

▼ Vorlesung

PD Dr. Ulf Lorenz
Sprechstunde: nach Vereinbarung
Email: lorenz@mathematik.tu-darmstadt.de
Raum 35, Dolivostr. 15

Anmeldungen zur Vorlesung: Melden Sie sich in TUCaN (<https://www.tucan.tu-darmstadt.de>) **zum Modul und zur Veranstaltung an**, um später an der Prüfung teilnehmen zu können!

▼ Assistent

Dipl.-Math. Thomas Opfer
Sprechstunde: nach Vereinbarung
Email: opfer@mathematik.tu-darmstadt.de
Raum 40, Dolivostr. 15

▼ Übungen

▼ Anmeldung:

Erfolgt nicht über TUCaN, sondern gesonderte Anmeldung im EVS.

↔ <http://www.mathematik.tu-darmstadt.de/EMS>

Überprüfen Sie, dass Ihre Anmeldung erfolgreich war.

↔ <https://www3.mathematik.tu-darmstadt.de/evs/e/3>

▼ Übungsgruppenleiter:

Stefan Lüttgen, Johanna Stumpf, Amos Poznanski, Alexej Tempel, Lars Burghardt, Ellen Carmesin, Monika Prager

Welcher Übungsgruppenleiter welche Übungsgruppe hält, und zu welchen Zeiten, entnehmen Sie bitte ebenfalls den Webseiten zu dieser Veranstaltung.

▼ Übungsbetrieb:

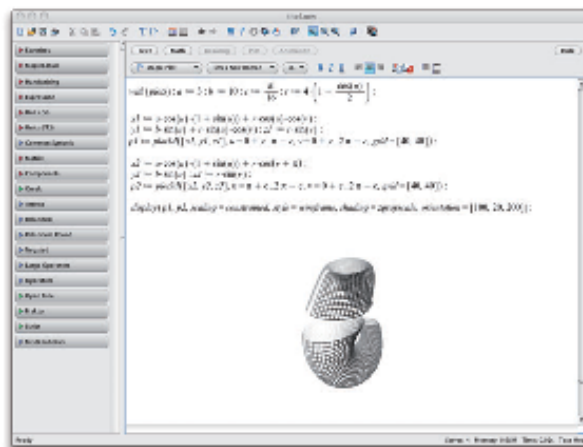
- Beginn: 22.10.2012
- Ausgabe der Übungen: Mi 19:00 Uhr (im Netz, keine Abgabe)
- Kleingruppenarbeit ist sinnvoll und wird eingefordert, in der Prüfung ist aber jeder allein
- **Sie können nur durch „Selbermachen“ lernen, nur lesen und zuhören reicht nicht!**

▼ Literatur

- Es gibt Begleitmaterial (Maplesheets).
- Skript: "Maple: Eine Einführung in das Computer-Algebra-System". U. Lorenz, T. Ederer, C. Juretzka, T. Opfer, M. Utz, S. Weber
RRZN-Handbuch, Kosten: 4,- Euro
Bezugsstelle: HRZ-Service, S1 | 03, Raum 020

Maple

Eine Einführung in das Computer-Algebra-System



1. Auflage
April 2011

- Buch: Introduction to Maple, 3rd Edition. André Heck. ISBN 0-387-00230-8

▼ Prüfung

- **im Anmeldezeitraum in TUCaN zur Prüfung anmelden!**
- Testaufgaben mit Maple, zum Ende der Vorlesung.
- Es werden sowohl **handwerkliche Fähigkeiten mit Maple**, als auch **Hintergrundwissen aus den Vorlesungen** abgeprüft.
- 3 ECTS, 1V + 1Ü, Schwerpunkt auf Übungen in Kleingruppen

▼ Modulhandbuch

Matrixarithmetik und lineare Gleichungssysteme, Unterschiede zwischen symbolischem und numerischem Rechnen, Differenzieren und Integrieren, Grenzwerte und Reihen, Graphik und Visualisierung, Definition von Funktionen und Programmierung.

▼ Ausblick für Vorlesung

- Orga, 1. Schritte
- Datentypen, Datenstrukturen, Numerisches/Symbolisches Rechnen, Programmieren
- Extremwertaufgabe Weizenglas designen
- Matrixoperationen und Anwendung, z.B. Abstand Punkt zu Gerade
- Merkwürdige Ergebnisse beim Rechnen mit Maple. Wie kommen die zu Stande?
- Google: Wie funktioniert so was?

Maple

Properties

- Software package
 - implemented in the programming language C
 - available for many Operating Systems, e.g. Windows, Unix, Linux
 - designed for numerical and **symbolic** expressions
- includes utilities for algebra, calculus, discrete mathematics, graphics, ...

History

- 1980: first development at the University of Waterloo, Canada
- 1988: Waterloo Maple Software was founded in order to sell and improve the software
- currently: Version 16

Getting started

Under Windows, we find the program Maple in the start menu, under Mac OS the program can be found in the application directory. Under Linux, we start Maple via the command `xmaple` inside a terminal shell:

- login to one of the machines in K313 or K309 in the Math building
- open a shell / a terminal
- type: `xmaple` (or `maple`, if you would like to work without windows; e.g. remote from home)

Menu bar at the top:

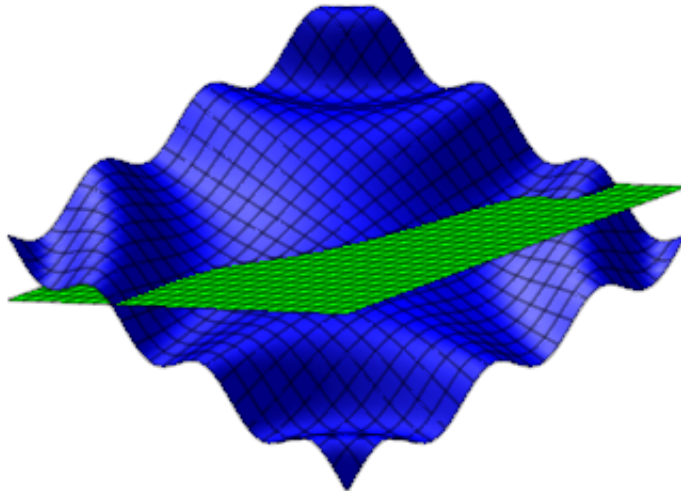
- allows you to save or load and edit your maple session
e.g. clicking on the **File** menu and selecting **Save** allows to save the current worksheet
- below the menu bar, there is a collection of shortcut-buttons

Maple Help

- help menu, "Maple Help"
 - `?command`; e.g. `?solve`, if you know the keyword in advance
- the help-window has two panels: the Help Navigator on the left and the help itself on the right
- each help page contains some examples; copying an example and pasting it into the worksheet is possible

#

```
plot3d([sin(x y), x + 2 y], x = -π..π, y = -π..π, color = [blue, green])
```



Basic Conventions

Entering a command, example

```
[> restart;  
 > 3 + 4;
```

7

(1)

Document Mode

in text mode, our input is not interpreted by Maple
 $4 + 7$; #in math mode, it is

11

(2)

Arithmetic operators



<i>Addition</i>	+	3 + 4
<i>Subtraction</i>	-	x - y
<i>Multiplication</i>	*	2*x
<i>Division</i>	/	x / y
<i>Exponentiation</i>	^	3^4
<i>Factorial</i>	!	3!

The precedence order follows the mathematical conventions:

```
[> (10 - 1) * 2
=
= 18 (3)
> 10 - 1 * 2
=
= 8 (4)
>
```

Rational Numbers:

```
[> 13 / 3;
=
= 13 / 3 (5)
> 13 / 2 + 23 / 7;
=
= 137 / 14 (6)
> % + 1 / 14;
=
= 69 / 7 (7)
```

Special commands to access previous results

- % latest one
- %% second most recent
- %%% third most recent

```
[> #this is a comment
=
> 2 * 4; # most recent result becomes 8
=
= 8 (8)
> % * 12; # this computes 8 * 12. 96 becomes most recent result
```

```

|                                     96                                     (9)
|=====
| > %% - %, # computes 8-96
|                                     -88                                   (10)
|=====
| >  $\frac{13}{2} + \frac{23}{7}$ ;
|                                      $\frac{137}{14}$                                    (11)
|

```

Floating point representation, Computing with Decimal Numbers

There are several possibilities how to enforce floating point representations. Most explicit is the command evalf.

```

[ > evalf(  $\frac{13}{2} + \frac{27}{13}$  );
|                                     8.576923077                                   (12)

```

There are hundreds of such functions / procedures / commands.

Moreover, there are pre-defined variables like Digits.

```

[ > Digits := 20;
|                                     Digits := 20                                   (13)
|=====

```

```

| > evalf(  $\frac{13}{2} + \frac{27}{13}$  );  $\frac{1.0}{3}$ ;  $\frac{1000.0}{3}$ ;
|                                     8.5769230769230769231
|                                     0.33333333333333333333
|                                     333.333333333333333333 (14)
|=====

```

```

| > Digits := 10;
|                                     Digits := 10                                   (15)
|=====

```

```

| > 10000!;
| 28462596809170545189064132121198688901480514017027992307941799942744113400037\ (16)
| 64443772990786757784775[...35460 digits...]
| 000000000000000000000000000000000000000000000000000000000000000000000000
| 000000000000000000000000000000000000000000000000000000000000000000000000
|

```

Expressions, functions and variables, Symbolic Computing

Defining Expressions with ":="

- expression: combination of numbers, variables and operators
- Syntax is *name:=expression*
- maybe most used concept in Maple

Example


```

> restart;
> f := (1 - x)^2 * x^5;
f := (1 - x)^2 x^5 (17)
> f; #careful! f(3) does not work
(1 - x)^2 x^5 (18)
> g := x -> x^2; # a 'function'
g := x -> x^2 (19)
> g(5);
25 (20)
>
> h := (x, y) -> x^2 - 2 * y^(1/3);
h := (x, y) -> x^2 - 2 y^(1/3) (21)
>

```

If you make a mistake, you can go back with the cursor, change the command-line and re-execute the line.

Example for function fitting

restart; with(plots) : with(Statistics) :

$X := \langle -3, -2, -1, 0, 1, 2, 3 \rangle;$

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

$Y := \langle 21, 8, 4, 2.5, 1.8, 1.6, 1.5 \rangle;$

$$\begin{bmatrix} 21 \\ 8 \\ 4 \\ 2.5 \\ 1.8 \\ 1.6 \\ 1.5 \end{bmatrix}$$

$f := a + \exp(c \cdot t);$

$$a + e^{ct} \quad (24)$$

`fit := Fit(f, X, Y, t);`

$$1.34146297217311 + e^{-0.990305984052394 t} \quad (25)$$

`display(plot(fit, t=-4..4, smartview=false), pointplot(X, Y, symbol=circle, symbolsize=15));`

