈ 2.718281828	`exp(1)`
γ	≈ 0.577215665	`gamma`
i	$\sqrt{-1}$	`I`
∞		`infinity`

- Maple kennt alle elementaren Funktionen
  - Winkelfunktionen und deren Umkehrfunktionen, z.B. `sin` und `arcsin`
  - Hyperbolische (Umkehr-)funktionen, z.B. `sinh` und `arcsinh`
  - Potenzieren: `x^y` zur Berechnung von x^y
  - Exponential- und Logarithmusfunktionen: `exp(x)`, `log(x)` für $\ln x$, `log[b](x)` für $\log_b x$
  - Wurzelfuktion `sqrt(x)`
- und alle höheren mathematischen Funktionen (Legendre-Polynome, Bessel-Funktionen, . . .)

Einige spezielle Funktionen

Maple stellt u.a. Funktionen zur Berechnung von Summen, Produkten und Grenzwerten zur Verfügung



Man beachte:

- Operatoren, die mit einem Großbuchstaben beginnen, führen keine Rechnung durch (nur symbolisch)
- Bereiche von Zahlenwerten werden durch zwei Punkte `..` angegeben, z.B. `i=1..10`

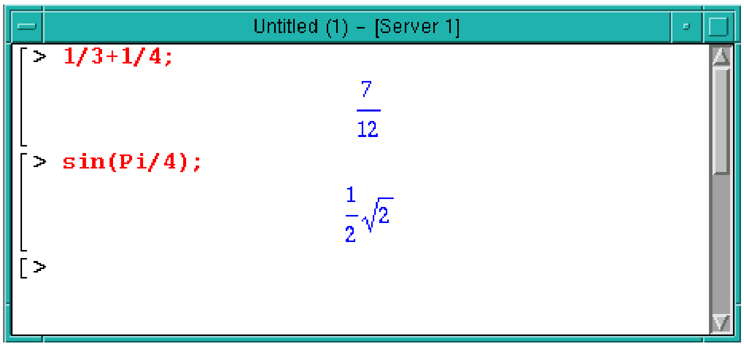
Numerische Berechnungen mit Maple

Was ist der Unterschied zwischen Maple und einem Taschenrechner?

- Maple ist ein Algebra-Programm

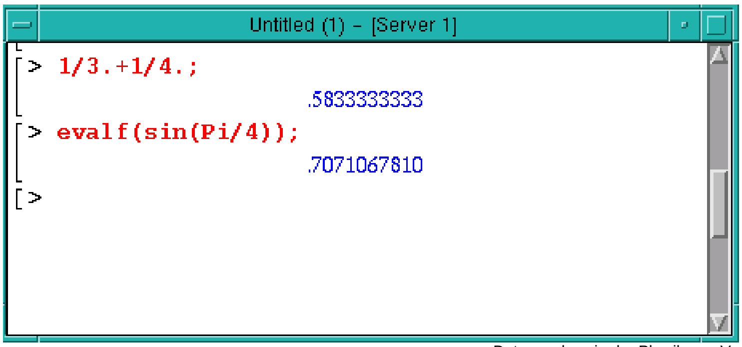
Es macht von sich aus keine Näherung und führt nur mathematisch exakte Vereinfachungen aus

Beispiel:



Wenn man numerisches Resultat haben will:

- entweder Eingabe von Fließkommazahl
- oder Benutzung des Maple Kommandos `evalf` (Evaluate floating point)



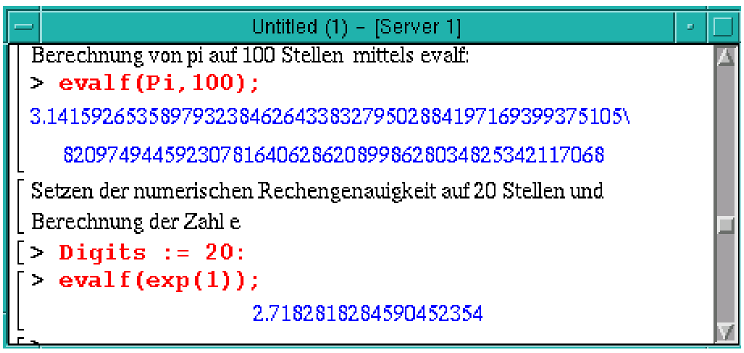
Universität Hamburg

Numerische Berechnungen mit Maple

- Maple kann numerische Berechnungen mit „beliebig“ vielen Stellen ausführen

Zahl der Stellen für numerische Umwandlungen kann spezifiziert werden

- durch 2. Argument von `evalf` nur für diese Rechnung
- setzen der Maple Umgebungsvariablen `Digits` für alle folgenden Rechnungen



Standard Einstellung (Default) ist 10 Stellen

Eine Zeilenfortsetzung (Ein- und Ausgabe) wird durch einen Backslash \ gekennzeichnet

Universität Hamburg

Numerische Berechnungen mit Maple

- Rechnungen mit großen, ganzen Zahlen werden nicht in Fließkommazahlen umgewandelt

Spezielle Maple-Funktionen für ganze Zahlen:

- `ifactor` zerlegt Zahl im Primfaktoren
- `isprime` testet, ob Zahl prim ist

Untitled (1) – [Server 1]

```
> 12^49;
```

$$75836984583351248111063210627854719374392938360471552$$

```
> ifactor(12345678901234567890123456789);
```

$$(3)^2\,(101)\,(3803)\,(3607)\,(27961)\,(3541)$$

```
> isprime(27961);
```

$$true$$

```
>
```


Variablen

In Maple ist der Begriff der Variablen erweitert:

Es können Namen für bestimmte Objekte vergeben werden

Zahlen, Terme, Geichungen, Listen, Sets, Vektoren, Matrizen, . . .

Zuweisung erfolgt durch den Operator `:=`

Beispiele:

Untitled (1) – [Server 1]

```
> a := Pi/3;
```

$$a := \frac{1}{3}\pi$$

```
> a;
```

$$\frac{1}{3}\pi$$

```
> term := cos(a) - sin(a);
```

$$term := \frac{1}{2} - \frac{1}{2}\sqrt{3}$$

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

$$a := a$$

```
> a;
```

$$a$$

- Durch *Variable* `:=` *'Variable'* wird Zuweisung rückgängig gemacht
- Der Befehl `restart` macht alle Variablenzuweisungen rückgängig

Einlesen von Daten

In Maple können Daten aus Textdateien eingelesen werden

Untitled (1) - [Server 1]

Öffne Datenfile aus erster Übung und lese in Liste mdata ein
File enthält 3 Spalten und 25 Zeilen

```
> fd1:=fopen("/home/mnich/dvphysik_ss02/vorlesung/result.dat",READ);
```

Zahl der Spalten (3) muss angeben werden

```
mdata:= readdata(fd,3):
> fclose("/home/mnich/dvphysik_ss02/vorlesung/result.dat");
```

$$fd1 := 0$$

Variable mdata wird als Liste definiert

```
whattype(mdata);
```

list

```
> A := Matrix(mdata);
```

$$A := \begin{bmatrix} \text{25 x 3 Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

```
> A[1,1],A[1,2],A[1,3];
```

$$88.231, 4.455, .118$$

Daten werden zunächst in Liste eingelesen, die dann in eine Matrix kopiert werden (→ Lineare Algebra)

Sequenzen, Listen, Variablentypen etc.

Untitled (1) - [Server 1]

Erzeuge Sequenz der ersten 6 Kubikzahlen

```
seq(i^3,i=1..6);
```

$$1, 8, 27, 64, 125, 216$$

Fülle Sequenz in eine Variable vom Typ list mit % – Operator
Liste wird durch eckige Klammern umschlossen

```
> Liste:=[%];
```

$$Liste := [1, 8, 27, 64, 125, 216]$$

Stelle Variablentyp fest

```
> whattype(Liste);
```

list

Überprüfe, ob es eine Liste ist

```
> type(Liste,list);
```

true

```
>
```

- Mit dem Befehl `seq` können Zahlensequenzen erzeugt werden
- Mit dem `%`-Operator kann das letzte Rechenergebnis zurück geholt werden, `%%` holt das vorletzte etc.
- Der Variablentyp `list` ist eine Liste von Zahlen (wie hier) oder anderen Maple-Objekten

  Listen werden von eckigen Klammern umschlossen `[...]`
- Der Typ einer Variablen kann mit den Befehlen `type` und `whattype` festgestellt werden

Eigene Funktionen

```
Untitled (2) - [Server 1]
> restart:
Definition einer eindimensionalen Funktion
> f := x -> 3*exp(sin(x)/2);
```

$$f := x \to 3\,e^{\left(\frac{1}{2}\sin(x)\right)}$$

```
> f(0.7);
```

$$4.140104928$$

```
Definition einer zweidimensionalen Funktion
> g := (u,v) -> 3*(u^2-v^2);
```

$$g := (u, v) \to 3\,u^2 - 3\,v^2$$

```
> g(sqrt(3),exp(1));
```

$$9 - 3\,(e)^2$$

```
Definition einerFunktion mit unapply
> volumen := 4/3*Pi*r^3;
```

$$volumen := \frac{4}{3}\pi\,r^3$$

```
> v := unapply(volumen,r);
```

$$v := r \to \frac{4}{3}\pi\,r^3$$

In Maple können eigene Funktionen definiert werden und wie die Maple Bibliotheksfunktionen verwendet werden

- Zuweisung der Funktionsvorschrift erfolgt durch den Operator `->`
- oder durch den Befehl `unapply`

  Erlaubt die Definition einer Funktion über Ergebnis einer algebraischen Berechnung

DESY Universität Hamburg

Algebra

```
Untitled (1) - [Server 1]
> a:=3*(x^2-2)^2;
```

$$a := 3\,(x^2 - 2)^2$$

```
> b:=subs(x=3*u+3,a);
```

$$b := 3\,((3\,u + 3)^2 - 2)^2$$

```
> expand(b);
```

$$243\,u^4 + 972\,u^3 + 1350\,u^2 + 756\,u + 147$$

```
> factor(b);
```

$$3\,(9\,u^2 + 18\,u + 7)^2$$

```
> normal( (x^2-y^2)/(x-y)^3);
```

$$\frac{x + y}{(x - y)^2}$$

```
> numer(5);denom(%%);
```

$$5$$

$$(x - y)^2$$

```
>
```

In Maple gibt es viele Befehle zum symbolischen Rechnen

- `subs` (substitute) ersetzt Variablen durch Werte oder Terme
- `expand` Ausmultiplizieren
- `factor` faktorisiert algebraische Ausdrücke
- `numer,denom` bezeichnen Zähler und Nenner eines Bruches
- `convert(Ausdruck, Typ)` versucht Ausdruck in bestimmte Form zu bringen
- `combine` fasst Terme zusammen
- `simplify` versucht Term zu vereinfachen
- `assume` schränkt Wertebereich einer Variablen ein
  normalerweise können Variablen Werte aus gesamten komplexen Zahlenbereich annehmen

DESY UHH Universität Hamburg

Pakete

Nach Aufruf von Maple steht nur eine Grundmenge an Kommandos zur Verfügung (sogenannte „built-in“-Befehle)

Sehr viel mehr weitere Kommandos sind in der Maple-Bibliothek definiert

Befehle zu einem Themenkreis sind in einem Paket („Package“) zusammengefasst

- Einbinden eines Paketes durch `with(`*package name*`)` z.B. `with(plots)`
- Liste der verfügbaren Pakete erhält man durch `?index,packages`
- Für uns wichtige Pakete sind:

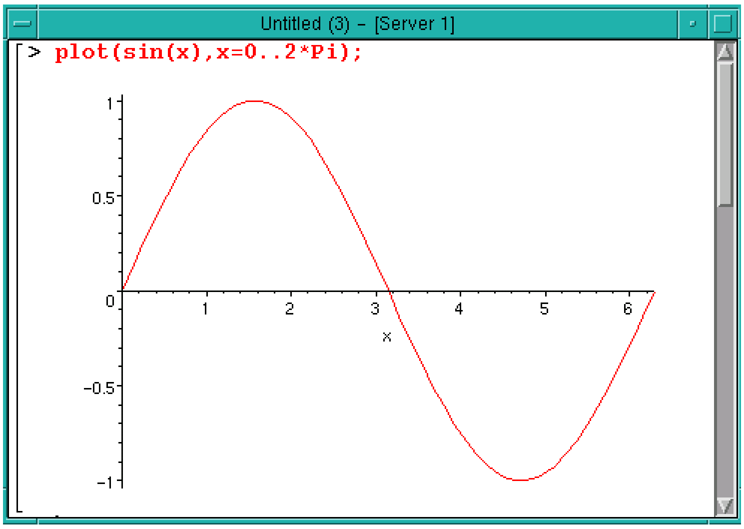
`plots`	Graphik
`LinearAlgebra`	Lineare Algebra
`DEtools`	Differentialgleichungen
`combinat`	Kombinatorik
`stats`	Statistik

Graphik

Maple bietet umfangreiche Möglichkeiten, Ergebnisse (Funktionen, Wertepaare) graphisch darzustellen

- der wichtigste Befehl ist `plot`
- der vollständige Befehlssatz ist im Paket `plots` definiert

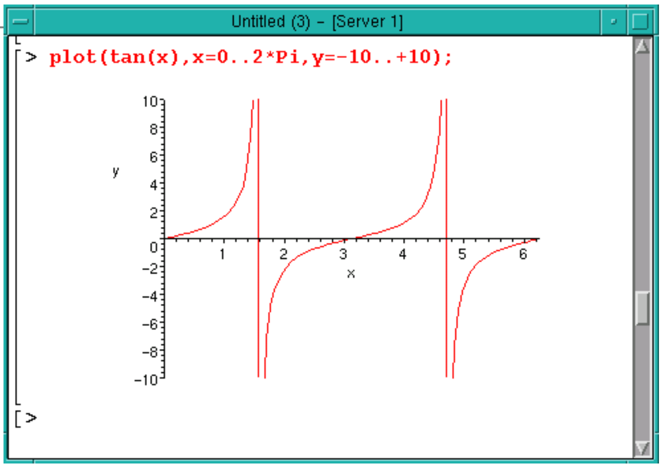
Tipp: Die Hilfe `?plot` enthält die komplette Beschreibung und sehr viele Beispiele



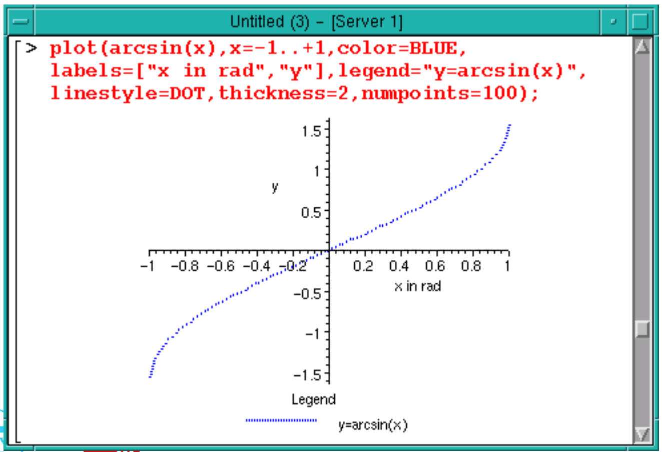
Plotten der Funktion $\sin x$ im Bereich $0 \leq x \leq 2\pi$

Die y-Achse wird automatisch skaliert

Graphik-Optionen



Wertebereich der y-Achse kann als Option angegeben werden



Alle Optionen sind in

`?plot,options`

aufgeführt und erklärt

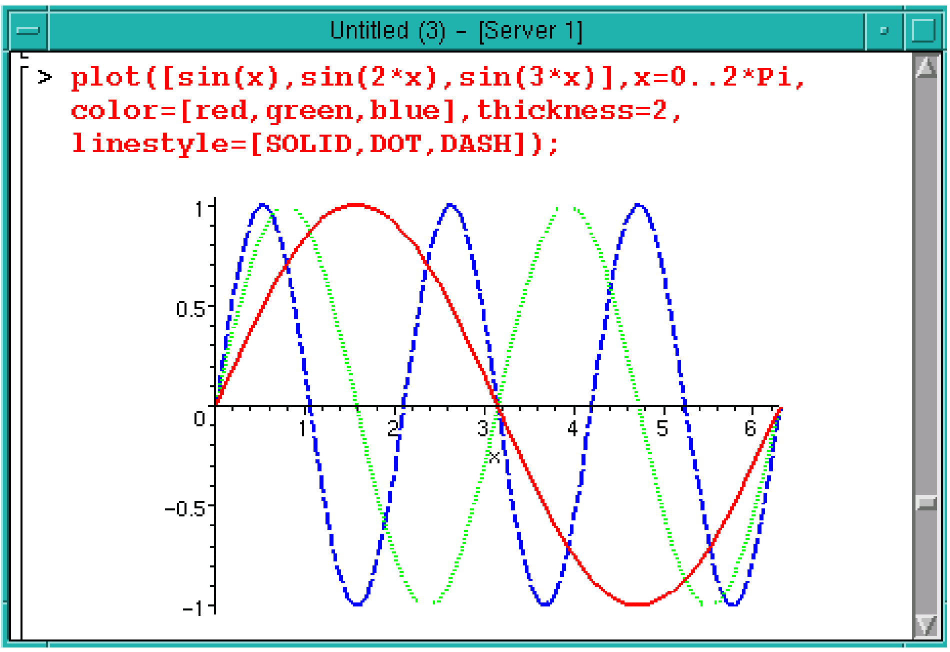
Im Beispiel links sind einige Optionen benutzt worden

Plotten mehrerer Funktionen

Darstellen mehrerer Funktionen in einer Graphik

- Angeben der Funktionen in einer Liste
- Optionen können für jede Funktion einzeln in Liste oder für alle gemeinsam angegeben werden

Beispiel:



Plotten von Wertepaaren

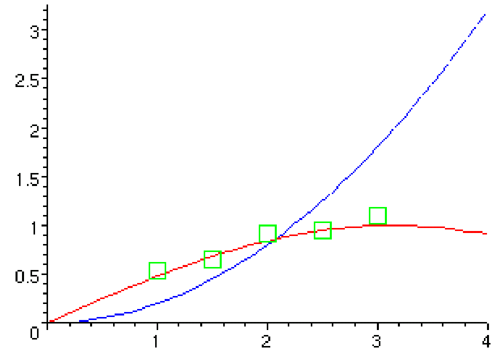
Untitled (1) - [Server 1]

```
> f(x) := x -> sin(x/2); g(x) := x -> x^2/5;
>
```

$$f(x) := x \rightarrow \sin\left(\frac{1}{2}x\right)$$

$$g(x) := x \rightarrow \frac{1}{5}x^2$$

```
> plot([f(x),g(x),punkte], 0..4,
  color=[red,blue,green],
  style=[line,line,point],
  linestyle=[SOLID,DASH,SOLID],
  symbol=box,symbolsize=20);
```



Es können auch Wertepaare, die in einer Liste enthalten sind, gezeichnet werden

und mit Funktionen verglichen werden

Dreidimensionale Graphiken

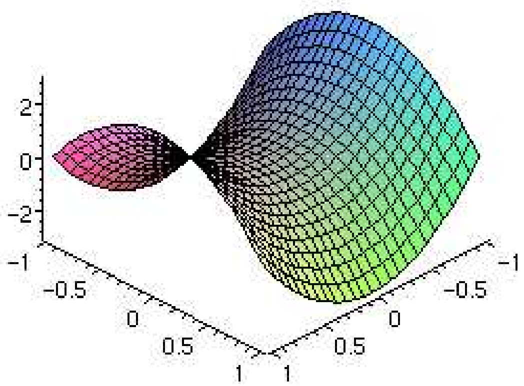
Die dreidimensionale Darstellung einer zweidimensionalen Funktion durch den Maple-Befehl `plot3d`

Untitled (1) - [Server 1]

```
> g := (u,v) -> 3*(u^2-v^2);
```

$$g := (u, v) \rightarrow 3u^2 - 3v^2$$

```
> plot3d(g,-1..+1,-1..+1,axes=FRAME);
```




```
>
>
```

Konturen

Als Beispiel für Befehle aus dem `plots`-Paket stellen wir hier Befehle zum zeichnen von Konturen (Höhenlinien) vor:

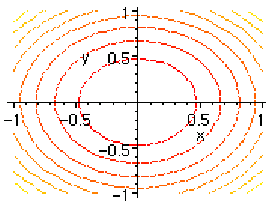
Zweidimensional

Untitled (1) - [Server 1]

```
> with(plots):
Warning, the name changecoords has been redefined
> V := (x,y) -> (x^2+y^2);
```

$$V := (x, y) \to x^2 + y^2$$

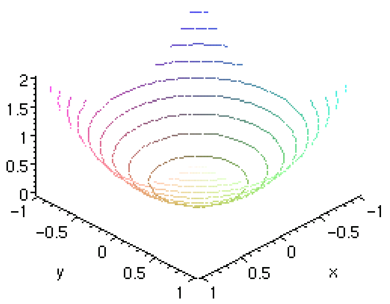
```
> contourplot(V(x,y),x=-1..+1,y=-1..+1,
  axes=NORMAL);
```



Dreidimensional

Untitled (1) - [Server 1]

```
> contourplot3d(V(x,y),x=-1..+1,y=-1..+1,
  axes=FRAME);
```




```
>
```


Darstellung dreidimensionaler Kurven

Mit dem Befehl `spacecurve` können dreidimensionale Linien, z.B. Bahnkurven, dargestellt werden:

Untitled (1) - [Server 1]

```
> restart:
> with(plots):
Warning, the name changecoords has been redefined
```

Parameterdarstellung einer Spirale

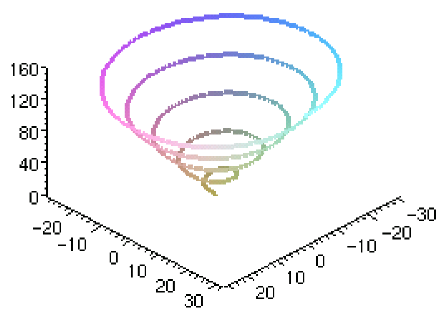
```
> x := t->t*sin(t); y := t->t*cos(t); z := t->5*t;
```

$$x := t \to t \sin(t)$$

$$y := t \to t \cos(t)$$

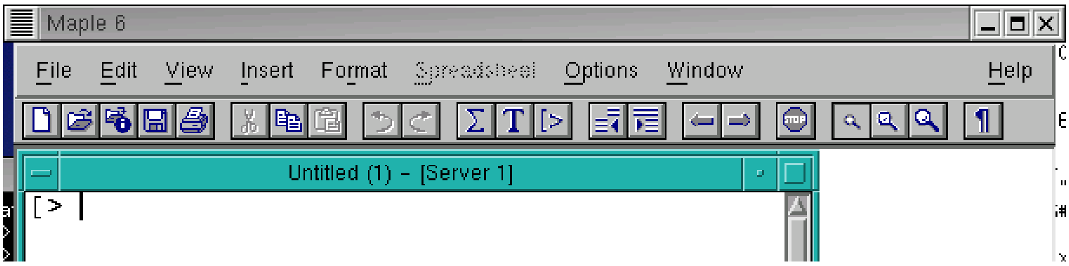
$$z := t \to 5\,t$$

```
> spacecurve([x(t),y(t),z(t)],t=0..10*Pi,
  numpoints=1000,thickness=3,axes=FRAME);
```



Arbeitsblätter (Worksheets) I

- Arbeitsblätter können erzeugt, geöffnet und abgespeichert werden
  über Menuleiste `File` → `...` oder Icons in der 2. Zeile:



  siehe `?helpmenu` und `?helptoolbar`

- Ändern, kopieren, einfügen von Kommandos & Text
  Menuleiste oder Control-Kommandos ähnlich zu Word (`C-x`, `C-v`, `C-v`, ...)
  Linux X-Windows Mauskommandos funktionieren auch

Ein Arbeitsblatt kann enthalten:

- Maple-Eingabe
  Eingabe-Prompt [>
  Kommentare: alles hinter # in Eingabezeile wird ignoriert
- Text (in schwarz)
- Resultate der Maple-Rechnungen (in blau)
- Graphische Ausgabe (Plots)

Arbeitsblätter II

Strukturierung eines Arbeitsblattes

- Zusammenfassen von Maple-Kommandos können zu einer Execution Group
  Menuleiste `Edit` → `Split or Join ...`
  oder Abschluss der Eingabe mit `<shift return>` statt `<return>`
  Nach Änderung eines Kommandos und `return` wird ganze Gruppe neu ausgeführt
- Section und Subsection
  Arbeitsbätter können in Kapitel und Unterkapitel eingeteilt werden
  Menuleiste `Insert` → `...`
  Anzeige der (Unter)-Kapitel kann durch Clicken auf ⊞ an
  bzw. durch Clicken auf ⊟ abgeschaltet werden
- Alle Ausgaben (Rechnungen und Plots) des Arbeitsblattes können entfernt werden (z.B. vor Abspeichern) Menuleiste `Edit` → `Remove Output` → `...`

Beispiel: Strukturierung eines Arbeitsblatt

BeispielArbeitsblatt.mws – [Server 1]

Das ist das ein Beispiel für ein strukturiertes Arbeitsblatt
Einfügen der Sections und Subsections über Menu–Leiste

Section 1: Plotten von Wertepaaren und Funktionen

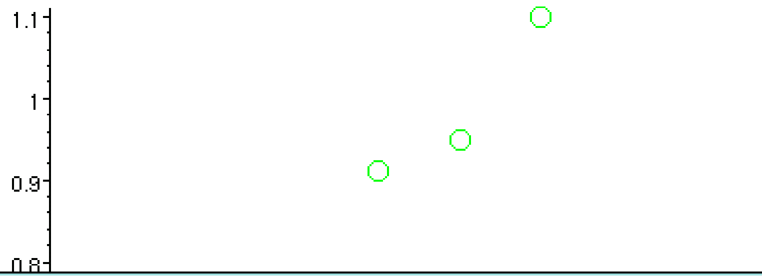
+ Erzeugen einer Liste mit seq–Befehl

– Plotten von Datenpunkten und Funktionen

Erzeugen einer Liste mit den Wertepaaren

```
> punkte:=[[1,0.53],[1.5,0.65],[2,0.91],
  [2.5,0.95],[3,1.10]];
> plot(punkte,x=0..4,color=green,style=point,
  symbol=CIRCLE,symbolsize=20);
```

$$punkte := [[1, .53], [1.5, .65], [2, .91], [2.5, .95], [3, 1.10]]$$



Lösen von Gleichungen

- Lösen von Gleichungen mit dem Befehl `solve`
  2. Argument gibt Variable(n) an, nach der aufgelöst werden soll
  Sonst gibt Maple Lösungen aller Unbekannten an
- Lösen von Gleichungssystemen
  Darstellung eines Gleichungssystems als Menge von Gleichungen
- Lösung kann durch einsetzen mit `subs` überprüft werden
- Numerisches Lösen von Gleichungen durch `fsolve`

Untitled (1) – [Server 1]

Lösen einer einfachen Gleichung

```
> gl := x^2 +2*x+a=0;
```

$$gl := x^2 + 2\,x + a = 0$$

Die Gleichung hat 2 Lösungen (Lösungsmenge ls)

```
> ls := solve(gl,x);
```

$$ls := -1 + \sqrt{1-a}, -1 - \sqrt{1-a}$$

Auf die beiden einzelnen Lösungen kann so zugegriffen werden

```
> x1:= ls[1];x2 := ls[2];
```

$$x1 := -1 + \sqrt{1-a}$$

$$x2 := -1 - \sqrt{1-a}$$

Auch Gleichungssysteme können gelöst werden

```
> gls := { 2*x - 3*y = 4, x + 2*y = -2};
```

$$gls := \{x + 2\,y = -2, 2\,x - 3\,y = 4\}$$

```
> lss := solve(gls);
```

$$lss := \{y = \frac{-8}{7}, x = \frac{2}{7}\}$$

Überprüfung der Lösung

```
> subs(lss,gls);
```

$$\{-2 = -2, 4 = 4\}$$

Numerisches Lösen mit fsolve am Beispiel einer transzendenten Gleichung

```
> gl := x*exp(x) =sqrt(2);
```

$$gl := x\,e^x = \sqrt{2}$$

```
> fsolve(gl,x);
```

$$.7013383834$$

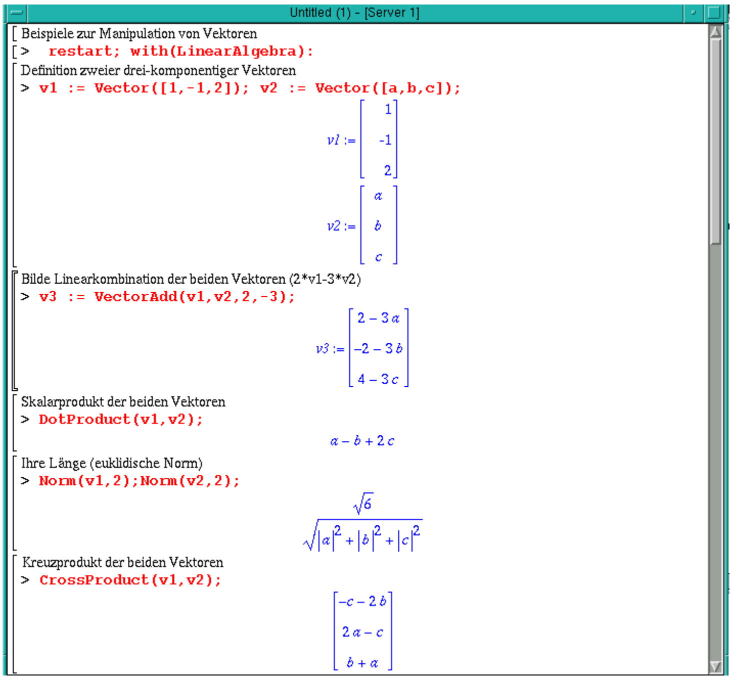
Lineare Algebra

Mit Maple können auch Rechnungen der Linearen Algebra ausgeführt werden

- Es gibt viele Kommandos für alle Manipulationen von Vektoren und Matrizen
- Enthalten im Paket: `LinearAlgebra`

Vektoren

- `Vector` definiert Vektoren als als Listen von Zahlen, Variablen, Funktionen, . . .
  Funktionen zum Erstellen von Sequenzen können benutzt werden
- Variablen wie a, b, c in dem Beipiel werden als komplexe Zahlen behandelt
- Addition von Vektoren und Multiplikation mit Skalaren mit `VectorAdd`
- Skalar- und Kreuzprodukt zweier Vektoren werden mit `DotProduct` und `CrossProduct` berechnet
- Den Betrag (Länge) eines Vektors erhält man auch mit `Norm(`*vektor*`, 2)`



Matrizen I

- Matrizen werden über Listen von Listen mit dem Befehl `Matrix` zeilenweise definiert

  Es gilt sinngemäß das für Vektoren gesagte
- Zugriff auf Komponenten über Indizierung mit eckigen Klammern
- Addition und Bildung von Linearkombinationen von Matrizen mit `MatrixAdd` oder Add
- Matrixmultiplikation mit `MatrixMatrixMultiply` oder `Multiply`

LinearAlgebra.mws - [Server 1]

Matrizen

```
> restart;with(LinearAlgebra):
```

Definiere ein quadratische und eine nicht-qaudratische Matrix

```
> A := Matrix([[1,2,3],[4,5,6],[-1,2,0]]);
  B := Matrix([[-1,2],[5,-2],[1,0]]);
```

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ -1 & 2 & 0 \end{bmatrix}$$

$$B := \begin{bmatrix} -1 & 2 \\ 5 & -2 \\ 1 & 0 \end{bmatrix}$$

Zugriff auf einzelne Komponenten

```
> A[1,3];
```

$$3$$

Addition und Liniearkombination von Matrizen

```
> Add(A,A,2,3); # entspricht MatrixAdd
```

$$\begin{bmatrix} 5 & 10 & 15 \\ 20 & 25 & 30 \\ -5 & 10 & 0 \end{bmatrix}$$

Matrixmultiplikation

```
> Multiply(A,B);    #entspricht MatrixMatrixMultiply
```

$$\begin{bmatrix} 12 & -2 \\ 27 & -2 \\ 11 & -6 \end{bmatrix}$$

```
>
```


Matrizen II

- Die Befehle `Transpose`, `Inverse`, `Determinant`, `Trace` berechnen die transponierte (A^{T}) und inverse (A^{-1}) Matrix bzw. die Determinante und Spur der Matrix

  Außer A^{T} sind die anderen Operation nur für quadratische Matrizen möglich

Untitled (1) - [Server 1]

```
>
```

Bei quadratischen Matrixen:
Berechnung der transponierten Matrix

```
> Transpose(A);
```

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

Berechnung der inversen Matrix

```
> MatrixInverse(A);
```

$$\begin{bmatrix} \frac{-4}{5} & \frac{2}{5} & \frac{-1}{5} \\ \frac{-2}{5} & \frac{1}{5} & \frac{2}{5} \\ \frac{13}{15} & \frac{-4}{15} & \frac{-1}{5} \end{bmatrix}$$

Berechnung der Determinaten

```
> Determinant(A);
```

$$15$$

Und Berechnung der Spur

```
> Trace(A);
```

$$6$$

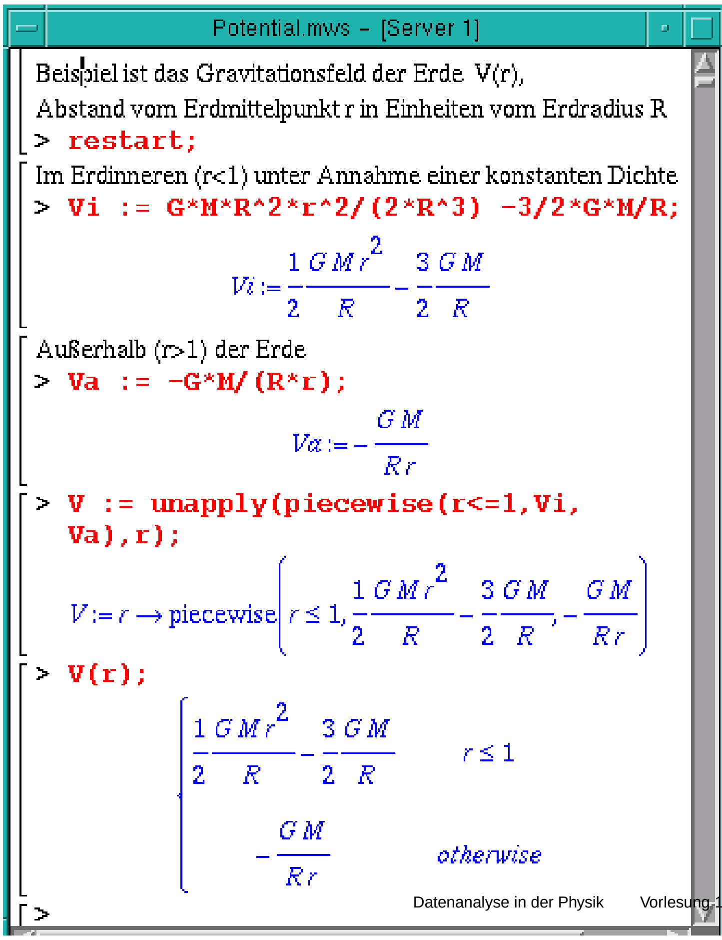
```
>
```

Funktionen und Prozeduren

Mehr über Definitionen von Funktionen

- Funktionen können mit dem Kommando `piecewise` abschnittsweise definiert werden
- Prozeduren
  Komplizierte Funktionen lassen sich in Prozeduren definieren
  In einer Prozedur lassen sich viele Bedingungen und Maple-Programmteile zusammenfassen
  Prozedur entspricht daher etwa einer Funktion in C
  Maple beinhaltet auch `if`-Verzweigungen und `for`-Schleifen

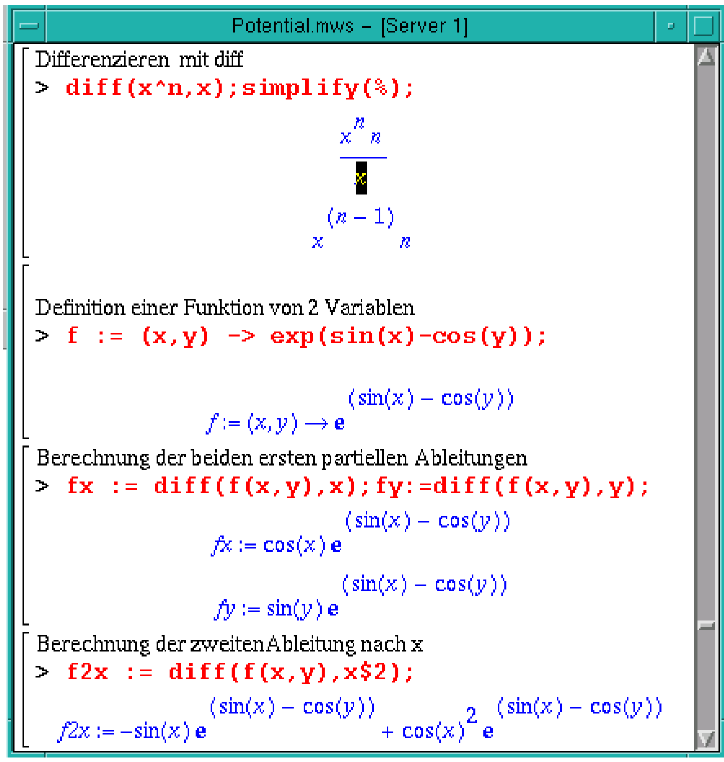
  Mehr über Prozeduren in der Online-Hilfe
  `?procedure`, `?if`, `?for`



Analysis

Eine besonders nützliche Eigenschaft von Maple ist die Möglichkeit, Funktionen zu Differenzieren und Integrieren

- Differenzieren von Funktionen oder algebraischen Ausdrücken mit dem Befehl `diff`
  symbolische Form ist `Diff`
- 2. Argument gibt Variable an, nach der abgeleitet werden soll
- Höhere Ableitungen werden mit dem Wiederholungsoperator `$` gebildet



Differenzieren II

- Daneben gibt es noch den Differentialoperator D, der nur auf Funktionen angewendet werden kann
- Höhere Ableitung können auch hier mit dem Wiederholungsoperator $ gebildet werden
- oder mit dem Operator @@:

  `D[1](f)@@2;`

  ist äquivalent zu

  `D[1$2](f);`

Potential.mws - [Server 1]

Verwendung des Differentialoperators D
Die Ableitungen werden in Funktionen abgebildet

```
> fx := D[1](f); fy := D[2](f);
```

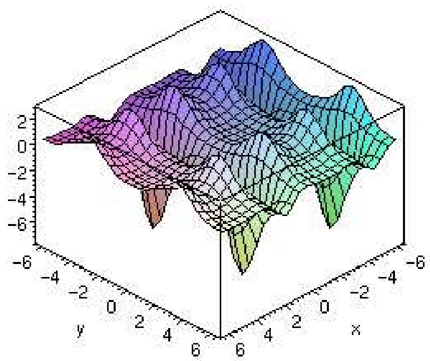
$$fx := (x, y) \to \cos(x)\, e^{(\sin(x) - \cos(y))}$$

$$fy := (x, y) \to \sin(y)\, e^{(\sin(x) - \cos(y))}$$

```
> f2x := D[1$2](f);
```

$$f2x := (x, y) \to -\sin(x)\, e^{(\sin(x) - \cos(y))} + \cos(x)^2\, e^{(\sin(x) - \cos(y))}$$

```
> plot3d(f2x(x,y),x=-2*Pi..2*Pi,
  y=-2*Pi..2*Pi,axes=BOXED);
```



Integrieren

- Stammfunktionen mit dem Befehl `int`
  Symbolische Form `Int`
  2. Argument ist die Integrationsvariable
- Bestimmte Integrale durch Angabe der Grenzen der Integrationsvariablen
- Mehrfachintegrale durch Verschachteln des `int`-Befehls
- Numerisches Integrieren mit `evalf`

  Tipp:
  `evalf(Int(...))` ist schneller als `evalf(int(...))` weil keine Stammfunktion gesucht wird

Untitled (1) - [Server 1]

Unbestimmtes Integral

```
> Int(cos(ln(x)),x)=int(cos(ln(x)),x);
```

$$\int \cos(\ln(x))\, dx = \frac{1}{2}\cos(\ln(x))\, x + \frac{1}{2}\sin(\ln(x))\, x$$

Bestimmetes Integral

```
Int(x/(exp(x)+1),x=0..infinity) =
int(x/(exp(x)+1),x=0..infinity);
```

$$\int_0^\infty \frac{x}{e^x + 1}\, dx = \frac{1}{12}\pi^2$$

Mehrfachintegral

```
Int(Int(x*y,y=0..sqrt(a^2-x^2)),x=0..1)=
int(int(x*y,y=0..sqrt(a^2-x^2)),x=0..1);
```

$$\int_0^1 \int_0^{\sqrt{a^2 - x^2}} x\, y\, dy\, dx = -\frac{1}{8} + \frac{1}{4}a^2$$

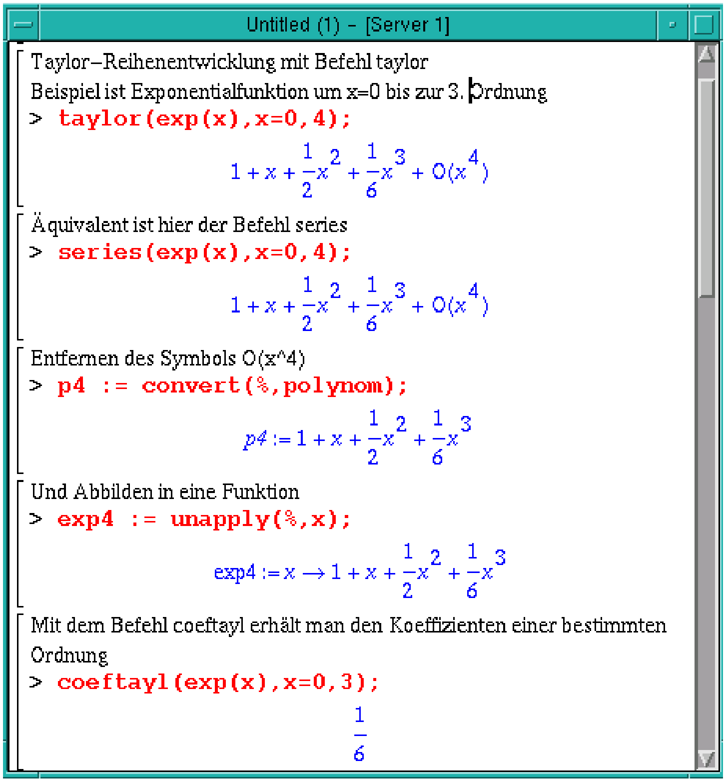
Numerische Berechnung eines bestimmten Integrals

```
Int(exp(-sin(x)),x=-Pi..Pi) =
evalf(Int(exp(-sin(x)),x=-Pi..Pi));
```

$$\int_{-\pi}^{\pi} e^{(-\sin(x))}\, dx = 7.954926521$$

Reihenentwicklung

- Mit Maple können Taylor-Entwicklungen von Funktionen durchgeführt werden
  `taylor`
- Alternative ist der allgemeinere Befehl `series`
- Argumente sind die Stelle, um die entwickelt werden soll, und die Zahl der Koeffizienten
- Mit `convert(%,polynom)` kann die Darstellung des ersten fehlenden Gliedes entfernt werden
- Mit `coeftaylor` wird nur ein Koeffizient berechnet



Universität Hamburg

Differentialgleichungen

- Differentialgleichungen (DGL) sind Gleichungen, die Funktionen und deren Ableitungen enthalten
- DGLs sind von besonderer Bedeutung in der Physik

Maple löst DGLs und ganze Systeme von gekoppelten DGLs

- Aufstellen der DGL mit `diff` oder `D`
- Lösen mit dem Befehl `dsolve` Argumente
  1. Gleichungen und Randbedingungen
  2. Funktion, nach der gelöst werden soll (falls nicht eindeutig)
  3. Optionen (siehe `?dsolve`)
     Z.B. kann, falls keine analytische Lösung gefunden wird, mit `type=series` ein Reihenansatz oder mit `type=numeric` eine numerische Lösung versucht werden
- Ohne Randbedingungen gibt Maple die Lösung mit den Integrationskonstanten *_C1*, *_C2*, ... an
- Übergabe der DGL, oder des Systems von DGLs, mit den Randbedingungen als Menge (set)

Universität Hamburg

Beispiel zur Lösung einer DGL

- Aufstellen der DGL
- Lösen ohne Randbedingungen
  Integrationskonstante *_C1*
- Setzen der Randbedingungen und Übergabe mit DGL als Menge von Gleichungen
- Maple erechnet Lösung als Gleichung *s2*
  `rhs` extrahiert rechte Seite der Gleichung, so dass mit `unapply` eine Funktion definiert werden kann

Untitled (1) - [Server 1]

```
> restart;
```

Beispiel zur Lösung einer nichtlinearen Differentialgleichung 1.Ordnung

```
> dgl := f(x) * diff(f(x),x) * (1+x^2) = x;
```

$$dgl := f(x)\left(\frac{\partial}{\partial x} f(x)\right)(1+x^2) = x$$

Lösen der DGL ohne Anfangsbedingung

```
> s1 := dsolve(dgl);
```

$$s1 := f(x) = \sqrt{\ln(1+x^2) + \_C1}, f(x) = -\sqrt{\ln(1+x^2) + \_C1}$$

Setzen einer Anfangsbedingung

```
> bed := f(0)=1;
```

$$bed := f(0) = 1$$

```
> s2 := dsolve({dgl,bed});
```

$$s2 := f(x) = \sqrt{\ln(1+x^2) + 1}$$

Abbilden der gefundenen Lösung in einer Funktion g(x)
Benutzt den Befehl rhs um die rechte Seite der Gleichung s2 zu isolieren

```
> g := unapply(rhs(s2),x);
```

$$g := x \rightarrow \sqrt{\ln(x^2+1) + 1}$$

```
> g(0);
```

$$1$$

Hilfsfunktionen zum Lösen von DGLs

Lösen von DGLs ist auch für Maple schwierig

Maple stellt Befehle zur Suche nach Lösung und deren Auswertung zur Verfügung

- Im Paket `DEtools` gibt es Hilfsfunktionen zur Lösung von DGLs
  z.B. mit dem Befehl `odeadvisor(`*dgl*`)` eine Klassifizierung der DGL durchgeführt und `odeadvisor(`*dgl*`,help)` gibt Hilfestellung zur Lösung
- Der Befehl `odeplot` im Paket `plots` stellt numerische Lösung einer gewöhnlichen DGL graphisch dar

Beispiel aus der Physik:
Bahnkurven im Gravitationsfeld der Erde

`Kepler.mws`