

Introduktion till Maple

Allmänt

Ett modernt datoralgebrasystem har som huvudfunktion att göra symboliska beräkningar, i motsats till numeriska. Det kan utföra algebraiska manipulationer och förenklingar, lösa ekvationssystem, integrera och derivera symboliskt och lösa differentialekvationer. Dessutom kan de flesta sådana system också utföra numeriska beräkningar och har kraftfulla grafiska funktioner. Behöver man mer omfattande numeriska beräkningar använder man sig dock hellre av Matlab eller specialskrivna numeriska program.

Maple är ett av de ledande datoralgebrasystemen. Några av de starkaste konkurrenterna är Mathematica, Macsyma och Reduce. LTH har en generell licens för Maple och systemet bör därför finnas tillgängligt på studentdatorer. Det kan köras från de flesta vanliga operativsystem.

Maple är ett mycket mångsidigt program, och man kan ägna mycket tid åt att utforska dess användningsmöjligheter. Dess mångsidighet gör det samtidigt svårare att använda än t. ex. Matlab.

Liksom de flesta programpaket uppdateras Maple relativt ofta. Versionerna skiljer sig något åt, och anvisningarna nedan bör fungera bra för den version som finns installerad i datorsalarna.

Mer information om Maple än vad som ryms i denna korta introduktion kan man finna i det inbyggda hjälpsystemet samt i en stort antal skrifter och böcker.

Komma igång

Starta Maple genom att klicka på ikonen på skrivbordet eller i Start-menyn. Man går ur Maple genom att välja *Exit* i *File*-menyn. Behöver man avbryta någon process kan man använda stoppknappen på menyraden.

Hjälp

Maple har ett omfattande inbyggt hjälpsystem. Om man klickar på *Help* i menyraden, så får man tillgång till hjälpfunktioner. Dessa fungerar på lite olika sätt, beroende på vilken Mapleversion man kör, men brukar vara lätta att använda.

Man kan också få hjälp genom att skriva ett sökord föregånget av ett frågetecken, t. ex. `?exp`, `?inifcn`. I slutet på varje hjälptext finns exempel och hänvisningar till andra kommandon som kan vara av nytta.

Fördefinierade konstanter och funktioner

Maple har några fördefinierade konstanter (se `?constant`), t. ex.

`I`, `Pi` och `infinity` (står för i , π resp ∞)

Dessa är reserverade namn, som inte får användas till annat. Observera att Maple skiljer på stora och små bokstäver. Exempelvis måste π (3.14...) skrivas `Pi` (stort P). Skriver man `pi` får man ett π på skärmen, men är detta enbart en symbol och har inget numeriskt värde. För talet e finns ingen symbol utan man skriver `exp(1)` (se `?exp`).

I Maple finns många fördefinierade funktioner (se `?inifcn`), däribland de välkända funktionerna

| | | | | | | | |
|------------------|-------------------|------------------|------------------|---------------------|---------------------|---------------------|---------------------|
| <code>sin</code> | <code>cos</code> | <code>tan</code> | <code>cot</code> | <code>arcsin</code> | <code>arccos</code> | <code>arctan</code> | <code>arccot</code> |
| <code>abs</code> | <code>sqrt</code> | <code>exp</code> | <code>ln</code> | <code>log</code> | <code>log10</code> | | |

(Här står `abs` för absolutbelopp, `sqrt` för kvadratroten, `exp` för exponentialfunktionen (e^x) både `ln` och `log` för naturliga logaritmen och `log10` för tiologaritmen (\lg).

Grundläggande kommandon och aritmetiska beräkningar

I Maple-fönstret hittar du en 'prompt' `>`, som betyder att Maple väntar på ett kommando. Skriv in `7+5`; och tryck på **Enter**, så utförs summationen.

Om man avslutar ett Maple-kommando med ett kolon `:`, så utförs operationen, men resultatet kommer inte upp på skärmen. Dett kan vara användbart om man inte vill fylla skärmen med långa delräkningar som man ändå inte är intresserad av. (I äldre versioner av Maple var det nödvändigt att avsluta varje kommando med ett semikolon `;`, men det behövs inte längre. I de flesta exempel nedan används ändå detta semikolon, mest av gammal vana.)

Kontrollera att Maple kan fungera som en vanlig räknedosa. Prova i tur och ordning

```
2-3*5; 1/2+1/3; sqrt(4)*sqrt(3);
```

Observera att Maple ger exakta svar, inte närmevärden.

Närmevärden erhålles genom kommandot `evalf`; , som står för 'evaluate using floating point arithmetic'. För att få ett närmevärde för den *senast utförda beräkningen* ($2\sqrt{3}$ i vårt fall) kan man skriva

```
evalf(%);
```

Man kan naturligtvis också skriva `evalf(sqrt(4)*sqrt(3))`; . På samma sätt kan resultatet av den näst senaste beräkningen återopas genom `%`, etc.

En praktisk detalj i Maple är möjligheten att gå tillbaka till gamla kommandon på skärmen med pil-*up* och pil-*ner* tangenterna, ändra i kommandona och utföra dem på nytt. Man kan också använda musen för att snabbt flytta sej till olika delar av skärmen. Testa genom att gå tillbaka till kommandot `1/2+1/3`; och gör en annan bråkberäkning genom att ändra siffror.

När det gäller kommandot `evalf` så kan man ur hjälpfunktionen, som nås med `?evalf`, utläsa att man genom ett andra argument kan ange med hur många siffror man vill ha svaret. Använd detta för att få ett värde på π med 1000 decimaler, `evalf(Pi,1000)`; .

Variabler

Maple kan inte bara räkna med tal utan också med variabler och med funktioner. Detta gör att ett Maplesystem blir större och ofta mer invecklat att programmera än ett vanligt programmeringspråk, men också oerhört mycket mera kraftfullt.

Ge kommandot `(x+1)^3`; . Maple svarar med samma sak. Som svar på `expand(%)`; så utvecklar Maple uttrycket enligt binomialsatsen. Prova även med

```
expand((a-b)*(a+b)); expand(x*(x+1)*(x+2)*(x+3)); expand(cos(x+y));
```

Som bekant vill man ofta gå åt andra hållet, och faktoruppdelat ett givet uttryck. Detta görs med kommandot `factor`. Prova detta på de polynomuttryck som du fick ovan.

Ett annat kommando för förenkling är `simplify`. Prova detta genom att skriva

```
simplify(1/(x-1)+1/(x+1));
```

Maple gör inte alltid det man önskar. Ett annat kommando man kan ha nytta av för att göra omskrivningar är `normal`.

Tilldelningssatser

Man kan tilldela en variabel ett värde, numeriskt eller symboliskt. Tilldelningssymbolen är liksom i Pascal och Simula `:=`. Prova med `x:=2`; . Kommandot `x`; ger nu variabelvärdet 2. Prova också med `(x+1)^2`; .

En variabel som fått ett värde behåller detta tills man ger den ett annat värde eller tar bort värdet eller går ur Maple. För att ta bort värdet använder man kommandot `x:='x'`; , som gör att Maple tolkar bokstaven `x` som en variabel, betecknad `x`, inte som variabelns värde. Det är lätt att glömma bort att man gett en variabel ett värde tidigare, vilket kan leda till obegripliga resultat av räkningar. Vill man ta bort alla värden på variabler och helt och hållet börja om från början skriver man `restart`; .

Variabler kan förutom numeriska värden även ha Mapleuttryck som värden. Genom tilldelningskommandot

```
f:=(x+1)^3;
```

så sätter vi f lika med $(x + 1)^3$. (Blir svaret 27 så tag bort det tidigare värdet från x och försök igen.) Man kan sedan räkna vidare med f och skriva t. ex. f^2 ; och `expand(f^2)`;

Om man vill beräkna värdet av uttrycket f^2 för t. ex. $x=2$ utan att varaktigt tilldela x ett värde så kan man skriva

```
eval(x=2,f^2); eller subs(x=2,f^2);
```

Funktioner

Maple kan hantera inte bara analytiska uttryck, som vi sett exempel på ovan, utan även *funktioner*. Dessa kan definieras på flera olika sätt. Det för våra ändamål enklaste påminner om beteckningen

$$x \mapsto g(x)$$

för en funktion g . Ge kommandot

```
g:= x->(x+1)^3;
```

Nu kan funktionsvärden beräknas på det sätt man är van vid. Prova t. ex. med

```
g(0); , g(-1); , g(a); och g(y+z); .
```

Anmärkning: Man måste noga hålla isär begreppen *funktion* och *funktionsuttryck*, där det senare behandlades under rubriken 'Tilldelningssatser' ovan. För säkerhets skull upprepar vi skillnaden:

Om vi ger ett värde till F genom tilldelningen `F:=exp(x)-sin(x)`; så är F ett *uttryck* med variabeln x inbyggt. Om vi i stället ger F ett värde genom `F:=x->exp(x)-sin(x)`; så är F en *funktion*, där x bara används för att definiera en regel, och där man skulle kunna använda vilken annan symbol som helst. Funktioner är mycket mer flexibla, men ibland något mer svårhanterliga än funktionsuttryck. Det finns ett enkelt sätt att göra om ett uttryck till en funktion, nämligen genom att använda kommandot `unapply`. Till exempel ger kommandot `F:=unapply(exp(x)-sin(x),x)`; samma sak som `F:=x->exp(x)-sin(x)`;

Maple klarar också funktioner av flera variabler. Funktionen $h(x, y) = x^2 - y^3 + x^2y^3$ definieras i Maple på följande sätt:

```
h:= (x,y) -> x^2-y^3+x^2*y^3;
```

Vad ger `h(0,0)`; och `h(3,2)`;

Derivation

Derivator beräknas med kommandot `diff`. Användningen framgår ur exemplet

```
diff(x^3+2*x^2,x); och diff(tan(x),x);
```

Andraderivator beräknas enligt

```
diff(tan(x),x,x);
```

På samma sätt beräknas derivator av högre ordning, där antalet x anger ordningen. För detta finns ett kortare skrivsätt. Beräkna fjärdederivatan av $\tan(x)$ genom att skriva

```
diff(tan(x),x$4);
```

Här betyder `$4` att derivationen upprepas 4 gånger. Prova även att förenkla svaret med hjälp av `factor(%)`;

Vi kan derivera det tidigare definierade *funktionsuttrycket* f och *funktionen* g genom att skriva

`diff(f,x);` resp `diff(g(x),x);`

Tänk ut vad som kommer ut ur kommandona

`diff(f,y);` , `diff(g,x);` , `diff(g(y),y);`

och prova om det stämmer.

Anmärkning: Maple har också en derivationsoperator `D`, se `?D`. Prova t. ex. `D(g);` och `D(sin);`. Bokstaven `D` kan inte användas som symbol för något annat.

Man kan även lätt beräkna partiella derivator. Till exempel får man

$$\frac{\partial h}{\partial x} \quad \text{och} \quad \frac{\partial h}{\partial y}$$

för funktionen h ovan genom kommandona

`diff(h(x,y),x);` resp `diff(h(x,y),y);`

Högre derivator fås genom uppräknings av variabelnamnen. Prova t. ex.

`diff(h(x,y),x,y);`

Integration

Med hjälp av kommandot `int(f,x);` kan man beräkna en primitiv funktion till funktionen $f(x)$ med avseende på x . Integrationskonstanten får man sedan själv lägga till om det skulle behövas. Om vi definierat ett funktionsuttryck f och en funktion g som på förra sidan, så fås en primitiv funktion med

`int(f,x);` resp `int(g(x),x);`

Repetera nu några primitiva funktioner från envariabelanalysen t. ex.: $\cos x$ (med kommandot `int(cos(x),x);`), e^t (skrivet `exp(t)`), och $\frac{1}{\sqrt{1-z^2}}$ med avseende på respektive variabel. Kvadratrotfunktionen skrivs i Maple som `sqrt`.

Om Maple inte kan någon primitiv funktion, så svarar Maple genom att ge tillbaka det man stoppat in. Försök med `int(x^x,x);` så får du tillbaka

$$\int x^x dx.$$

Prova också

$$\int \sqrt{\frac{1-2x^2}{1-x^2}} dx \quad \text{och} \quad \int e^{-x^2} dx .$$

Dessa exempel visar att Maple känner till fler primitiva funktioner än de vanliga elementära funktionerna. Den första av integralerna är exempel på en så kallad *elliptisk integral*. Vissa elliptiska integraler finns i Maple, se t. ex. `?EllipticE`. I svaret på den andra integralen ingår den så kallade 'error function', `erf`. Denna uppträder bl a inom statistiken och i samband med värmeledning.

För att beräkna integraler, till exempel

$$\int_{-2}^3 (x+1)^3 dx$$

måste även gränserna anges. Detta görs i Maple genom `x=-2..3`, eller allmänt `x=a..b` för $a \leq x \leq b$. Integralen ovan beräknas genom

`int((x+1)^3,x=-2..3);`

Gränserna får gärna innehålla variabler, försök t. ex. med `int(f,x=-y..(z+1));`.

Även generaliserade integraler kan beräknas. Prova t. ex. med

`int(1/(1+x^2),x=0..infinity);`

Använder man en stor begynnelsebokstav (t. ex. skriver `Int(1/(1+x^2),x=0..infinity);`) gör inte Maple någon beräkning utan svarar med en snygg utskrift av det man skrivit in. Detta är bra att använda om man vill kontrollera att man skrivit rätt.

Summation

Maple kan också användas för att beräkna summor. Syntaxen för summation är naturlig, $\sum_{k=m}^n a_k$ fås genom Maplekommandot `sum(a(k),k=m..n)`.

Beräkna potenssummorna

$$\sum_{k=1}^n k \quad \text{och} \quad \sum_{k=1}^n k^2$$

med hjälp av kommandona `sum(k,k=1..n)`; och `sum(k^2,k=1..n)`; . Förenkla med kommandot `simplify`. Även vissa serier kan Maple beräkna. Vad blir $\sum_{k=1}^{\infty} \frac{1}{k^2}$?

Gränsvärden

Gränsvärden beräknas med kommandot `limit`, prova

```
limit(sin(x)/x,x=0);    limit((sqrt(1+exp(x))-1)/exp(x),x=infinity);
```

Taylor- och Maclaurinutvecklingar

Med kommandot `taylor` får man Taylor- (Maclaurin)utvecklingen till ordning 5, t. ex. ger

```
taylor(exp(sin(x)),x);
```

Maclaurinutvecklingen av $e^{\sin x}$. Vill man istället taylorutveckla $\sin x$ kring punkten $x = 1$ till ordning 9 (restterm av ordning 10) skriver man

```
taylor(sin(x),x=1,10);
```

Ekvationslösning

Maple kan användas för att lösa ekvationer, se `?solve` och `?fsolve`. Prova de tre varianterna

```
solve(x^3+x^2+x-3=0,x);
fsolve(x^3+x^2+x-3=0,x);
fsolve(x^3+x^2+x-3=0,x,complex);
```

och läs om dem i hjälpfunktionen. Prova också

```
solve(sin(x)=1/2,x);
```

I det senare fallet ger Maple endast en lösning. För att få alla lösningar kan man skriva

```
_EnvAllSolutions := true: solve(sin(x)+cos(x)=sqrt(2),x);
```

Differentialekvationer

Maple kan också lösa differentialekvationer, se `?dsolve`. Prova med att lösa

$$y'' + 2y' + 2y = e^{-x}$$

med kommandot

```
dsolve(diff(y(x),x$2)+2*diff(y(x),x)+2*y(x)=exp(-x), y(x));
```

Har vi dessutom begynnelsevillkoren $y(0) = y'(0) = 0$ kan vi skriva

```
dsolve({diff(y(x),x$2)+2*diff(y(x),x)+2*y(x)=exp(-x),
y(0)=0,D(y)(0)=0}, y(x));
```

Grafik

Maple har omfattande grafiska möjligheter. Om du vill att Maples figurer hamnar i egna fönster (vilket kan vara lättare att hantera) så går du in under menyn *File*, väljer *Preferences* och sen *Plotting, Window*. Antalet grafikfönster har en tendens att växa snabbt. Man stänger dem man inte behöver genom att gå in under *File* och dra till *Close* när grafikfönstret är uppe.

Med kommandot `plot` ritar man tvådimensionella figurer i Maple, t. ex. funktionskurvor. Se hjälpen till kommandona `plot`. Prova med

```
plot(sin(x),x=-Pi..Pi);
```

Axlarna skalas automatiskt så att bilden fyller grafikfönstret maximalt. Om man vill ha samma skala på båda axlarna kan man gå in på menyraden vid *Projection* och välja *Constrained*. Detta kan man också åstadkomma genom att direkt skriva

```
plot(sin(x),x=-Pi..Pi,scaling=constrained);
```

Vill man hellre ha en grön kurva skriver man

```
plot(sin(x),x=-Pi..Pi,scaling=constrained,color=green);
```

Man kan också välja olika typer av axlar, linjetjocklek, linjestil antingen från menyraden eller direkt på kommandoraden.

För att rita flera kurvor i en figur använder man `[]`-parenteser (även `{ }`-parenteser fungerar). Prova

```
plot([sin(x),x-x^3/6],x=-Pi..Pi);
```

Vad är det för speciellt med polynomet?

Man kan också rita kurvor i parameterform. Enhetscirkeln $(x, y) = (\cos t, \sin t)$, $-\pi \leq t \leq \pi$ ritas med

```
plot([cos(t),sin(t),t=-Pi..Pi],scaling=constrained);
```

Observera att parameterintervallet skrivs innanför hakparentesen.

Maple har många fler grafikrutiner. För att få tillgång till dessa görs kommandot

```
with(plots);
```

Här är `plots` namnet på ett så kallat 'package'. Det finns ett antal sådana i Maple, för olika användningar. Avslutar man med `;` får man en lista över de nya kommandon man får tillgång till. Ett annat 'package' för grafik är `plottools`.

Nu kan vi också rita implicit givna kurvor, t. ex. $\frac{x^2}{4} + \frac{y^2}{9} = 1$, med kommandot `implicitplot`. Ange ekvationen samt ett lämpligt intervall för x och y .

```
implicitplot(x^2/4+y^2/9=1,x=-3..3,y=-3..3);
```

Välj här *Projection Constrained*.

Ibland är det bra att först definiera och lagra bilden och sen titta på den. Om man vill se de båda kurvorna $x^2/4 + y^2/9 = 1$ och $(x, y) = (2 \cos t, 3 \sin 3t)$, $0 \leq t \leq 2\pi$ i samma bild kan man göra så här: Låt A betyda bilden av den första kurvan och B bilden av den andra, skriv

```
A:=implicitplot(x^2/4+y^2/9=1,x=-3..3,y=-3..3,color=red):  
B:=plot([2*cos(t),3*sin(3*t),t=0..2*Pi],color=green):
```

Avsluta här med `:`, ej med `;`. (Avslutar man med `;` här visas ingen bild, utan skärmen fylls av siffror.) Maple visar nu inga bilder utan lagrar bilderna A resp B . Om vi nu skriver

```
display([A,B]);
```

visar Maple båda kurvorna i en bild. Välj här *Projection Constrained*.

Maples tredimensionella grafik har en mängd varianter, läs om dem i ?plot3d.

Funktionsytan $z = xe^{-x^2-y^2}$, $-2 \leq x \leq 2$, $-2 \leq y \leq 2$ kan ritas med

```
plot3d(x*exp(-x^2-y^2),x=-2..2,y=-2..2);
```

Man kan vrida och vända på bilden genom att placera musmarkören i grafikfönstret och hålla vänster musknapp nere och flytta på musen eller klicka pilarna bredvid vinkelbeteckningarna ϑ och ϕ i menyraden. Prova med hjälp av menyer de olika möjligheterna för 'axes', 'color' och 'style'.

Definitionsmängden behöver ej vara en rektangel, gränserna i y kan bero på x , prova t. ex.

```
plot3d(x*exp(-x^2-y^2),x=0..2,y=-x..x);
```

Vad är definitionsmängden här?

Vill man bara se funktionens nivåkurvor skriver man

```
contourplot(x*exp(-x^2-y^2),x=-2..2,y=-2..2);
```

Det går också bra att rita ytor i parameterform. Cylindern $y^2 + z^2 = 1$ längs x -axeln kan ritas med

```
plot3d([s,cos(t),sin(t)],s=-2..2,t=0..2*Pi,scaling=constrained);
```

Observera att parameterintervallen här skrivs utanför hakparentesen. (Här skiljer sig alltså det tre- och tvådimensionella fallen åt.)

Vill man titta på två cylindrar (en längs x -axeln och en längs y -axeln) som skär varandra kan man göra så här:

```
A:=plot3d([s,cos(t),sin(t)],s=-2..2,t=0..2*Pi):
B:=plot3d([cos(t),s,sin(t)],s=-2..2,t=0..2*Pi,color=s*t):
display([A,B]);
```

Här är gjort ett färgval på cylindern B , färgen på cylindern A kan varieras från menyraden. Välj här *Projection Constrained*. Vrid och vänd på figuren. Prova olika möjligheter av 'color' och 'style'. Vill man särskilt studera skärningen mellan cylindrarna ($y^2 + z^2 = 1$ resp $x^2 + z^2 = 1$) dvs den kropp som ligger 'innanför' båda cylindrarna skriver man

```
plot3d({-min(sqrt(1-x^2),sqrt(1-y^2)),min(sqrt(1-x^2),sqrt(1-y^2))},
x=-1..1,y=-1..1);
```

Pröva även olika *Light Scheme* under *Color*.

För att rita cylindrar, och andra ytor med rotationssymmetri kring z -axeln, snyggt är det bra att använda *cylinderplot*. Exempelvis fås konen $z = \sqrt{x^2 + y^2}$, $x^2 + y^2 \leq 4$ av

```
cylinderplot(z,t=0..2*Pi,z=0..2);
```

Här anges avståndet från z -axeln ('radien') $r = \sqrt{x^2 + y^2}$ som funktion av z . Den första variabeln anger vinkeln till x -axeln. Paraboloiden $z = x^2 + y^2$, ($r = \sqrt{x^2 + y^2} = \sqrt{z}$) fås av

```
cylinderplot(sqrt(z),t=0..2*Pi,z=0..2);
```

Samma sak erhålls med

```
plot3d(sqrt(z),t=0..2*Pi,z=0..2,coords=cylindrical);
```

För mera komplicerade ytor se ?cylinderplot.

En sfär kan även ritas med kommandot

```
sphereplot(1,t=0..2*Pi,s=0..Pi,scaling=constrained);
```

alternativt

```
plot3d(1,t=0..2*Pi,s=0..Pi,coords=spherical,scaling=constrained);
```

Här är den första variabeln den vinkel som kallas φ i flerdimboken och den andra variabeln θ , se även ?sphereplot och ?coords.

Implicit givna ytor kan ritas med `implicitplot3d`, pröva t. ex.

```
implicitplot3d(x^2+y^2-z^2=1,x=-2..2,y=-2..2,z=-2..2);
```

(Denna yta kan ritas snyggare med `cylinderplot`.) Prova också

```
implicitplot3d(x^2+4*y^2+9*z^2=1,x=-1..1,y=-1/2..1/2,  
z=-1/3..1/3,scaling=constrained);
```

Fler roliga exempel på funktionsytor finns att hämta i övningshäftet i flervariabelanalys.

Kurvor i rummet ritas med `spacecurve`, prova

```
spacecurve([t*cos(t),t*sin(t),t],t=0..3*Pi):
```

Här skall parameterintervallet skrivas utanför hakparentesen.

Vill man se kurvan tydligare kan man lägga in den på en kon (jfr flerdimboken sid 24) skriv

```
A:=spacecurve([t*cos(t),t*sin(t),t],t=0..3*Pi,color=red,thickness=4):  
B:=cylinderplot(z,t=0..2*Pi,z=0..10):  
display([A,B]);
```

Spara och skriva ut

Man kan spara all text i Maple-fönstret genom att klicka på *File* och *Save as* samt ange ett filnamn. Texten sparas då i en fil med namnet `filnamn.mws`. När du senare vill ta fram texten igen klickar du på *File* och *Open* och anger hela filnamnet, även `.mws`. Det går också att samla alla kommandon i en scriptfil med ett enkelt namn, t. ex. `prov`. För att köra scriptet skriver du i Maple-fönstret `read prov`; I vissa versioner används filändelsen `.mw` i stället för `.mws`.

Om du vill ta ut en bild på skrivare, kan du spara den i Post-Script format genom att klicka på *File* och *Print* samt ange filnamn.ps. Alternativt kan man skicka bilden direkt till skrivaren.