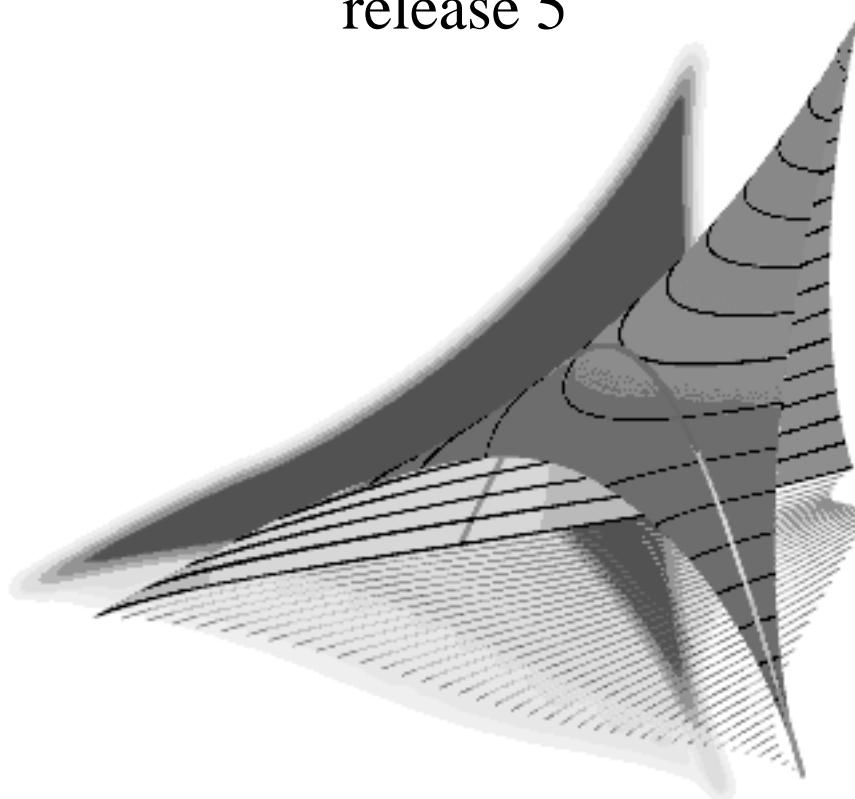




Användarhandledning för

# Maple V

release 5



Utgåva 5, juni 1998  
Sammanställd av  
Linda Kann  
Viggo Kann  
Jonas Sjöstrand

© 1998 Nada, KTH, Stockholm

Utgåva 5, juni 1998

Synpunkter på denna användarhandlednings innehåll och utformning är välkomna. Skriv dem i ett datorbrev till

`linda@nada.kth.se`

eller på en lapp adresserad till *Linda Kann, Nada* som du lämnar till någon handledare eller lärare på KTH eller lägger själv i postfacket märkt *Linda Kann* på Nada (Osquars backe 2, plan 4). Du kan också skicka lappen direkt till

*Linda Kann  
Nada, KTH  
100 44 Stockholm*

# Innehållsförteckning

Innehållsförteckning	3
1. Inledning	4
2. Start av Maple	5
3. Inskrivningssätt	9
4. Aritmetiska uttryck och elementära funktioner	10
5. Egendefinierade funktioner	12
6. Förenklingskommandon	13
7. Derivering, integration och summation	15
8. Ekvationslösning	16
9. Serieutveckling och gränsvärden	20
10. Datastrukturer	21
11. Grafik	23
12. Programmering	25
13. Att spara på fil och skriva ut	29
14. Hjälp	30
Bihang A. Laborationer	31
Bihang B. Övningsuppgifter	37
Bihang C. Operatörer och reserverade ord	41
Bihang D. Funktioner	42
Bihang E. Lathund	45
Sakregister	48

# 1 Inledning

## 1.1 Vad är Maple?

Maple är ett system för formelbehandling som har utvecklats vid Waterloo-universitetet i Kanada. Den första kommersiella versionen av programmet (Maple 3.3) kom 1985 och sedan dess pågår forskning och utveckling av programmet i samarbete med ETH i Zürich. Namnet Maple sägs kunna härledas ur beskrivningen "Mathematical Manipulation Language" men i själva verket valdes namnet för att det har kanadensisk klang.

Maple kan räkna både symboliskt och numeriskt med godtycklig precision. Dessutom finns kommandon för 2- och 3-dimensionell grafik. Biblioteksfunktioner (för till exempel linjär algebra, statistik och geometri) kan laddas in vid behov (för att Maple inte ska ta större datorresurser i anspråk än nödvändigt). Ofta används Maple interaktivt (man ger ett kommando som direkt tolkas och utförs) men man kan också skriva program i Maple.

Varje stort program innehåller buggar och Maple är inget undantag. Därför är det viktigt att aldrig lita på Maples resultat utan att ha kollat rimligheten i dem. Kolla till exempel genom att göra överslagsräkningar för hand, beräkna specialfall numeriskt, sätta in ekvationslösningar i ekvationen igen, rita upp resultaten, jämför med fysiska experiment och sunt förnuft.

Maple kan köras på datorer med följande operativsystem: Unix (även Linux), 386 DOS, Windows (95, NT och 3.1) och Macintosh.

I denna handledning beskrivs främst hur man använder Maple V release 5 under X, Windows och på Macintosh. De matematiska kommandona är dock samma i alla miljöer.

## 1.2 Litteraturlista

- K.M. Heal, M.L. Hansen, K.M. Rickard. *Maple V Learning Guide*. Springer-Verlag (1996)
- M.B. Monagan, K.O. Geddes, K.M. Heal, G. Labahn, S.M. Vorkoetter. *Maple V Programming Guide*. Springer-Verlag (1996)
- D. Redfern. *The Maple Handbook - Maple V release 4*. Springer-Verlag (1995)
- Robert M. Corless *Essential Maple*. Springer-Verlag (1995)
- André Heck *Introduction to Maple*. Springer-Verlag (1993)
- Robert J. Lopez *Maple via Calculus*. Birkhäuser (1994)
- Walter Gander, Jirí Hřebíček *Solving Problems in Scientific Computing Using Maple and Matlab*. Springer-Verlag (1993)
- Dominik Gruntz, Kathrin Meier, Michael Monagan. *Maple: An Introduction*. Institute for Scientific Computation, ETH Zürich (1992)
- Michael Monagan. *Programming in Maple: The Basics*. Institut für Wissenschaftliches Rechnen, ETH Zürich
- URL i World wide web: <http://daisy.uwaterloo.ca>  
<http://www.maplesoft.com>

En fullständigare lista över Maple-böcker finns på  
<http://www.maplesoft.com/Products/Maple/MapleBooks.html>

## 2 Start av Maple

Här följer fyra beskrivningar av hur man startar och arbetar med Maple i olika miljöer: X (under UNIX och VMS), Windows (på PC), Macintosh och på textterminal.

### X under Unix och VMS

Starta Maple genom att skriva `xmaple` eller `xmaple&` (det senare startar Maple i bakgrunden under Unix).

När Maple-fönstret kommit upp visar systemet med tecknet `>` att det är redo att ta emot kommandon.

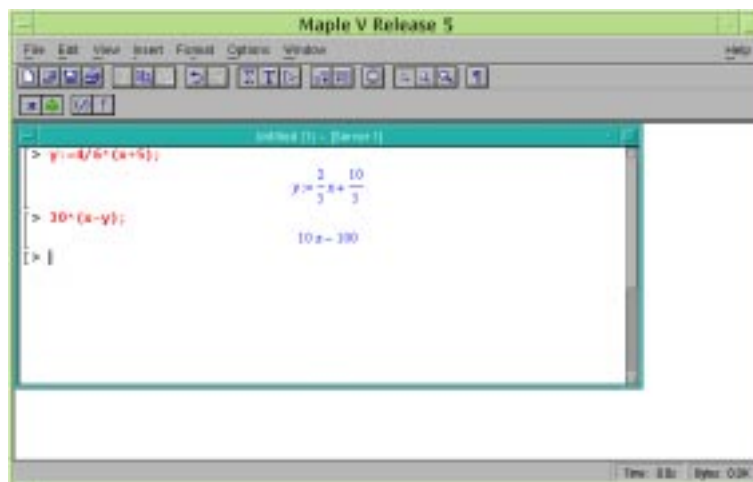
**OBS! Varje Maplekommando måste avslutas med ett semikolon;**

Semikolonet talar om för Maple att inmatningen är slut. Skriv `y:=4/6*(x+5);` följt av en returtryckning. Maple svarar med

$$y:=\frac{2}{3}x+\frac{10}{3}$$

Nu har `y` fått ett värde. Om nu `y` används i ett uttryck kommer värdet att sättas in. Prova det genom att beräkna uttrycket `30*(x-y);` Observera att Maple skiljer på gemena och versala bokstäver. Variabeln `Y` är alltså inte detsamma som variabeln `y`.

Maple-fönstret ser ut så här:



Överst finns åtta rubriker till rullgardinsmenyer; **File**, **Edit**, **View**, **Insert**, **Format**, **Options**, **Window** och **Help**. Därinunder ligger en rad med nitton symboler som var och en motsvarar ett kommando. På tredje raden finns ytterligare fyra kommandosymboler. Om du vill se vilket kommando en viss symbol motsvarar kan du välja **Balloon Help** i menyn **Help** och placera musmarkören på aktuell symbol (utan att klicka på den).

När Maple arbetar med en beräkning omvandlas musmarkören, som normalt är en pil, till en klocka. Om du tycker att beräkningen tar för lång tid kan du klicka på **STOP**-symbolen (den femtonde på första symbolraden).

Det är enkelt att redigera det man gjort i Maple. Man kan flytta runt markören bland de kommandon man skrivit själv (man kan inte ändra i resultaten) både med piltangenterna och med musen. Under menyn **Edit** kan man välja **Cut**, **Copy** och **Paste** för att klippa och klistra. På vissa tangentbord suddar suddtangenten nästa tecken istället för föregående tecken i Maple. Man kan dock alltid använda **Ctrl-H** för att sudda föregående tecken.

För att avsluta Maple ger du kommandot `quit;` eller väljer **Exit** i menyn **File**.

Du som inte har kört Maple tidigare bör nu hoppa direkt till Bihang A, sida 31 och göra lab1.

## Windows på PC

Starta Maple via startmenyn eller genom att dubbelklicka på ikonen (den ser ut som ett lönnlöv). När Maplefönstret kommit upp visar systemet med tecknet `>` att det är redo att ta emot kommandon.

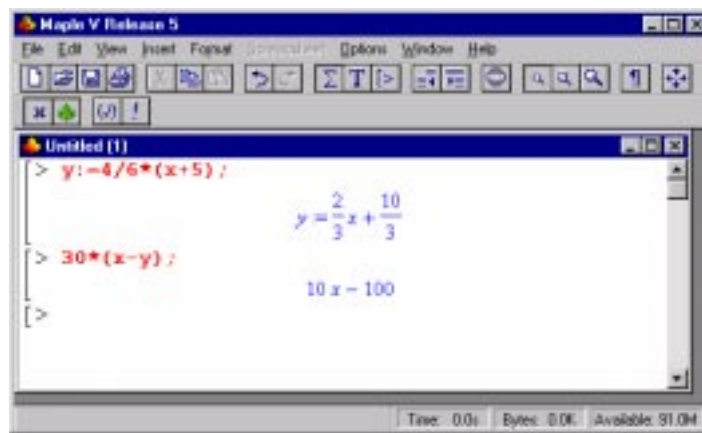
**OBS! Varje Maplekommando måste avslutas med ett semikolon;**

Semikolonet talar om för Maple att inmatningen är slut. Skriv `y:=4/6*(x+5);` och tryck sedan på Enter. Maple svarar med

$$y:=\frac{2}{3}x + \frac{10}{3}$$

Nu har `y` fått ett värde. Om nu `y` används i ett uttryck kommer värdet att sättas in. Prova det genom att beräkna uttrycket `30*(x-y);` Observera att Maple skiljer på gemena och versala bokstäver. Variabeln `Y` är alltså inte detsamma som variabeln `y`.

Maplefönstret ser ut så här:



Överst finns nio rubriker till rullgardinsmenyer; **File**, **Edit**, **View**, **Insert**, **Format**, **Spreadsheet**, **Options**, **Window** och **Help**. Därnunder ligger en rad med tjugo symboler som var och en motsvarar ett kommando. På tredje raden finns ytterligare fyra kommandosymboler samt ett enradsfönster som visar aktuellt uttryck. Om du vill se vilket kommando en viss symbol motsvarar kan du välja **Balloon Help** i menyn **Help** och placera musmarkören på aktuell symbol (utan att klicka på den).

När Maple arbetar med en beräkning omvandlas musmarkören, som normalt är en pil, till ett timglas. Om du tycker att beräkningen tar för lång tid kan du klicka på STOP-symbolen (den femtonde på första symbolraden).

Det är enkelt att redigera det man gjort i Maple. Man kan flytta runt markören bland de kommandon man skrivit själv (man kan inte ändra i resultaten) både med piltangenterna och med musen. Under menyn **Edit** kan man välja **Cut**, **Copy** och **Paste** för att klippa och klistra. För att avsluta Maple ger du kommandot `quit;` eller väljer **Exit** i menyn **File**.

Du som inte har kört Maple tidigare bör nu hoppa direkt till Bihang A, sida 31 och göra lab1.

## Macintosh

Starta Maple genom att dubbelklicka på ikonen som heter Maple (den ser ut som ett lönnlöv). När Maple-fönstret kommit upp visar systemet med tecknet `>` att det är redo att ta emot kommandon.

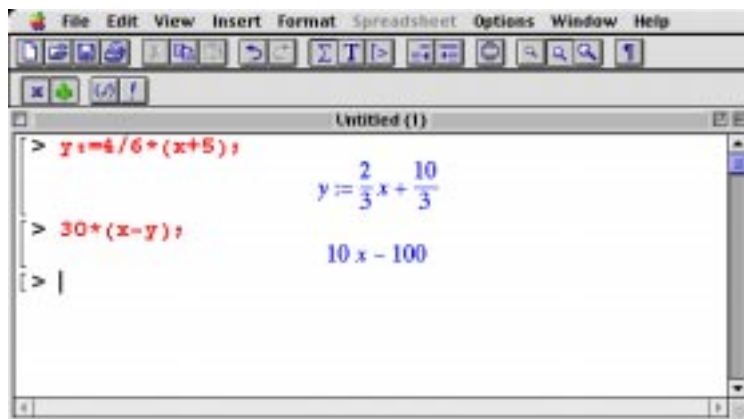
**OBS! Varje Maplekommando måste avslutas med ett semikolon;**

Semikolonet talar om för Maple att inmatningen är slut. Skriv `y:=4/6*(x+5);` och tryck sedan på Retur. Maple svarar med

$$y:=\frac{2}{3}x+\frac{10}{3}$$

Nu har `y` fått ett värde. Om nu `y` används i ett uttryck kommer värdet att sättas in. Prova det genom att beräkna uttrycket `30*(x-y);` Observera att Maple skiljer på gemena och versala bokstäver. Variabeln `Y` är alltså inte detsamma som variabeln `y`.

Maple-fönstret ser ut så här:



Överst finns nio rubriker till rullgardinsmenyer; **File**, **Edit**, **View**, **Insert**, **Format**, **Spreadsheet**, **Options**, **Window** och **Help**. Därnunder ligger en rad med nitton symboler som var och en motsvarar ett kommando. På tredje raden finns ytterligare fyra kommandosymboler samt ett enradsfönster som visar aktuellt uttryck. Om du vill se vilket kommando en viss symbol motsvarar kan du välja **Show balloons** i menyn **Help** och placera musmarkören på aktuell symbol (utan att klicka på den).

När Maple arbetar med en beräkning omvandlas musmarkören, som normalt är en pil, till en klocka. Om du tycker att beräkningen tar för lång tid kan du klicka på STOP-symbolen (den femtonde på första symbolraden).

Det är enkelt att redigera det man gjort i Maple. Man kan flytta runt markören bland de kommandon man skrivit själv (man kan inte ändra i resultaten) både med piltangenterna och med musen. Under menyn **Edit** kan man välja **Cut**, **Copy** och **Paste** för att klippa och klistra.

För att avsluta Maple ger du kommandot `quit;` eller väljer **Quit** i menyn **File**.

Du som inte har kört Maple tidigare bör nu hoppa direkt till Bihang A, sida 31 och göra lab1.

## Textterminal

Starta Maple genom att skriva `maple`

Maplesystemet skriver ut lite information och visar sedan med tecknet `>` att det är redo att ta emot kommandon.

**OBS! Varje Maplekommando måste avslutas med ett semikolon;**

Semikolonet talar om för Maple att inmatningen är slut. Skriv `y:=4/6*(x+5);` följt av en returtryckning. Maple svarar med

$$y := \frac{2}{3}x + \frac{10}{3}$$

Nu har  $y$  fått ett värde. Om nu  $y$  används i ett uttryck kommer värdet att sättas in. Prova det genom att beräkna uttrycket `30*(x-y);` Observera att Maple skiljer på gemena och versala bokstäver. Variabeln  $Y$  är alltså inte detsamma som variabeln  $y$ .

På skärmen ser det ut så här:

```

  |V|V| Maple V Release 5 (KTH)
  .-| | | | |.- Copyright (c) 1981-1997 by Waterloo Maple Inc. All rights
  \ MAPLE / reserved. Maple and Maple V are registered trademarks of
  <-----> Waterloo Maple Inc.
  | Type ? for help.
> y:=4/6*(x+5);
                                     y := 2/3 x + 10/3

> 30*(x-y);
                                     10 x - 100

> █

```

När Maple arbetar med en beräkning som tar lång tid skriver den ut hur mycket minne och tid den har använt. Om du tycker att beräkningen tar för lång tid och vill avbryta trycker du på `Ctrl-C` (eventuellt två gånger).

Man kan använda piltangenterna för att hämta tidigare kommandon och modifiera dem.

För att avsluta Maple ger du kommandot `quit;`

Nästan alla kommandon i Maple går bra att utföra på textterminal. En nackdel är dock att grafiken inte blir så snygg. Bilder som ritas med `plot` kan se hyfsade ut, men tredimensionella grafer (som ritats med `plot3d`) blir svårtolkade.

Du som inte har kört Maple tidigare bör nu hoppa direkt till Bihang A, sida 31 och göra lab1.

En textversion av Maple finns inbyggd i Maple V release 5 (under namnet Command Line Maple). Den är snabb och minnessnål och kan således vara användbar om man vill göra stora beräkningar eller har ont om minne.



## 3 Inskrivningssätt

Under menyn **View** kan man välja **Palettes** och **Symbol Palette**, **Expression Palette** och **Matrix Palette**. Då får man fram små fönster med matematiska symboler. Klickar man på en symbol i symbolpaletten skrivs mapleordet för symbolen i huvudfönstret. När man klickar på en symbol i uttrycks- eller matrispaletten skrivs motsvarande maplefunktion i huvudfönstret. Man behöver bara fylla i parametrarna.

Den som inte vill ha ful maplekod i huvudfönstret kan under menyn **Options** välja **Input Display** och **Standard Math**. Då kommer det fram en ruta upptill, där man skriver in maplekoden och trycker på retur. I huvudfönstret syns sedan bara vackra matematiska uttryck. Använder man paletter kommer uttrycket upp direkt i huvudfönstret där man kan redigera det. Man återställer inmatningsmoden genom att under menyn **Options** välja **Input Display** och **Maple Notation**.

## 4 Aritmetiska uttryck och elementära funktioner

Här följer en tabell över de grundläggande aritmetiska operatörer som Maple känner till:

$x+y$	addition
$x-y$	subtraktion
$x*y$	multiplikation
$x/y$	division
$x**y$	exponentiering
$x^y$	exponentiering
$x \bmod y$	modulo
$x!$	fakultet

Observera att Maple räknar exakt med heltal:

```
>3*4+5/6;
```

$$\frac{77}{6}$$

För att hänvisa till föregående resultat använder man procenttecken %, som även får förekomma i uttryck. (I tidigare versioner av Maple användes " i stället för %.)

```
>6*%/11;
```

7

Näst sista resultatet får man med två procenttecken %% och näst näst sista resultatet med tre procenttecken %%%. Om du vill ha ett decimaltal som resultat kan du låta ett decimaltal ingå i uttrycket:

```
>3.0*4+5/6;
```

12.83333333

Följande konstanter finns definierade i Maple:

I	$\sqrt{-1}$
Pi	$\pi$
infinity	$\infty$
true	<i>sant</i>
false	<i>falskt</i>

Observera att I och Pi ska skrivas med stor bokstav.

Prova att räkna med komplexa tal:

```
>(3+I)*(3-I);
```

10

Maple har många inbyggda funktioner. Här är en tabell över de vanligaste. En fullständig lista över Maples funktioner hittar du i Bihang D, sida 42.

<code>abs(x)</code>	absolutbeloppet	$ x $
<code>exp(x)</code>	exponentialfunktionen	$e^x$
<code>ln(x)</code>	naturliga logaritmen	$\ln x$
<code>log[10](x)</code>	tiologaritmen	$\log_{10} x$
<code>sqrt(x)</code>	kvadratroten	$\sqrt{x}$
<code>sin(x)</code>	sinus	$\sin x$
<code>cos(x)</code>	cosinus	$\cos x$
<code>tan(x)</code>	tangens	$\tan x$

```
>x:=ln(sqrt(sin(Pi/3)+cos(3*Pi/4)));
```

$$x := \frac{1}{2} \ln\left(\frac{1}{2}\sqrt{3} - \frac{1}{2}\sqrt{2}\right)$$

För att få en flyttalsapproximation till ett värde kan du använda kommandot `evalf`:

```
>evalf(x);
```

-.9196815065

Maple kan göra beräkningar i godtycklig precision, men använder normalt bara tio siffror. Genom att sätta om Maplevariabeln `Digits` kan man öka eller minska precisionen:

```
>Digits:=50;
```

*Digits:=50;*

```
>evalf(x);
```

-.91968150767026707665881288856609214194321712718395

Det tar förstås längre tid att utföra beräkningar i hög precision, så glöm inte att återställa `Digits` till tio när du inte längre behöver så hög noggrannhet.

Man kan tilldela flera variabler med ett enda kommando:

```
>x,y,z:=1,2,3;
```

## 5 Egendefinierade funktioner

Man har möjlighet att namnge och definiera egna funktioner i Maple. Detta gör man i en tilldelningsats, med funktionens namn i vänsterledet och definitionen i högerledet. I definitionen anger man först vilka variabler funktionen beror av, sen skriver man en pil (ett minustecken följt av tecknet större än) och sist själva funktionsuttrycket:

```
>poly:=x->x^6-4*x^5-2*x+8;
```

$$poly:=x \rightarrow x^6 - 4x^5 - 2x + 8$$

En egendefinierad funktion kan användas i beräkningar på samma sätt som Maples inbyggda funktioner:

```
>poly(2.75);
```

- 194.0954590

```
>poly(a-1)/poly(Pi);
```

$$\frac{(a-1)^6 - 4(a-1)^5 - 2a + 10}{\pi^6 - 4\pi^5 - 2\pi + 8}$$

Volymen av en lutande cirkulär cylinder beror av tre variabler: cylinderns radie  $r$ , sidlängden  $l$  och lutningsvinkeln  $\theta$ . Volymfunktionen kan definieras så här:

```
>V:=(r,l,theta)->Pi*r^2*l*cos(theta);
```

$$V:= (r, l, \theta) \rightarrow \pi r^2 l \cos(\theta)$$

Observera att Maple känner igen de engelska namnen på grekiska bokstäver och omvandlar till rätt symbol i utskriften.

Man kan ändra en funktions värde i en punkt.

```
>V(0,0,0):=1;
```

Tyvärr accepterar Maple också att man ändrar på en funktions värde av en variabel, t ex

```
>poly(t):=4*t;
```

**Undvik detta skrivsätt!** Det definierar inte en hel funktion utan bara vad  $poly(t)$  ska betyda. Det definierar alltså inte exempelvis  $poly(s)$  eller ens  $poly(t+1)$ .

Om man har ett uttryck som man vill göra om till en funktion kan man använda `unapply`.

```
>u:=(2*x+y^2)^3+y;
```

```
>f:=unapply(u,(x,y));
```

För en styckvis definierad funktion finns kommandot `piecewise`.

```
p:=piecewise(x<0,-1,x>1,2*x,x^2);
```

$$p:= \begin{cases} -1 & x < 0 \\ 2x & 1 < x \\ x^2 & \text{otherwise} \end{cases}$$

Det går också att programmera in mer komplicerade funktioner med hjälp av konstruktionen `proc`, se avsnitt 12.2, sida 26.

## 6 Förenklingskommandon

Maples utmatning blir inte alltid som man vill ha den. Det finns ett antal kommandon för förenkling eller omordning av ett uttryck. Här följer några av dem:

<code>expand(u)</code>	utvecklar uttrycket u
<code>normal(u)</code>	skriver om uttrycket som en kvot
<code>collect(u)</code>	samlar termer m a p en viss variabel
<code>combine(u)</code>	kombinerar termer
<code>factor(u)</code>	faktorerar polynom
<code>simplify(u)</code>	förenklar uttryck
<code>op(n,u)</code>	plockar ut den n:e operanden ur u
<code>rhs(u)</code>	plockar ut högerledet ur u
<code>lhs(u)</code>	plockar ut vänsterledet ur u
<code>subs(x=a,u)</code>	byter ut x mot a överallt i uttrycket u
<code>evalf(u)</code>	beräknar en flyttalsapproximation till u

Exempel:

```
>t:=sin(a)*(sin(a)+cos(a)*cos(a)/sin(a));
```

$$t := \sin(a) \left( \sin(a) + \frac{\cos(a)^2}{\sin(a)} \right)$$

```
>expand(t);
```

$$\sin(a)^2 + \cos(a)^2$$

```
>simplify(t);
```

1

Observera att anropen av `expand` och `simplify` inte förändrar t:s värde, utan bara returnerar det förenklade resultatet. Detsamma gäller övriga funktioner.

```
>normal(t/sin(a));
```

$$\frac{\sin(a)^2 + \cos(a)^2}{\sin(a)}$$

```
>collect((x+y)^6-(2*x^2-y^3)^2,x);
```

$$x^6 + 6x^5y + (-4 + 15y^2)x^4 + 20x^3y^3 + (15y^4 + 4y^3)x^2 + 6xy^5$$

```
>factor(%);
```

$$x(x^2 + 2x + 3xy + 3y^2)(x^3 + 3x^2y - 2x^2 + 3xy^2 + 2y^3)$$

Oftast används `combine` med en andra parameter som anger vilken typ av sammanslagning det är frågan om. Man kan bland annat välja bland `trig`, `exp`, `ln` och `power`.

```
>combine(sin(x+Pi)*cos(x-Pi), trig);
```

$$\frac{1}{2}\sin(2x)$$

Med `op` kan man få fram enskilda termer eller faktorer.

```
>s:=exp(a)/tan(a);
```

$$s := \frac{e^a}{\tan(a)}$$

Första operanden i uttrycket `s` är täljaren.

```
>op(1, s);
```

$$e^a$$

Andra operanden är nämnaren inverterad.

```
>op(2, s);
```

$$\frac{1}{\tan(a)}$$

```
>subs(a=2, s);
```

$$\frac{e^2}{\tan(2)}$$

```
>evalf(%);
```

$$-3.381657344$$

Vänsterledet i en ekvation fås med `lhs` och högerledet med `rhs`.

```
>lhs(a=b);
```

$$a$$

```
>rhs(a=b);
```

$$b$$

## 7 Derivering, integration och summation

Med kommandona `diff`, `int` och `sum` kan man beräkna derivator, integraler och summor (både bestämda och obestämda). Till dessa kommandon måste man skicka med två parametrar. Den första ska vara en funktion och den andra en variabel (som man deriverar/integrerar/summerar med avseende på).

<code>diff(f(x), x);</code>	deriverar $f$ m a p $x$
<code>int(f(x), x);</code>	integrerar $f$ m a p $x$
<code>int(f(x), x=a..b)</code>	integrerar $f$ då $x$ går från $a$ till $b$
<code>sum(f(n), n);</code>	summerar $f$ m a p $n$
<code>sum(f(k), k=a..b);</code>	summerar $f$ då $k$ går från $a$ till $b$

Exempel:

```
>diff(exp(x^2), x);
```

$$2xe^{x^2}$$

```
>int(x^3+5, x);
```

$$\frac{1}{4}x^4 + 5x$$

```
>int(ln(x), x=1..5);
```

$$-4 + 5\ln(5)$$

Vill man derivera flera gånger är det bara att lägga till variabler i parameterlistan. Exempel:

```
>diff(x*y*z*w, x, y, z);
```

$$w$$

Maple klarar i princip att hitta primitiva funktionen till varje funktion (förutsatt att den kan uttryckas i elementära funktioner). När en integral inte kan beräknas svarar Maple återigen med en integral.

```
>int(sqrt(tan(x)/ln(x)), x);
```

$$\int \frac{\sqrt{\tan(x)}}{\sqrt{\ln(x)}} dx$$

```
>sum(n^2, n);
```

$$\frac{1}{3}n^3 - \frac{1}{2}n^2 + \frac{1}{6}n$$

```
>sum(exp(-n), n=0..infinity);
```

$$\frac{e}{e-1}$$

Variabeln i `diff`, `int` och `sum` måste vara obestämd, dvs den får inte ha tilldelats ett värde. En variabel kan återställas genom ett kommando av typen `n := 'n'`;

## 8 Ekvationslösning

Maple kan lösa ekvationer och ekvationssystem, med både linjära och icke-linjära ekvationer och olikheter. Dessutom finns funktioner för lösning av ordinära differentialekvationer samt system av dessa. Se Bihang E, sida 45 för en fullständig lista över Maples ekvationslösningssfunktioner.

<code>solve(ekv, x);</code>	löser ekvationen $ekv$ m a p variabeln $x$
<code>solve({e1, e2}, {x1, x2});</code>	löser ekvationssystemet $e1, e2$ m a p $x1$ och $x2$
<code>fsolve(ekv, x);</code>	ger en numerisk lösning av ekvationen $ekv$ m a p $x$
<code>fsolve({e1, e2}, {x1, x2});</code>	ger en numerisk lösning av ekvationssystemet $e1, e2$ m a p $x1$ och $x2$
<code>dsolve(diffekv, y(x));</code>	löser differentialekvationen $diffekv$ m a p $y(x)$
<code>dsolve({diffekv, bv}, y(x));</code>	löser differentialekvationen $diffekv$ med begynnelsevillkor $bv$ m a p $y(x)$
<code>dsolve({d1, d2, bv1, bv2}, {f1, f2});</code>	löser ett system av differentialekvationer $d1, d2$ med begynnelsevillkor $bv1, bv2$ m a p $f1$ och $f2$

På alla platser i tabellen ovan där det står två ekvationer, variabler eller funktioner kan man också ha fler än två.

### 8.1 solve

```
> solve(sin(2*x-5)=1, x);
```

$$\frac{1}{4}\pi + \frac{5}{2}$$

Observera att `solve` bara ger en lösning till ekvationen ovan, inte hela lösningsmängden. Man kan i regel inte räkna med att Maple hittar alla lösningar till en ekvation.

```
> solve(x^2-3*x-4=0, x);
```

$$4, -1$$

När högerledet i ekvationen är noll räcker det att ge vänsterledet. Om ekvationen bara beror av en variabel behöver man inte ta med den heller. Uttrycket ovan kan därför förenklas till:

```
> solve(x^2-3*x-4);
```

$$4, -1$$

Om ekvationen inte kan lösas ger Maple inget svar alls.

```
> solve(abs(x)=-1);
```



Ett ekvationssystem måste omslutas av klamrar, likaså en uppräknings av variabler.

```
>solve({3*x-y=0, x+y+z=1, x-7*z=5}, {x, y, z});
```

$$\left\{ y = \frac{36}{29}, z = \frac{-19}{29}, x = \frac{12}{29} \right\}$$

Om svaret innehåller rötter av heltal löser Maple i vissa fall inte ut dessa utan använder istället `RootOf` för att uttrycka rötterna. Med `allvalues` får man lösningarna ur ett `RootOf`-uttryck.

```
>solve({x+y=2, y^2=50});
```

$$\{y=5\text{RootOf}(\_Z^2-2), x=-5\text{RootOf}(\_Z^2-2)+2\}$$

```
>allvalues(%);
```

$$\{y = 5\sqrt{2}, x = -5\sqrt{2} + 2\}, \{y = -5\sqrt{2}, x = 5\sqrt{2} + 2\}$$

När uttrycket som returneras av en Maplefunktion blir komplicerat kan Maple införa egna hjälpvariabler

```
>solve(sin(x)^3-sin(x)^2+1, x);
```

$$\arcsin\left(-\frac{1}{6}\%2 - \frac{2}{3}\%1 + \frac{1}{3}\right), \arcsin\left(\frac{1}{12}\%2 + \frac{1}{3}\%1 + \frac{1}{3} + \frac{1}{2}I\sqrt{3}\left(-\frac{1}{6}\%2 + \frac{2}{3}\%1\right)\right),$$

$$\arcsin\left(\frac{1}{12}\%2 + \frac{1}{3}\%1 + \frac{1}{3} - \frac{1}{2}I\sqrt{3}\left(-\frac{1}{6}\%2 + \frac{2}{3}\%1\right)\right)$$

$$\%1 := \frac{1}{(100 + 12\sqrt{69})^{1/3}}$$

$$\%2 := (100 + 12\sqrt{69})^{1/3}$$

Maples hjälpvariabler har namn som inleds med % följt av ett tal. När hjälpvariabler har använts i ett uttryck som Maple returnerar så finns de definierade på raderna under uttrycket.

## 8.2 fsolve

Flyttalslösningar fås med `fsolve`.

```
>fsolve(exp(x)-3*x, x);
```

1.512134552

I regel nöjer sig `fsolve` med att beräkna en rot, men man kan få fram andra rötter genom att ange vilket intervall man är intresserad av.

```
>fsolve(exp(x)-3*x, x=0..1);
```

.6190612867

Genom att lägga till argumentet `complex` till ett anrop av `fsolve` kan man även få ut komplexa rötter:

```
>fsolve(x^3+4*x^2-2*x+6, x);
```

-4.697635679

```
>fsolve(x^3+4*x^2-2*x+6, x, complex);
```

-4.697635679, .3488178394-1.074971800I, .3488178394+1.074971800I

Man kan spara rötterna i en *följd* (se avsnittet om datastrukturer), om man vill använda dem i senare beräkningar:

```
>rot:=fsolve(x^3+4*x^2-2*x+6,x,complex);
```

```
rot:=-4.697635679, .3488178394-1.074971800I, .3488178394+1.074971800I
```

```
>rot[1]*rot[2]*rot[3];
```

```
-6.000000002+.351510492 10-10I
```

Man kan skicka med en startgissning:

```
>fsolve(sin(x),x=3.1);
```

```
3.141592654
```

Om man är ointresserad av vissa rötter (eller om singularpunkter hotar) kan man betacka sig för dessa:

```
>fsolve(sin(x),x,avoid={x=0,x=2*Pi});
```

```
-3.141592654
```

## 8.3 dsolve

För att lösa differentialekvationen  $3y(x) + \frac{d^2}{dx^2}y(x) = 0$  skriver man:

```
>dsolve(3*y(x)+diff(y(x),x,x)=0,y(x));
```

```
y(x) = _C1 sin( $\sqrt{3}x$ ) + _C2 cos( $\sqrt{3}x$ )
```

Observera att Maple inför egna beteckningar, `_C1` och `_C2`, för konstanterna i lösningen. Dessa konstanter inleds med understrykningstecken för att man inte ska blanda ihop dem med de variabler man har definierat själv.

För att ange begynnelsevillkoren kan man behöva D-operatoren.

Exempel:  $y(3)$  skrivs som  $D(y)(3)$

Differentialekvationen ovan med begynnelsevillkoren  $y(3) = 1$  och  $y'(3) = 0$  blir, med D-operatoren:

```
>dsolve({3*y+D(D(y))(x),y(3)=1,D(y)(3)=0},y(x));
```

$$y(x) = \frac{\sin(3\sqrt{3})\sin(\sqrt{3}x)}{\cos(3\sqrt{3})^2 + \sin(3\sqrt{3})^2} + \frac{\cos(3\sqrt{3})\cos(\sqrt{3}x)}{\cos(3\sqrt{3})^2 + \sin(3\sqrt{3})^2}$$

Ofta när man använder `solve`-funktionerna får man ett svar som består av högerled och vänsterled.

Vill man plocka ut högerledet och spara det i en variabel kan man använda `rhs` (right hand side).

Funktionen `lhs` (left hand side) ger vänsterledet.

```
losning:=rhs(dsolve({D(D(y))(x)-y,y(2)=1,D(y)(0)=0},y(x)));
```

$$\frac{e^x}{e^2 + e^{-2}} + \frac{e^{-x}}{e^2 + e^{-2}}$$

Nu kan man använda funktionen `subs` för att ersätta `x` med `0` i uttrycket:

```
subs(x=0,losning);
```

$$2 \frac{e^0}{e^2 + e^{-2}}$$

och förenkla med `simplify`:  
`simplify(%);`

$$2\frac{1}{e^2 + e^{-2}}$$

Partiella differentialekvationer kan lösas med kommandot `pdesolve`. Mer information fås med `>?pdesolve`

## 9 Serietveckling och gränsvärden

<code>series(f(x), x=a);</code>	Serietvecklar uttrycket f kring x=a
<code>limit(f(x), x=a);</code>	Gränsvärdet av f när x går mot a

Kommandot `series` beräknar Taylorutvecklingen av ett uttryck kring en given punkt.

```
>series(exp(x^2), x=0);
```

$$1 + x^2 + \frac{1}{2}x^4 + O(x^6)$$

```
>series(x^x, x=1);
```

$$1 + x - 1 + (x-1)^2 + \frac{1}{2}(x-1)^3 + \frac{1}{3}(x-1)^4 + \frac{1}{12}(x-1)^5 + O((x-1)^6)$$

Man får automatiskt termer upp till 6:e ordningen; vill man ha fler (eller färre) kan man ange det med en tredje parameter:

```
>series(ln(cos(x)), x=0, 10);
```

$$-\frac{1}{2}x^2 - \frac{1}{12}x^4 - \frac{1}{45}x^6 - \frac{17}{2520}x^8 + O(x^{10})$$

För att kapa resttermen och kunna räkna med serien kan man konvertera den till ett polynom med kommandot `convert`

```
>convert(series(ln(cos(x)), x=0, 10), polynom);
```

$$-\frac{1}{2}x^2 - \frac{1}{12}x^4 - \frac{1}{45}x^6 - \frac{17}{2520}x^8$$

Kommandot `limit` beräknar gränsvärdet av ett uttryck.

```
>limit(sin(t)/t, t=0);
```

1

```
>limit(exp(x^(-x))*x!, x=infinity);
```

∞

```
>limit(tan(x), x=Pi/2);
```

*undefined*

## 10 Datastrukturer

Maple använder följande datastrukturer:

sequence	följd	1, 2, 2, 3
list	lista	[1, 2, 2, 3]
set	mängd	{1, 2, 3}
table	tabell	table([a=1 p=3]);
array	vektor, matris etc.	[1 2 2 3]

En följd är en samling Mapleobjekt åtskilda av kommatecken. Följder används till mycket: parametrarna man skickar med till en funktion utgör en följd, vissa Maplekommandon (t ex `solve`) kan returnera en följd som resultat mm. Det finns ett särskilt kommando, `seq`, som skapar en följd:

```
>seq(k^2, k=4..7);
```

16, 25, 36, 49

Men man kan också skapa en följd genom att räkna upp elementen med kommatecken emellan:

```
>s:=sqrt(x), 28.3, sin(n*Pi);
```

s:=  $\sqrt{x}$ , 283,  $\sin(\pi n)$

En lista är en följd omgiven av hakparenteser.

```
>l:= [seq(k mod 3, k=1..10)];
```

l:=[1, 2, 0, 1, 2, 0, 1, 2, 0, 1]

En mängd konstrueras som en följd men omges av klamrar istället. I en mängd tas alla multipler av element bort.

```
>m:={seq(k mod 3, k=1..10)};
```

m:={0, 1, 2}

Mängdoperationerna `union`, `intersect` (snitt) och `minus` finns definierade i Maple.

```
>m intersect {0,2,4,6,8};
```

{0, 2}

För att komma åt ett visst element i en datastruktur skriver man elementets ordningsnummer inom hakparenteser efter variabelnamnet:

```
>s[2], l[9], m[3];
```

28.3, 0, 2

En tabell är en indexerad variabel, där man kan ha index av valfri typ. Argumentet till kommandot `table`, som skapar en tabell, är en lista med tabellens index och motsvarande element:

```
> dagar:=table([ juni=30, juli=31, augusti=31]);
```

```
dagar:=table([
  augusti=31
  juni=30
  juli=31
])
```

```
> dagar[ juli ];
```

```
31
```

En array är ett specialfall av en tabell, nämligen en tabell som bara kan ha heltalsindex.

`convert(uttryck, typ)` ; kan användas för att konvertera mellan olika typer (t ex mellan en lista och en mängd).

```
> s := [8, a, -3, 7, 8, 17];
```

```
s := [8, a, -3, 7, 8, 17]
```

```
> t := convert(s, set);
```

```
t := {-3, 7, 8, 17, a}
```

`map(funktion, uttryck)` ; anropar funktionen för varje element i uttrycket (som kan vara vilken datstruktur som helst) och skapar en likadan struktur bestående av funktionsvärdena.

```
> map(x -> x^2, t);
```

```
{9, 49, 64, 289, a^2}
```

## Linalg-paketet

Vektorer och matriser finns inbyggda i `linalg`-paketet som laddas med kommandot `with(linalg)` ;. Då får man tillgång till ett hundratal funktioner från linjär algebra.

Exempel:

```
> v := vector([10, -6, 11, 4, 3]);
```

```
v := [10 -6 11 4 3]
```

```
> norm(v);
```

```
11
```

```
> norm(v, 2);
```

```
 $\sqrt{282}$ 
```

När man skapar en matris ska man som första argument ange matrisens dimensioner.

```
> A := matrix(3, 3, [1, -5, 10, 2, 0, -1, 3, 4, 2]);
```

```
A :=  $\begin{bmatrix} 1 & -5 & 10 \\ 2 & 0 & -1 \\ 3 & 4 & 2 \end{bmatrix}$ 
```

```
> inverse(A);
```

```
 $\begin{bmatrix} \frac{4}{119} & \frac{50}{119} & \frac{5}{119} \\ \frac{-1}{17} & \frac{-4}{17} & \frac{3}{17} \\ \frac{8}{119} & \frac{-19}{119} & \frac{10}{119} \end{bmatrix}$ 
```

```
> charpoly(A, x);
```

```
 $x^3 - 3x^2 - 14x - 119$ 
```

# 11 Grafik

<code>plot(f,x=a..b);</code>	ritar en graf över $f$ i intervallet $a..b$
<code>plot({f,g},x=a..b);</code>	ritar funktionerna $f$ och $g$ i samma bild
<code>plot3d(f,x=a..b,y=c..d);</code>	gör en tredimensionell graf över $f$

## 11.1 plot

Plot-kommandot ritat upp en graf över en funktion av en variabel. Man anger funktionsuttrycket som första parameter och önskat intervall för variabeln som andra parameter.

```
>plot(5*x*exp(sin(x)),x=-15..20);
```

Som tredje parameter kan man ange vilket intervall på y-axeln som ska ritas upp.

```
>plot(5*x*exp(sin(x)),x=-15..20,0..200);
```

Vill man rita flera grafer i samma bild måste man samla uttrycken i en mängd som man skickar med som första parameter till `plot`:

```
>plot({cos(x)^2,sin(x^2),sin(x)},x=-2..2);
```

Man kan också rita en parametriserad kurva genom att som enda parameter till `plot` skicka med en lista innehållande de två funktionerna samt parametrarnas utsträckning:

```
>plot([sin(t),cos(t),t=-0..2*Pi]);
```

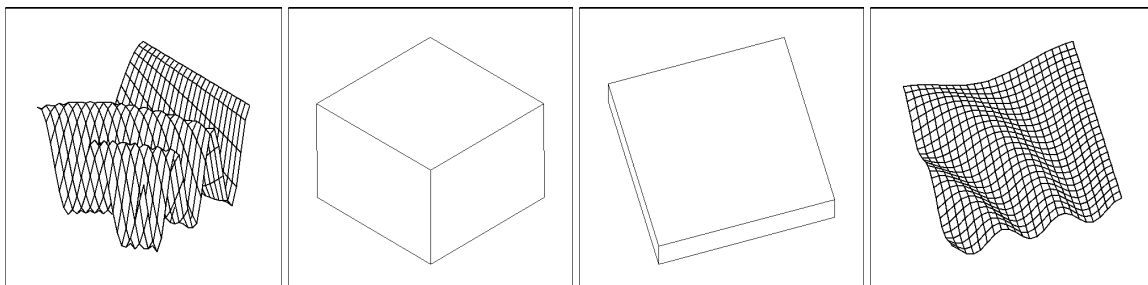
Genom att klicka med höger musknapp (ALT+musknapp på en enkelknappsmusad Mac) i bilden får man fram en meny som styr vissa parametrar, t ex linjestil och axlar.

## 11.2 plot3d

För att rita tredimensionella kurvor används `plot3d`.

```
>plot3d(cos(3*x*y),x=0..2,y=1..3);
```

Om synvinkeln inte passar är den lätt att ändra: klicka med vänsterknappen i bilden så framträder ett rätblock som visar hur grafen ligger. Justera till önskad synvinkel (fortfarande med vänsterknappen) och dubbelklicka slutligen i bilden så ritas grafen upp igen. Se bildserien nedan. I Windows behöver man bara trycka ner den vänstra musknappen, dra något i musen och släppa upp knappen.



## 11.3 plots-paketet

Med kommandot `with(plots)` laddas plots-paketet. Det innehåller bland annat plotfunktioner för polära, sfäriska och cylindriska koordinater samt animering. Skriv `?plots` för mer information.

## 11.4 Automatisk grafitning

I Windows och på Macintosh kan man rita en graf på ett enkelt sätt: Man markerar ett svarsuttryck var som helst, till exempel  $\sin(x)$ . Sedan trycker man på den högra musknappen (på Macintosh håller man ned alt-tangenten och trycker på musknappen) så att en liten meny kommer fram. I den väljer man **Plots** och **2D-Plot**. Då ritar Maple en sinuskurva och väljer själv lämpligt intervall. Om man markerar ett svarsuttryck som beror av två variabler, till exempel  $\sin(x+y)$ , så väljer man **Plots** och **3D-Plot** i den lilla menyn. Då får man välja mellan **x,y** och **y,x**, det vill säga mellan höger- och vänstersystem. I Unix finns ännu inte automatisk grafitning.



## 12 Programmering

Maple är inte bara ett interaktivt algebrasystem utan även ett programmeringsspråk. De flesta av Maples matematiska funktioner är faktiskt skrivna i Maple och inte i C eller Lisp. Här följer en mycket kort beskrivning av hur man programmerar i Maple. Mer information finns i Maples hjälpsystem, i Maples referensmanual och i häftet "Programming in Maple: The Basics", se *Litteraturlista*, sidan 4.

### 12.1 Villkorssatser och slingor

En if-sats i Maple har följande syntax:

```
if uttryck then satser
  [elif uttryck then satser]
  [else satser]
fi
```

De avsnitt som omges av klamrar kan utelämnas och `elif`-delen får upprepas vid behov.

Exempel:

```
>if (a>b) or (a>100) then a:=b; b:=0;
>elif b=100 then a:=0;
>else a:=0; b:=0;
>fi;
```

Det finns två varianter av Maples `for`-slinga. Första varianten ser ut så här:

```
[for namn] [from uttryck] [by uttryck] [to uttryck] [while uttryck]
do satser od
```

Om [`for namn`] utelämnas används en skräpvariabel, om [`from uttryck`] utelämnas antas uttrycket vara 1, om [`by uttryck`] utelämnas antas uttrycket vara 1, om [`to uttryck`] utelämnas antas uttrycket vara  $\infty$  och om [`while uttryck`] utelämnas antas uttrycket vara true.

Exempel:

```
>for i from 2 by 5 to 47 while i^2-i<100 do evalf(sqrt(i)) od;
      1.414213562
      2.645751311
```

Observera att den styrande variabeln i slingan ovan inte är lokal för slingan; om man använder `i` efter slingan ser man att den har värdet 12. För att återge variabeln i dess symbolvärde skriver man

```
>i:='i';
```

Andra varianten är kortare:

```
[for namn] [in uttryck] [while uttryck] do satser od
```

Uttrycket efter `in` kan vara datastruktur (t ex en sekvens eller lista) eller ett uttryck (t ex en summa).

Exempel (utan `while`-del):

```
>for k in ln(x),log[2](x),log[10](x) do diff(k,x); od;
```

En ren `while`-slinga kan se ut så här:

```
>while solve(f,x)<>b do f:=f/2+b; od;
```

## 12.2 Funktioner

```
procnamn:=proc(parameterlista)
  [local variabellista];
  [options väljarlista];
  satser
end
```

Efter `local` deklaras lokala variabler och efter `options` anges väljare av olika slag (se nedan). Man kan returnera ett värde i en `RETURN`-sats, men om ingen sådan påträffas är det det sist beräknade värdet som returneras.

Exempel:

```
deriv := proc(uttryck)
  local i,n,u;
  u := uttryck;
  n := readstat("Antal deriveringar: ");
  for i to n do u := diff(u,x) od;
  RETURN(u);
end;
```

Satsen `readstat` används när man vill att användaren ska skriva in ett värde. Ledtexten måste omslutas av citattecken eller bakåtapostrofer. När man kör funktionen ovan ser det ut så här:

```
>deriv(x^5+x^2);
  Antal deriveringar: 3;
                        60x2
```

Observera semikolonet efter trean i inmatningen (inmatningen är ett vanligt Mapleuttryck).

## 12.3 Rekursion

Det går utmärkt att skriva rekursiva funktioner i Maple. Här är ett exempel på en rekursiv funktion som beräknar fibonaccital ( $F_n = F_{n-1} + F_{n-2}$ ,  $F_1 = 1$ ,  $F_2 = 1$ )

```
F:=proc(n)
  options remember;
  if n<=2 then RETURN(1) else RETURN(F(n-1)+F(n-2)); fi;
end;
```

Väljaren `remember` gör att alla funktionsvärden som beräknas sparas i en intern tabell. När funktionen nästa gång anropas med samma parameter (eller parametrar) utförs inte beräkningen utan det sparade värdet returneras direkt. Detta kan spara mycket tid. Prova gärna att räkna ut  $F(20)$  med och utan `options remember`.

När man löser problem med dynamisk programmering behöver man spara värden i en tabell. Eftersom detta sker automatiskt med `options remember` blir sådana problem enkla att lösa – man skriver bara den rekursiva definitionen som en funktion.

## 12.4 Lokala och globala variabler

Om man deklarerar en funktion inuti en annan funktion övertar den inre funktionen de variabler som deklarerats lokalt i den yttre funktionen. (Detta är ett nytt beteende i release 5.) I funktionen `test1` nedan är `b` en lokal variabel som ges värdet 2. Det `b`-värde som `test2` returnerar är däremot den globala variabeln `b` (som inte har fått något värde).

```
> test1:=proc()
>   local b;
>   test2:=proc()
>     local l; l:=0;
>     global g; g:=1
>     RETURN(b);
>   end;
>   b:=2;
>   RETURN(l,g,test2());
> end;
```

En körning av funktionen ger följande resultat:

```
>test1();
      l, 1, 2
```

## 12.5 Utmatning

Normalt sett skriver Maple ut varje beräknat uttryck. Utmatning kan förhindras genom att man skriver kolon efter uttrycket istället för semikolon. Detta kan vara användbart i t ex `for`-slingor där man inte är intresserad av resultaten i varje varv.

I funktioner (dvs mellan `proc` och `end`) sker däremot ingen utmatning. Vill man skriva ut ett värde i en funktion använder man kommandot `print`.

Exempel:

```
print(x);
```

Flera värden på samma rad kan skrivas ut med `lprint` eller `printf`.

## 12.6 Kommentarer

Efter kommentartecknet `#` tolkas resten av raden som en kommentar.

## 12.7 Inläsning av Maplefiler

Ibland är det enklare att skriva in sina Maplesatser med Emacs eller någon annan texteditor i en fil och sedan läsa in filen i Maple. Filen behöver inte inledas eller avslutas på något speciellt sätt; man skriver bara in satserna precis som man skulle göra om man körde Maple interaktivt. När man sparar filen och vill läsa in den i Maple skriver man

```
>read testfil;
```

så utförs alla satserna i filen `testfil`.

Om filnamnet innehåller tecken med speciell betydelse måste det omslutas av citattecken eller bakåtapostrofer.

```
>read "testfil.txt";
```

## 12.8 Avlusning och felsökning

Avlusningskommandon (brytpunkter, spårutskrifter mm) finns inbyggda i Maple. Skriv `?debugger` för en fullständig beskrivning.

Programmet `mint` kontrollerar syntaxen i ett Mapleprogram. På Mac och under Windows startas programmet genom att man dubbelklickar på ikonen som heter Mint. Där kan man sedan välja vilken fil man vill undersöka.

Under X och på textterminal anropar man `mint` direkt från Maple:

```
>system("mint testfil");
```

Detta kommando returnerar 0 om syntaxen är korrekt och ett annat värde annars. Observera att utmatningen från kommandot kommer i terminalfönstret även när man kör `xmaple`.

Maplevariabeln `printlevel` styr mängden utskrifter såväl i egna funktioner som i Maples funktioner. Normalt är den satt till 1, vilket betyder att inga utskrifter görs i funktioner. Skriv `printlevel:=1000;` om du vill att utskrifter alltid ska göras i alla funktioner.

Vid felsökning i en viss funktion är kommandot `trace` mer praktiskt. `trace(f);` gör att information (text parametrarnas värden och returvärdet) kommer att skrivas ut varje gång `f` anropas i fortsättningen. Det stängs av igen med `untrace(f);`. Man får samma effekt genom att lägga till `options trace;` i funktionen.

## 13 Att spara på fil och skriva ut

### 13.1 Spara på Mapleformat

Om man vill spara allt man gjort i Maple för att kunna fortsätta en annan dag sparar man lämpligen i en fil på Mapleformat. Välj **Save** i **File**-menyn så öppnas en ruta där man väljer filnamn. Normalt väljer man ett filnamn med suffixet `mws`, t ex `testfil.mws`.

För att läsa in filen i Maple igen väljer man **Open** i **File**-menyn.

Observera att en fil på Mapleformat bara kan läsas av Maple; om filen ska användas i något annat program kan det vara lämpligt att spara på textformat, se nedan. Det är inte heller tillrådligt att försöka skriva ut en fil sparad på Mapleformat.

På textterminal finns förstås inga menyer, så då sparar man på följande sätt:

```
>save "testfil.m";
```

Filnamnet måste omslutas av citattecken eller bakåtapostrofer eftersom det innehåller en punkt. Det är viktigt att filnamnets suffix är just `.m` och ingenting annat.

För att läsa in filen i Maple igen skriver man

```
>read "testfil.m";
```

### 13.2 Spara på textformat

Välj **Export as** och **Plain text...** i **File**-menyn för att spara som en textfil. Det är bra om man t ex vill föra över data till något annat program (som Matlab). Filen kan läsas in igen med **Open** i **File**-menyn.

Vill man spara i HTML-format kan man välja **Export as** och **HTML...** i **File**-menyn.

Maple kan också generera LaTeX-, C- och FORTRAN-kod. Skriv `?latex`, `?C` respektive `?fortran` för att få reda på hur man gör.

### 13.3 Utskrift

Alternativet **Print** i **File**-menyn skriver ut Mapledokumentet på skrivaren. Under X kan man sedan behöva klicka på knappen **Print Command** för att fönstret ska skrivas ut direkt på skrivaren och inte på en fil.

## 14 Hjälp

### 14.1 Hjälp om Maplekommandon

Det enklaste sättet att få hjälp om ett visst Maplekommando är att skriva ett frågetecken, följt av kommandots namn. För att få hjälp om plot-kommandot skriver man:

```
>?plot
```

Härvid öppnas ett hjälpfönster (på textterminal skrivs hjälpen ut direkt på skärmen) med information om det aktuella kommandot. Ett ensamt frågetecken ger allmän information. Hjälptexten innehåller alltid exempel på användning av kommandot. Vill man bara ha ut dessa exempel skriver man tre frågetecken, t ex `??plot`

Om man inte vet namnet på det kommando man söker kan man klicka på **Introduction** i **Help**-menyn (längst uppe till höger i Maplefönstret) så får man ett hjälpfönster med ämnesindex att söka i.

För att söka efter ett kommando kan man använda **Topic Search** eller **Full Text Search** i **Help**-menyn.

Om du vill se vilket kommando en viss symbol motsvarar kan du välja **Balloon Help** (på Macintosh: **Show balloons**) i menyn **Help** och placera musmarkören på aktuell symbol (utan att klicka på den).

### 14.2 Vanliga misstag

Det allra vanligaste felet man gör i Maple är att glömma semikolonet efter kommandot man skrivit in. Maple skriver då ut `warning, premature end of input`. Om man nu skriver ett semikolon och trycker på retur/enter så utförs kommandot.

Den som är van att köra Matlab eller programmera i C gör lätt fel i tilldelningar. Om man skriver `x=5` när man menar `x:=5` kommer inte Maple att protestera, men `x` kommer inte att få värdet 5.

En variabel som har fått ett värde kan inte användas i vissa Maplefunktioner (t ex `int` och `diff`) där Maple förväntar sig en symbol. Om man har satt en variabel till ett värde och vill göra den obestämd igen skriver man en tilldelningssats med variabeln själv i vänsterledet och variabeln inom apostrofer i högerledet:

```
>x:= 'x';
```

Det kan vara knepigt att hitta rätt apostrofer på tangentbordet. Framåtapostrofer hindrar direkt evaluering medan bakåtapostrofer eller citattecken används omkring textsträngar.

De inbyggda konstanterna i Maple är det lätt att göra fel med. Skriver man `pi` i ett uttryck skrivs symbolen  $\pi$  på skärmen, precis på samma sätt som om man skriver `Pi`. Men det är bara `Pi` som Maple känner igen. Prova t ex att beräkna `sin(pi)` och `sin(Pi)`.

Om en plot-bild blir tom kan det bero på att man använt `plot` istället för `plot3d` för att rita en yta eller definierat intervallet fel i `plot`-kommandot.

Skriv inte `f(x):=x+3;` eller `f(x):=x->x+3;` utan `f:=x->x+3;` för att definiera en funktion. Se *Egendefinierade funktioner*, sidan 12.

### 14.3 Fel i Maple

Många av de fel som fanns i Maple V release 4 är nu rättade, men det kan finnas nya, ännu oupptäckta fel i Maple V release 5. Kontrollera därför alltid Maples resultat. Se inledningen, sida 4.

# Bihang A Laborationer

## Lab 1

Starta Maple enligt anvisningarna för din dator, se kapitel 2.

På skärmen ser du tecknet `>` längst till vänster. Här skriver du dina Maplekommandon. Skriv in följande kommandon (observera att Maplekommandon alltid avslutas med semikolon) och notera Maples utmatning efter varje kommando.

Först några numeriska beräkningar. Maple räknar exakt med bråk och förenklar automatiskt.

```
>5/13+12/27;
```

Resultat kan också lagras i variabler.

```
>a:=7/17;
```

```
>3*a+4;
```

Beräkna nu  $4!^{4!}$ .

```
>4!^4!;
```

För att få reda på hur  $\pi$  ser ut med hundra siffror kan man använda kommandot `evalf` som ger en flyttalsapproximation med önskad noggrannhet. Ta först reda på hur `evalf` fungerar med hjälpkommandot frågetecken.

```
>?evalf
```

```
>evalf(Pi,100);
```

I symboliska beräkningar kan man räkna med obekanta variabler, som  $x$  här:

```
>u:=exp(x)-2*x^3-5;
```

Nu har variabeln  $u$  tilldelats ett värde. I varje beräkning där  $u$  ingår kommer  $u$  att evalueras till detta värde.

```
>2*u;
```

Däremot har inte  $x$  tilldelats något värde. Om variabeln  $x$  förekommer i ett uttryck evalueras den till symbolen  $x$ . I ett traditionellt programmeringsspråk som t ex Pascal skulle det vara fel att använda en variabel som inte har fått ett värde, men i Maple vill man ofta räkna symboliskt; man vill ha symboliska uttryck (formler) istället för tal som svar.

Om du nu ger  $x$  ett värde så kommer alla uttryck som innehåller  $x$  att förändras.

```
>x:=2;
```

```
>2*u;
```

---

Nu ska du prova derivering.

```
>diff(sin(t),t);
```

---

```
>diff(u,x);
```

---

Att det blir fel här beror på att  $x$  evalueras till värdet 2, och man kan ju inte derivera med avseende på 2. En speciell tilldelningssats krävs för att återge variabeln  $x$  dess symbolvärde. I högerledet står variabelns namn omgivet av vanliga apostrofer.

```
>x:='x';
```

Det finns flera apostrofer på tangentbordet. Om du skrivit rätt kommer Maple att svara med  $x:=x$ .

Nu har  $x$  blivit symbolen  $x$  igen. Gör ett nytt försök att derivera. Lagra resultatet i  $d$ .

```
>d:=diff(u,x);
```

---

Utnyttja det för att beräkna andraderivatan.

```
>d2:=diff(d,x);
```

---

Andraderivatan som beräknades ovan sparades i variabeln  $d2$  så att man ska kunna använda den senare utan att behöva skriva om kommandot.

Uttrycket  $u$  ska integreras också. Om du vill veta mer om `int` kan du anropa hjälpfunktionen med kommandot `?int`. Skriv sedan

```
>int(u,x);
```

---

Hur reagerar Maple inför något som inte går att integrera? Försök att integrera  $\frac{1}{u}$

```
>int(1/u,x);
```

---

Det är lätt att rita en graf över  $u$ .

```
>plot(u, x=-3..3);
```

Vill man rita upp två uttryck i samma graf sätter man klamrar runt dem.

```
>plot({u,d2}, x=-3..3);
```

Både  $u$  och  $d2$  är uttryck, men man kan också definiera funktioner i Maple.

```
>f:=x->exp(x)-2*x^3-5;
```

---



Man kan nu t ex beräkna funktionens värde i punkten  $x=0$ .

```
>f(0);
```

Ekvationslösningskommandot `solve` tar två parametrar, ekvationen som ska lösas och variabeln den ska lösas med avseende på. Prova att lösa en andragradsekvation med `solve`.

```
>solve(x^2-5*x-7=0,x);
```

Med `evalf` kan du få numeriska värden på rötterna.

```
>evalf(%);
```

Procenttecken betecknar alltså det senast beräknade resultatet. Det kan vara praktiskt ibland, men är samtidigt farligt: Om går upp med musen och utför ett kommando som står tidigare i huvudfönstret blir resultatet av detta det senast beräknade. Procenttecken betecknar alltså inte alltid resultatet på raden ovanför.

Rita nu upp grafen för funktionen i ett lämpligt intervall, t ex  $[-5..10]$ , så att bägge rötterna syns.

```
>plot(x^2-5*x-7,x=-5..10);
```

Eftersom Maple räknar symboliskt är det lika enkelt att lösa en generell andragradsekvation.

```
>solve(x^2+p*x+m=0,x);
```

Rita nu upp grafen för följande fjärdegradspolynom

```
>plot(x^4-12*x^2+16,x=-5..5);
```

Intervall i y-led kan också justeras

```
>plot(x^4-12*x^2+16,x=-5..5,y=-20..20);
```

Lös ekvationen.

```
>solve(x^4-12*x^2+16=0,x);
```

Som du ser ger `solve` fyra rötter som svar. Lagrar man svaret i en variabel så kan man med hakparenteser plocka ut ett värde i taget.

```
>s:=solve(x^4-12*x^2+16=0,x);
```

```
>s[1]; s[2]; s[3]; s[4];
```

Multiplitera ihop rötterna

```
>s[1]*s[2]*s[3]*s[4];
```

och utför multiplikationerna. Resultatet bör bli konstantermen i fjärdegradspolynomet.

```
>expand(%);
```

-----

Om man vill lösa ett ekvationssystem måste både systemet och variabelistan omslutas av klamrar.

```
>solve({5*x+y=2,x-y=1/3},{x,y});
```

-----

Om ekvationerna innehåller decimaltal görs lösningen om till decimaltalsform.

```
>solve({x+y=3.2,x/7-z=0,x+y+z=sqrt(5)},{x,y,z});
```

-----

En tredimensionell yta i rummet kan ritas upp med `plot3d`.

```
>plot3d(y^4-x^2,x=-4..4,y=-4..4);
```

Prova att vrida på grafen, se kapitel 11, sidan 23 i häftet för instruktioner.

Nu är det dags att spara allt du gjort hittills, och det kan du göra på tre olika sätt:

- Välj **Save** i **File**-menyn och ge filen namnet `lab1.mws` så sparas alla kommandon du skrivit.  
Filen `lab1.mws` sparas på Mapleformat, som bara Maple kan läsa. Försök *inte* skriva ut filen på skrivaren. Nästa gång du kör Maple kan du läsa in filen genom att välja **Open** i **File**-menyn.
- Spara som en textfil genom att välja **Export as Text** i **File**-menyn och ge filen namnet `lab1.txt`. Nu är alla kommandon och resultat sparade som text. Filen kan läsas in igen med **Open** i **File**-menyn.
- Om skrivaren är ledig kan du prova att skriva ut ditt arbetsblad på papper genom att välja alternativet **Print** i **File**-menyn.

Avsluta Maple genom att välja nedersta alternativet i **File**-menyn (**Exit/Quit**) eller på textterminal genom att skriva

```
>quit;
```

## Lab 2

### Hjälpssystemet

Det finns flera olika sätt att få hjälp i Maple. Klicka på **Help** så får du en meny över tillgängliga hjälpfunktioner. **Introduction** låter dig se vilka Maple-kommandon som finns inom olika områden. I **Topic Search** ger du ett nyckelord och får tillbaka en lista över funktioner som har med nyckelordet att göra. Du kan också få hjälp om ett visst kommando genom att direkt i Maple-fönstret skriva `?kommando`, `tex ?int` för att få hjälp om integrationskommandot.

- Vad heter kommandot för plottning i polära koordinater?
- Vilka aritmetiska operatorer finns i Maple?
- Hur beräknar man det hundra primtalet?
- Finns det några funktioner som skulle vara intressanta för en bankman?
- Hur använder man kommandot `subs`?

### Enkla beräkningar

Maple ger normalt ett exakt svar, men du kan alltid få en flyttalsapproximation genom att använda kommandot `evalf`. Konstanterna  $i$  och  $\pi$  skrivs `I` och `Pi` i Maple (observera versalerna). Glöm inte semikolon efter varje sats (kolon kan användas om du inte vill se resultatet).

- Beräkna följande uttryck:  $\cos\frac{\pi}{4} + \sin\frac{5\pi}{6}$  och  $(2i+3)^2 - \frac{4i+5}{6-i}$
- Vad blir  $\sqrt{2\sqrt{19549}+286} - \sqrt{173}$  ?
- Vad är differensen mellan  $\frac{1}{3} + \frac{1}{3} + \frac{1}{3}$  och  $\frac{1.0}{3.0} + \frac{1.0}{3.0} + \frac{1.0}{3.0}$  i Maple?
- Ge kommandot `Digits:=15` och räkna ut ovanstående skillnad igen.
- Använd `evalf` för att räkna ut  $\pi$  med tre, tio och hundra decimaler.
- Vilken av kvoterna  $\frac{19}{6}$ ,  $\frac{22}{7}$  och  $\frac{25}{8}$  är en hyfsad approximation till  $\pi$ ?

### Variabler, uttryck och funktioner

Om du givit variabeln `a` ett värde kan den återges sitt symbolvärde med kommandot `a:= 'a'`; Kommandot `anames()`; visar vilka variabler som tilldelats värden.

- Ge variabeln `a` värdet  $\frac{4}{17}$  och beräkna sedan  $\left(a + \frac{9}{34}\right)^2$
- Skriv in följande satser: `b=7;` `3*b;` Varför blir inte resultatet 21?
- Definiera funktionen `f` som `sin(x)cos(x)` och beräkna sedan `f(x+beta)` och `f(pi-x)`.
- Vad är felet i följande funktionsdefinition: `g(x):=x+5;` ? Försök att beräkna `g(3)`.
- Sätt `p` till uttrycket  $e^{x^2}$  och använd `subs` för att beräkna `p`'s värde i `x=5`.
- Använd `unapply` för att omvandla uttrycket `p` till en funktion `pf`. Beräkna sedan `pf(5)`.
- Definiera funktionen  $f(x,y) = \frac{x^3y - xy^3}{x^2 + y^2}$  dels med `->` och dels med `unapply`.

## Förenklingar

- a. Utveckla, faktorisera och förenkla uttrycket  $\frac{x^4 - 6x^3 + 13x^2 - 12x + 4}{x^3 - 6x^2 + 11x - 6}$
- b. Försök att ur polynomet  $(x+1)(x+y)^2$  få fram följande variationer:  
 $x^3 + 2x^2y + xy^2 + x^2 + 2xy + y^2$ ,  $x^3 + (1+2y)x^2 + (2y+y^2)x + y^2$ ,  $(x+1)y^2 + 2(x+1)xy + (x+1)x^2$
- c. Förenkla uttrycket  $\cos(2x+y) + 2\cos x \sin x \sin y + \cos y$ .

## Derivator och integraler

- a. Derivera  $\frac{x \sin x}{(x+2)^3}$ .
- b. Beräkna integralen  $\int e^{-x} \sin x dx$
- c. Beräkna  $\int_0^{\infty} e^{-x^2} (x-3)^6 dx$

## Ekvationslösning

- a. Lös följande ekvation med `solve` och `fsolve`  $48x^5 + 8x^4 - 6x^3 + 114x^2 - 37x + 18 = 0$ .
- b. Lös ekvationen  $x^5 = x - 1$ .
- c. Lös differentialekvationen  $\frac{d}{dx}y = \sqrt{y+1}$ .
- d. Lös ekvationssystemet  $x - y = 5$   
 $xy = y^2 - 2$
- e. Lös olikheterna  $1 + x^2 < 17$  och  $\{1 + x^2 < 17, x > 0\}$  med `solve`.

## Grafik

Speciella grafiska funktioner som `polarplot` och `animate` finns i `plots`-paketet som laddas med kommandot `with(plots)`;

- a. Rita upp kurvan  $x + \cos \pi x$  på intervallet  $x = -5 \dots 5$ . Prova alla menyalternativ under **Axes** i grafikfönstret.
- b. Rita i samma diagram upp kurvorna  $\ln x$ ,  $x$ ,  $x^2$  och  $e^x$ . Justera intervallängden i y-led så att alla kurvor syns bra.
- c. Rita upp  $\frac{x^3 y - x y^3}{x^2 + y^2}$  i ett symmetriskt intervall kring origo. Roterat ytan med musen.
- d. Rita med `polarplot` upp treklöverkurvan  $(4 \sin^2 t - 1) \cos t$  för  $t = -\pi \dots \pi$
- e. Använd `animate` för att göra en animerad graf av  $\sin(x+k)$  för  $k$  mellan 0 och  $\pi$ . Prova sedan alla alternativ i Animation-menyn (som kommer fram när du klickar i bilden).
- f. Använd `animate3d` för att göra en animerad graf av  $\sin(x+k) \cos(y+k)$ .

## Maples begränsningar

- a. Beräkna  $2^{19} - 9$  och lagra värdet i variabeln  $n$ . Använd sedan kommandot `length` för att undersöka hur många siffror talen  $10^n$  och  $10^{n+1}$  har.

## Bihang B Övningsuppgifter

- Beräkna  $\frac{1}{12} + \frac{11}{13} + \frac{1}{14}$
- Verifiera att  $95800^4 + 217519^4 + 414560^4 = 422481^4$
- Dela upp polynomet  $x^4 - 2x^3 - 13x^2 + 14x + 24$  i faktorer.
- Förenkla uttrycket  $\frac{1 + 2 \tan x - \tan^2 x}{\cos 2x + \sin 2x}$
- Beräkna första- och andraderivatan av  $f(x) = \ln \sqrt{\sin x \cos x}$ .

- Beräkna följande integraler:

$$\int (\tan x + \cot x)^2 dx \qquad \int_0^{\frac{\pi}{2}} x \cos x dx \qquad \int_0^{\infty} e^{-t^2} \cos at dt$$

- Beräkna följande summor:

$$\sum_{n=2}^9 \frac{1}{n(n-1)} \qquad \sum_{k=0}^{1000} 2k+1 \qquad \sum_{k=0}^{\infty} (-x)^k$$

- Lös ekvationen  $a + \ln(x-3) - \ln x = 0$ .
- Bestäm rötterna till ekvationen  $17^{(2x-1)} + 17^{(x+1)} = 9826$ . Kan rötterna förenklas?
- Lös ekvationssystemet nedan exakt och numeriskt.

$$\begin{aligned} \sqrt{3}x + 3y &= 1 \\ 3x + \pi z &= 9 \\ \pi y - z &= 0 \end{aligned}$$

- Lös följande differentialekvation med begynnelsevillkor:

$$\begin{aligned} y'' + y' + y &= 0 \\ y(0) &= 1 \\ y'(0) &= 1 \end{aligned}$$

- Serietveckla  $\sin(e^{\pi-t^2})$  kring  $t=0$ .
- Vad blir  $\lim_{x \rightarrow 0} (1+x)^{1/x}$ ?
- Skapa en mängd S som innehåller de femtio första udda talen, en mängd T med de trettiofyra första multiplerna av tre och beräkna sedan snittet mellan S och T.

15. Rita en graf över funktionen  $(x+y)^3 + \cos(3xy)$  för  $-4 \leq x \leq 4$  och  $-4 \leq y \leq 4$ . Byt synvinkel några gånger för att se hur grafen verkligen ser ut.

16. Rita en graf över kurvorna  $x^3 - x^2 - 1$ ,  $x^3 - 2x^2 - 4$ , ...,  $x^3 - 10x^2 - 100x$  dvs kurvscharan  $x^3 - kx^2 - k^2$  för  $k = 1 \dots 10$ . Lämpligt intervall är  $x = -5 \dots 5$ .  
Ledning: använd `seq`.

17. När man löser stora linjära ekvationssystem med `solve` får man som svar en mängd innehållande variablerna och deras värden. Exempel:

```
>solve( { 3*x+5*y+z, 5*z+3/2*y+3, 8*x-y-22 }, { x, y, z } );
```

$$\left\{ z = -\frac{30}{203}, x = \frac{520}{203}, y = -\frac{306}{203} \right\}$$

Eftersom variablerna i lösningen inte är ordnade på samma sätt som i anropet av `solve` kan man inte direkt plocka fram en viss variabls värde i lösningen. Skriv en funktion som tar en sådan mängd och en variabel som parametrar och returnerar variabelns värde i lösningen. Om variabeln inte finns med i mängden ska funktionen returnera ett meddelande om detta.

# Lösningar till övningsuppgifterna

1. `1/12+11/13+1/14;`
2. `95800^4+217519^4+414560^4-22481^4;`
3. `factor(x^4-2*x^3-13*x^2+14*x+24);`
4. `(1+2*tan(x)-(tan(x))^2)/(cos(2*x)+sin(2*x));  
simplify(%);`
5. `diff(ln(sqrt(sin(x)*cos(x))),x);  
diff(ln(sqrt(sin(x)*cos(x))),x,x);`
6. `int((tan(x)+cot(x))^2,x);  
int(x*cos(x),x=0..Pi/2);  
int(exp(-t^2)*cos(a*t),t=0..infinity);`
7. `sum(1/(n*(n-1)),n=2..9);  
sum(2*k+1,k=0..1000);  
sum((-x)^k,k=0..infinity);`
8. `solve(a+ln(x-3)-ln(x),x);`
9. `res:=solve(17^(2*x-1)+17^(x+1)-9826,x);  
simplify(res[1]);  
simplify(res[2]);`
10. `solve({sqrt(3)*x+3*y=1,3*x+Pi*z=9,Pi*y-z=0},{x,y,z});  
fsolve({sqrt(3)*x+3*y=1,3*x+Pi*z=9,Pi*y-z=0},{x,y,z});`
11. `dsolve({diff(y(x),x,x)+diff(y(x),x)+y(x),y(0)=1,D(y)(0)=1},y(x))  
;  
eller  
dsolve({D(D(y))(x)+D(y)(x)+y,y(0)=1,D(y)(0)=1},y(x));`
12. `series(sin(t*exp(Pi-t^2)),t=0);`
13. `limit((1+x)^(1/x),x=0);`
14. `S:={seq(2*k-1,k=1..50)};  
T:={seq(3*k,k=1..33)};  
S intersect T;`

15. `plot3d((x+y)^3+cos(3*x*y), x=-4..4, y=-4..4);`

16. `plot({seq(x^3-k*x^2-k^2, k=1..10)}, x=-5..5);`

17. `plocka:=proc(s,x)
 local e;
 for e in s do if lhs(e) = x then RETURN(rhs(e)) fi od;
 RETURN('variabeln förekommer inte');
end;`

Exempel:

`s:=solve({3*x+5*y+z, 5*z+3/2*y+3, 8*x-y-22}, {x,y,z});`

$$\left\{ z = -\frac{30}{203}, x = \frac{520}{203}, y = -\frac{306}{203} \right\}$$

`plocka(s,x);`

$$\frac{520}{203}$$



## Bihang C Operatörer och reserverade ord

Operatorernas prioritetsordning, från högsta prioritet till lägsta:

Tabell 1 *Operatorernas prioritetsordning*

1	%
2	&-operatorer
3	!
4	^, **, @@
5	*, &*, /, @, intersect
6	+, -, union, minus
7	mod
8	..
9	<, <=, >, >=, =, <>
10	\$
11	not
12	and
13	or
14	->
15	,
16	:=

Observera att exponentieringsoperatorerna  $\wedge$ ,  $**$ , och  $@@$  är ickeassociativa och därför är uttrycket  $a^b^c$  otillåtet i Maple (man måste använda parenteser).

### Reserverade ord

Följande trettio ord är reserverade i Maple:

and	by	do	done	elif
else	end	fi	for	from
if	in	intersect	local	minus
mod	not	od	option	options
or	proc	quit	read	save
stop	then	to	union	while

Om du måste använda något av dessa ord som ett namn så måste ordet omslutas av bakåtapostrofer.

# Bihang D Funktioner

## Funktioner

Maple känner till följande matematiska funktioner, i den mening att förenklingsprocedurer finns definierade för dem och/eller att de är kända för en eller flera av följande: diff, evalc, evalf, expand, series, simplify.

<b>abs</b>	absolute value of real or complex argument
<b>argument</b>	argument of a complex number or expression
<b>AiryAi, AiryBi</b>	Airy wave functions
<b>AngerJ</b>	Anger J function
<b>arccos</b>	arccos
<b>arccosh</b>	arccosh
<b>arccot</b>	arccot
<b>arccoth</b>	arccoth
<b>arccsc</b>	$\text{arccsc}(x) = \arcsin(1/x)$
<b>arccsch</b>	$\text{arccsch}(x) = \text{arcsinh}(1/x)$
<b>arcsec</b>	$\text{arcsec}(x) = \arccos(1/x)$
<b>arcsech</b>	$\text{arcsech}(x) = \text{arccosh}(1/x)$
<b>arcsin</b>	arcsin
<b>arcsinh</b>	arcsinh
<b>arctan</b>	arctan
<b>arctan(y,x)</b>	two-argument arctan: $\text{arctan}(y,x) = \text{argument}(x+i*y)$ in $(-\text{Pi}, \text{Pi}]$
<b>arctanh</b>	arctanh
<b>bernoulli</b>	Bernoulli numbers and polynomials
<b>BesselI</b>	modified Bessel function of the first kind $I(v,x)$
<b>BesselJ</b>	Bessel function of the first kind $J(v,x)$
<b>BesselK</b>	modified Bessel function of the second kind $K(v,x)$
<b>BesselY</b>	Bessel function of the second kind $Y(v,x)$
<b>Beta</b>	Beta function: $\text{Beta}(x,y) = \text{GAMMA}(x) * \text{GAMMA}(y) / \text{GAMMA}(x+y)$
<b>binomial</b>	binomial coefficients: $\text{binomial}(n,r) = n! / (r! * (n-r)!)$
<b>ceil</b>	$\text{ceil}(x) =$ smallest integer greater than or equal to $x$
<b>Chi</b>	hyperbolic cosine integral
<b>Ci</b>	cosine integral
<b>conjugate</b>	conjugate of a complex number or expression
<b>cos</b>	cos
<b>cosh</b>	cosh
<b>cot</b>	cot
<b>coth</b>	coth
<b>csc</b>	$\text{csc}(x) = 1/\sin(x)$
<b>csch</b>	$\text{csch}(x) = 1/\sinh(x)$
<b>csgn</b>	complex "half-plane" signum function
<b>dawson</b>	Dawson's integral
<b>Dirac</b>	Dirac delta function
<b>dilog</b>	dilogarithm integral = $\text{int}(\ln(t)/(1-t), t=1..x)$
<b>Ei</b>	exponential integral = $\text{int}(\exp(t)/t, t=-\text{infinity}..x)$

<b>EllipticCE, EllipticCK, EllipticCPi, EllipticE, EllipticF, EllipticK, EllipticModulus, EllipticNome, EllipticPi</b>	Complete, incomplete and complementary elliptic integrals and related functions (former LegendreF, LegendreE etc).
<b>erf</b>	error function = $2/\sqrt{\pi} \int_0^x \exp(-t^2) dt$
<b>erfc</b>	complementary error function = $1 - \text{erf}(x)$ ; iterated integrals
<b>euler</b>	Euler numbers and polynomials
<b>exp</b>	the exponential function: $\exp(x) = \sum_{i=0}^{\infty} x^i/i!$
<b>factorial</b>	the factorial function $\text{factorial}(n) = n!$
<b>floor</b>	$\text{floor}(x) =$ greatest integer less than or equal to $x$
<b>frac</b>	fractional part of a number
<b>FresnelC</b>	Fresnel cosine integral = $\int_0^x \cos(\pi/2 t^2) dt$
<b>FresnelF</b>	auxiliary function = $(1/2 - \text{FresnelS}(z)) \cos(\pi/2 z^2) - (1/2 - \text{FresnelC}(z)) \sin(\pi/2 z^2)$
<b>FresnelG</b>	auxiliary function = $(1/2 - \text{FresnelC}(z)) \cos(\pi/2 z^2) + (1/2 - \text{FresnelS}(z)) \sin(\pi/2 z^2)$
<b>FresnelS</b>	Fresnel sine integral = $\int_0^x \sin(\pi/2 t^2) dt$
<b>GAMMA</b>	the Gamma function $\text{GAMMA}(z) = \int_0^{\infty} \exp(-t) t^{z-1} dt$ and the incomplete Gamma function $\text{GAMMA}(a, x) = \int_x^{\infty} \exp(-t) t^{a-1} dt$
<b>GaussAGM</b>	Gauss arithmetic geometric mean
<b>HankelH1, HankelH2</b>	Hankel functions (Bessel functions of the third kind)
<b>harmonic</b>	$\text{harmonic}(n) = \sum_{i=1}^n 1/i = \Psi(n+1) + \gamma$ (known to evalf)
<b>Heaviside</b>	Heaviside step function
<b>hypergeom</b>	the generalized hypergeometric function
<b>ilog10, ilog</b>	integer logarithms
<b>Im</b>	imaginary part of a complex number
<b>JacobiAM, JacobiCN, JacobiCD, JacobiCS, JacobiDN, JacobiDC, JacobiDS, JacobiNC, JacobiND, JacobiNS, JacobiSC, JacobiSD, JacobiSN</b>	Jacobi elliptic functions
<b>JacobiTheta1, JacobiTheta2, JacobiTheta3, JacobiTheta4,</b>	Jacobi theta functions
<b>JacobiZeta</b>	Jacobi Zeta function
<b>KelvinBer, KelvinBei, KelvinHer, KelvinHei, KelvinKer, KelvinKei,</b>	Kelvin functions
<b>LambertW</b>	the function satisfying $\text{LambertW}(x) \exp(\text{LambertW}(x)) = x$

<b>LegendreF</b> etc	Obsolete Maple names for elliptic integrals, see EllipticF etc
<b>Li</b>	logarithmic integral
<b>ln</b>	natural logarithm (logarithm with base $E = 2.71828\dots$ )
<b>lnGAMMA</b>	ln of the GAMMA function
<b>log</b>	logarithm to arbitrary base
<b>log10</b>	log to the base 10
<b>max, min</b>	maximum/minimum of a list of real values
<b>MeijerG</b>	special case of the Meijer G function. See also ?MeijerG.
<b>O</b>	used to specify the order term for series, e.g. $1+x+O(x^2)$
<b>pochhammer</b>	pochhammer function
<b>polar</b>	polar representation of complex numbers
<b>polylog</b>	polylogarithm function
<b>Psi</b>	polygamma function $\Psi(x) = \text{diff}(\ln(\text{GAMMA}(x)), x)$ and its derivatives $\Psi(n, x) = \text{diff}(\Psi(x), x\$n)$
<b>Re</b>	real part of a complex number
<b>RootOf</b>	function for expressing roots of algebraic expressions
<b>round</b>	$\text{round}(x) =$ nearest integer to $x$ ( $\text{round}(.5) = 1$ )
<b>sec</b>	$\text{sec}(x) = 1/\cos(x)$
<b>sech</b>	$\text{sech}(x) = 1/\cosh(x)$
<b>signum</b>	sign of a complex argument $\text{signum}(z) = z/\text{abs}(z)$
<b>Shi</b>	hyperbolic sine integral
<b>Si</b>	sine integral
<b>sin</b>	sin
<b>sinh</b>	sinh
<b>sqrt</b>	square root
<b>Ssi</b>	shifted sine integral
<b>StruveH, StruveL</b>	Struve functions
<b>surd</b>	non-principal root function
<b>tan</b>	tan
<b>tanh</b>	tanh
<b>trunc</b>	$\text{trunc}(x) =$ nearest integer from $x$ in the direction of 0
<b>WeberE</b>	Weber E function
<b>WeirstrassP</b>	Weirstrass P-function
<b>WeirstrassPPrime</b>	Derivative of Weirstrass P-function
<b>WeirstrassZeta</b>	Weirstrass zeta-function
<b>WeirstrassSigma</b>	Weirstrass sigma-function
<b>Zeta</b>	Riemann and Hurwitz zeta functions

Fler matematiska funktioner finns definierade i diverse paket, som t ex linalg-paketet och simplex-paketet. Använd kommandot ?packages för att få reda på vilka paket som finns tillgängliga.

Många Maple-kommandon finns i inaktiv (inert) version som kan vara användbar vid bland annat ekvationslösning. Namnet är detsamma, men med stor begynnelsebokstav. Jämför t ex  $\text{int}(x*\exp(x), x)$  och  $\text{Int}(x*\exp(x), x)$ . För att evaluera en inaktiv funktion används kommandot value.

# Bihang E Lathund

## Operatorer

$x+y$	addition
$x-y$	subtraktion
$x*y$	multiplikation
$x/y$	division
$x**y$	exponentiering
$x^y$	exponentiering
$x \bmod y$	modulo
$x!$	fakultet

## Konstanter

I	$\sqrt{-1}$
Pi	$\pi$
infinity	$\infty$
true	<i>sant</i>
false	<i>falskt</i>

## Elementära funktioner

$\text{abs}(x)$	absolutbeloppet	$ x $
$\text{exp}(x)$	exponentialfunktionen	$e^x$
$\ln(x)$	naturliga logaritmen	$\ln x$
$\log[10](x)$	tiologaritmen	${}^{10}\log x$
$\log[b](x)$	logaritmen i bas b	${}^b\log x$
$\text{sqrt}(x)$	kvadratroten	$\sqrt{x}$
$\sin(x)$	sinus	$\sin x$
$\cos(x)$	cosinus	$\cos x$
$\tan(x)$	tangens	$\tan x$

## Förenkling

<code>expand(u)</code>	utvecklar uttrycket $u$
<code>normal(u)</code>	skriver om uttrycket som en kvot
<code>collect(u)</code>	samlar termer $m$ a $p$ en viss variabel
<code>combine(u)</code>	kombinerar termer
<code>factor(u)</code>	faktorerar polynom
<code>simplify(u)</code>	förenklar uttryck
<code>op(n,u)</code>	plockar ut den $n$ :e operanden ur $u$
<code>rhs(u)</code>	plockar ut högerledet ur $u$
<code>lhs(u)</code>	plockar ut vänsterledet ur $u$
<code>subs(x=a,u)</code>	byter ut $x$ mot $a$ överallt i uttrycket $u$
<code>evalf(u)</code>	beräknar en flyttalsapproximation till $u$

## Analys

<code>diff(f(x),x);</code>	deriverar $f$ m a $p$ $x$
<code>int(f(x),x);</code>	integrerar $f$ m a $p$ $x$
<code>int(f(x),x=a..b)</code>	integrerar $f$ då $x$ går från $a$ till $b$
<code>sum(f(n),n);</code>	summerar $f$ m a $p$ $n$
<code>sum(f(k),k=a..b);</code>	summerar $f$ då $k$ går från $a$ till $b$
<code>series(f,x=a);</code>	Serieutvecklar uttrycket $f$ kring $x=a$
<code>limit(f,x=a);</code>	Gränsvärdet av $f$ när $x$ går mot $a$

## Ekvationslösning

<code>solve</code>	löser ekvationer/ekvationssystem
<code>dsolve</code>	löser differentialekvationer
<code>pdesolve</code>	löser partiella differentialekvationer
<code>fsolve</code>	ger flyttalslösningar
<code>isolve</code>	ger heltalslösningar
<code>msolve</code>	ger lösningar modulo ett primtal
<code>rsolve</code>	löser rekursionsekvationer
<code>linalg[linsolve]</code>	löser matrisekvationer
<code>Groebner[gsolve]</code>	löser m h a Gröbnerbaser

## Evaluering

Normalt evalueras globala variabler så långt det är möjligt och lokala variabler evalueras i en nivå. Kommandot `eval` tvingar fram en evaluering (`eval(uttryck, n)` evaluerar i n nivåer). För att ta reda på vad ett uttryck blir om man ger en av de ingående variablerna ett värde använder man inte `eval` utan `subs`.

<code>eval</code>	evaluerar globala variabler
<code>evalf</code>	ger flyttalsresultat
<code>evalm</code>	evaluerar ett matrisuttryck
<code>evaln</code>	evaluerar till ett namn
<code>evalhf</code>	evaluerar med "flyttalshårdvara"
<code>evalb</code>	boolesk evaluering
<code>evala</code>	evaluerar i en algebraisk talkropp
<code>evalc</code>	ger komplext resultat
<code>Eval</code>	evaluerar polynom
<code>value</code>	evaluerar en inaktiv funktion, t ex <code>Int</code>

## Datastrukturer

<code>sequence</code>	följd	1, 2, 2, 3
<code>list</code>	lista	[1, 2, 2, 3]
<code>set</code>	mängd	{1, 2, 3}
<code>table</code>	tabell	<code>table([a=1 p=3]);</code>
<code>array</code>	vektor, matris etc.	[1 2 2 3]

## Grafik

<code>plot(f, x=a..b);</code>	ritar en graf över f i intervallet a..b
<code>plot({f, g}, x=a..b);</code>	ritar funktionerna f och g i samma bild
<code>plot3d(f, x=a..b, y=c..d);</code>	gör en tredimensionell graf över f
<code>plots[animate]</code>	gör en animerad graf
<code>plots[animate3d]</code>	gör en tredimensionell animerad graf

## Symboler

" 10

# 27

% 17

-> 12

? 30

??? 30

' 27

' 15, 30

## A

absolutbeloppet 11

addition 10

allvalues 17

animering 24

apostrof 30

aritmetiska operatörer 10

array 21, 22

avlusning 28

avsluta 5

## B

bakåta-postrof 27, 30, 41

## C

charpoly 22

citattecken 27, 30

C-kod 29

collect 13

combine 13

convert 20, 22

cosinus 11

## D

datastrukturer 21

decimaltal 10

derivering 15

diff 15

differentialekvationer 16

Digits 11

division 10

D-operatorn 18

dsolve 18

## E

ekvationer 16

ekvationslösning 16

ekvationssystem 16

eval 47

evalf 11, 13

Exit 5

expand 13

exponentialfunktionen 11

exponentiering 10

Export as > HTML... 29

Export as > Plain text... 29

## F

factor 13

fakultet 10

false 10

flyttalsapproximation 11

följd 18

for 25

förenkling 13

FORTRAN-kod 29

framåt-postrof 30

fsolve 17

funktioner 42

funktioner, egendefinierade 12, 26

## G

globala variabler 27

grafik 23

gränsvärden 20

grekiska bokstäver 12

## H

Help Browser 30

Help-menyn 30

hjälp 30

## I

if 25

inaktiv funktion 44

inert 44

infinity 10

inläsning 27

Int 44

int 15

integration 15

inverse 22

## K

kolon 27

kommentarer 27

komplexa rötter 17

konstanter 10

kvadratroten 11

## L

LaTeX 29

lhs 13

limit 20

linalg-paketet 22

list 21

logaritmen 11

lokala variabler 26, 27

## M

Macintosh 7

map 22

matrix 22

mint 28



- modulo 10
- ms-format 29
- multiplikation 10
- N**
- naturliga logaritmen 11
- norm 22
- normal 13
- O**
- olikheter 16
- op 13
- Open 29
- operatorernas prioritetsordning 41
- P**
- packages 44
- paket 44
- parametriserad kurva 23
- piecewise 12
- plot 23
- plot3d 23
- plots-paketet 24
- precision 11
- primitiva funktioner 15
- Print 29
- printlevel 28
- proc 26
- programmering 25
- Q**
- quit 5
- R**
- read 29
- readstat 26
- rekursion 26
- remember 26
- reserverade ord 41
- RETURN 26
- rhs 13, 18
- RootOf 17
- S**
- save 29
- semikolon 27
- seq 21
- sequence 21
- series 20
- seriutveckling 20
- set 21
- simplify 13
- sinus 11
- slingor 25
- solve 16, 45, 46
- sqrt 11
- subs 13
- subtraktion 10
- sum 15
- summation 15
- synvinkel 23
- T**
- table 21
- tangens 11
- textformat 29
- textterminal 8
- tiologaritmen 11
- trace 28
- true 10
- U**
- unapply 12
- utmatning 27
- utskrift 29
- V**
- value 44
- variabel 30
- vector 22
- villkorssatser 25
- W**
- while 25
- Windows 6
- with 22
- X**
- X 5