



Maple 10

Handledning

Ivan Milles

OM MAPLE

Varför Maple?

Maple ger dig möjligheten att snabbt gå igenom beräkningar och se var du räknat fel, ge snabba svar när det inte är räkningen som är viktig, att rita upp grafer och att experimentera med begrepp för att få en bättre förståelse för dem... Många uppgifter man stöter på i verkligheten är alldeles för svåra att räkna för hand. Dessutom är det ett bra sätt att kolla ifall man har fått rätt svar, om svaret i facit ser annorlunda ut: mata in ditt svar och facit-svaret i Maple och be Maple förenkla bägge uttrycken, och se ifall de är detsamma.

Mer än en gång har studenter som använt Maple i de inledande mattekurserna sagt att det är krångligare, mer arbete och i största allmänhet ett slöseri med tid att sätta sig in i Maple. Förhoppningsvis ska den här handledningen göra det lättare att lära sig Maple, så att de (fåtal) timmar det tar att lära sig betalar sig i längden. Om man ser bortom studietiden så är Maple ett program som används enormt mycket inom ingenjörsarbete och forskning, så det är åtminstone en investering inför framtiden. Maple används i "verkligheten".

Varför då inte sitta med grafräknare? Dels är Maple mycket mer kapabelt och bekvämt att arbeta med. Det viktigaste är dock att Maple är *symbolhanterande* – det vill säga att Maple inte bara kan beräkna ett uttryck, utan även förenkla, derivera, integrera; och då inte bara ge derivatans värde i en punkt, utan ge den som en funktion

Maple kan förstås ersättas av andra programspråk. Det kan vara lockande för den som redan kan C/C++, Java, Python eller liknande att skriva sina beräkningar i det språket. Naturligtvis finns det inget som hindrar det, men Maple är specifikt skapat för att hantera just matematiska uttryck. Kan man ett programmeringsspråk sedan tidigare tar det inte lång tid att lägga Maple till sin repertoar.

Vad är Maple?

Maple är kort och gott två saker: ett programmeringsspråk och ett datorprogram. Man matar in kommandon i programmet och får ut resultatet. I Maple finns så gott som alla matematiska funktioner inbyggda, så de flesta uppgifter går att mata in ganska lätt. Till det finns uttömmande hjälpavsnitt, som inte bara förklarar hur programmet fungerar, utan som även förklarar de matematiska begreppen. Hjälpavsnitten kan alltså användas som extra uppslagsbok utöver en mattebok.

Maple installeras på din dator som vilket annat program som helst. Den version som används i den här handledningen är Maple 10, som har blivit mycket lättare att använda än tidigare versioner.

Hur använder man Maple?

Här kommer några korta exempel på vad man kan göra med Maple:

- Skriva ut π med 50 decimaler:

```
> evalf[50](Pi);  
3.1415926535897932384626433832795028841971693993751
```
- Lösa ekvationen $e^{2x} + e^x = 4$

```
> solve (exp(2x) + exp(x) = 4);  
ln(-1/2 ± 1/2 * sqrt(17))
```
- Beräkna derivatan av $f(x) = 3x * \cos(\sin(x))$

```
> diff(3x * cos(sin(x)));  
3*(cos(sin(x)) - x*sin(sin(x)))*cos(x)
```

Komma igång med Maple	3
<i>Hur Maple fungerar</i>	3
<i>Goda Maple-vanor</i>	4
<i>Enkel inmatning</i>	4
<i>Matematiska funktioner och standarduttryck</i>	5
<i>Grafisk inmatning</i>	7
<i>Paket</i>	7
<i>Enkel grafitrning</i>	7
Algebra	9
<i>Ekvationslösning</i>	9
<i>Ekvationssystem</i>	10
<i>Förenkling, utveckling, faktorisering och substituering</i>	11
<i>Komplexa tal</i>	13
Linjär algebra	14
<i>Skapa matriser och vektorer</i>	14
<i>Matrisoperationer</i>	14
<i>Gausselimination och radoperationer</i>	14
<i>Determinanter, inverser, egenvärden och egenvektorer</i>	15
Analys	16
<i>Derivator</i>	16
<i>Integraler</i>	16
<i>Styckvis definierade funktioner</i>	17
<i>Gränsvärden</i>	17
<i>Differentialekvationer</i>	18
<i>Partiella differentialekvationer</i>	19
<i>System av differentialekvationer</i>	20
<i>McLaurin- och Taylorutvecklingar</i>	20
Studiehjälp i Maple	21
<i>Studiehjälp för inledande matematikkurser</i>	21
Plottning	22
<i>Plot Builder</i>	22
<i>Plottning av parametriska funktioner ($R \rightarrow R^n$)</i>	23
<i>Animation och interaktion ($R^2 \rightarrow R$)</i>	23
<i>Plottning i 3D</i>	23
<i>Spara plottar som bild</i>	24
<i>Exempel: animera hyperboloider</i>	24
Maples hjälpavsnitt	26
<i>Öppna Maples hjälpavsnitt</i>	26
<i>Om Maples hjälpavsnitt</i>	26
<i>Hur man läser kommandohjälpen</i>	26
Simulering	27
<i>Inledande programmering</i>	27
<i>Flödeskontroll</i>	27
<i>Monte Carlo-metoden. Approximera π.</i>	28
Rapporter	30
<i>Städa upp arbetsbladet</i>	30
<i>Textinsättning, formgivning, LaTeX och utskrift</i>	30
Felsökning	31
<i>Vanliga felmeddelanden</i>	31
<i>Vanliga fel</i>	32

KOMMA IGÅNG MED MAPLE

Hur Maple fungerar

I och med version 10 blev Maple lättare att använda. Tidigare använde man sig nästan uteslutande av inmatade textkommandon, vilket är ett snabbt sätt att arbeta, men känsligt för felstavningar, missade mellanslag och liknande. I Maple 10 kan man fortfarande klara sig helt på samma kommandon som i Maple 9 och tidigare, men många uppgifter kan lösas med musen.

Maple startar med ett tomt arbetsblad (“worksheet”) som helt enkelt motsvarar ett papper att räkna på. Detta arbetsblad kan man spara, maila och skriva ut, precis som vilket textdokument som helst.

Det är gott om knappar och menyer, men man klarar sig utan de flesta. Det man behöver bekymra sig om är textfönstret till höger, och i viss mån verktygen Expression, Common Symbols och Matrix till vänster.

I textfönstret skriver man in funktioner, uttryck och ekvationer, samt

Maple-kommandon. Den vanliga arbetsgången är att man skriver in ett uttryck eller ekvation och trycker på Enter eller Retur-knappen på tangentbordet för att

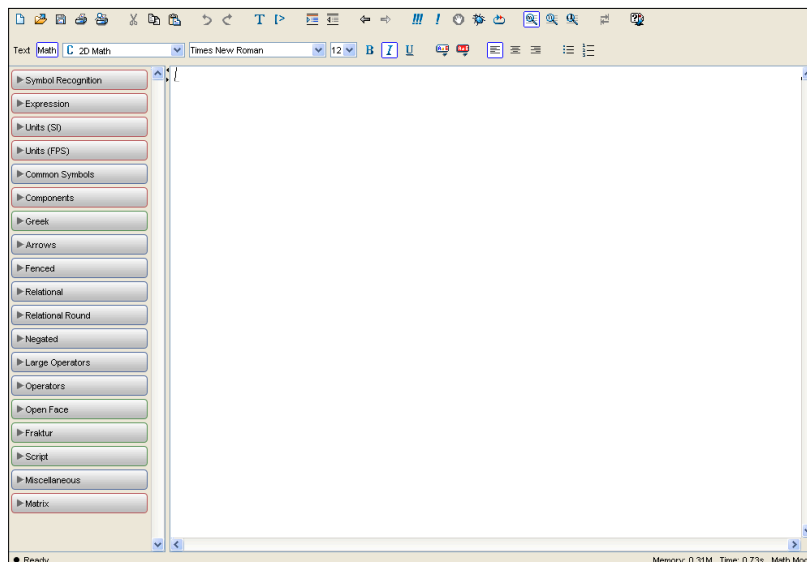
mata in det i Maple. Maple kommer då att tolka det du skrev in, och skriva ut resultatet i mitten av fönstret. Du kan sedan fortsätta att arbeta med resultatet – kanske be Maple derivera en funktion, lösa en ekvation eller rita upp en graf över en lösningskurva.

Man kan tilldela namn till de uttryck man arbetar med. På samma sätt som en variabel i vanlig programmering kan man tilldela namn till vanliga värden, men även till hela uttryck eller till och med hela ekvationer. Namntilldelning görs med menyvalet Assign to a Name eller med := (kolon, likhetstecken) inte enbart med likhetstecken. Ett likhetstecken betyder just det i Maple – att vänster och höger led är lika (en ekvation). Några exempel:

```
a := 2; # Vanlig variabel
2x + 5; # Vanligt uttryck
2x + 5 = 1; # Ekvation
2x + 5 = a # Ekvation som använder tilldelad variabel
minekvation := 2x - 5 = b - 1; # Ekvation döpt till "minekvation"
solve(minekvation); # Be Maple lösa ekvationen
```

Den sista raden visar ett textkommando i Maple. Kommandot solve löser ekvationen och matar ut svaret i blå text.

Genom att högerklicka (ctrl-klicka på Mac, om man har en enknappsmus) på minekvation (antingen den blå utmatningen från Maple, namnet minekvation eller själva ekvationen) får man upp en meny med olika kommandon att utföra. Man kan härifrån välja antingen Solve (vilket gör precis samma sak som textkommandot solve) eller Solve for variable (vilket låter dig lösa ekvationen för antingen x eller a). Lika lätt är det att integrera, derivera, förenkla... (Ekvationen minekvation ovan går inte att derivera, eftersom man bara kan derivera uttryck.) Prova gärna att mata in och högerklicka på $(x-1) * (3x-2)$ och välj Differentiate för att se hur lätt det är. Maple svarar alltid med blå text, utom om det blir något fel. Då får du ett felmeddelande i röd text, som inte alltid är helt lätt att tolka. Titta gärna i kapitlet Felsökning för tips om vanliga felmeddelanden.



En annan sak som är värd att ha i bakhuvudet är att Maple gör mycket arbete bakom kulisserna och därför är ett ganska långsamt program i vissa situationer. Är man van vid en vanlig miniräknare så kan man bli frustrerad av att vänta flera sekunder, eller tro att programmet har hängit sig. Ett bra tips är i så fall att titta efter texten "Evaluating..." i övre (eller nedre, beroende på plattform och version) vänstra hörnet.

Goda Maple-vanor

Även om Maple är ett tämligen enkelt program att använda finns det fallgropar och bättre eller sämre sätt att arbeta. Det är lätt att svepas med och börja arbeta fort, men precis som med vanlig handräkning kan man spara mycket tid på att vara extra noggrann.

- Ett grundläggande tips som *kommer* att underlätta arbetet är att dela upp stora uppgifter i flera små. Lite planering räcker långt. Att tackla en stor uppgift genom att bara mata in den i Maple kommer – åtminstone i början – leda till att man inte kommer någon vart. Istället för att försöka rita upp två hyperbelytor i rymden som representerar en lösning på ett system av differentialekvationer direkt i ett enda svep, måste du angripa det på rätt sätt: mata in de variabler och värden du har och namnge dem, mata in ekvationerna och sätt in dem i ett namngivet system. Sätt in begynnelsevärden och namnge och lös ekvationerna. Välj bra intervall att rita upp resultaten över, och sätt slutligen in resultaten och intervallen i uppritningsfunktionen. Genom att arbeta stegvis, noggrant och kontrollera resultatet efter varje steg är det mycket lättare att upptäcka fel. Att hitta ett fel på en rad är mycket lättare än att hitta ett fel i ett helt arbetsblad.
- Skriv in alla värden som följer med uppgiften i namngivna variabler. På så sätt undviker man att skriva in fel värden någonstans, man får det mycket lättare att ändra värdena i efterhand, lättare att testa olika lösningar och lättare att hitta fel. Vissa uppgifter handlar dessutom om att testa samma situation med olika värden eller ändrade förutsättningar. Man vill inte tvingas skriva om allting om man kan komma undan med att ändra några variabler och köra om programmet.
- Sudda bort sådant som blir fel. Skriver man in en felaktig ekvation kan man alltid sudda och göra om. Om man bara sparar sådant som faktiskt blev rätt blir arbetsbladet mycket mer lättöläst, lättare att leta fel i, och det blir lättare att komma ihåg vad programmet faktiskt gör när man kommer tillbaka nästa lektionstillfälle.

Enkel inmatning

Det mesta arbetet görs med tangentbordet och genom att skriva in text. Man skriver in en rad eller ett kommando, och trycker sedan på Enter/retur för att utföra det som står på raden. I äldre versioner av Maple måste varje inmatningssekvens avslutas med ett semikolon (;, skift+kommatecken) eller ett kolon (:, skift+punkt). Maple 10 klarar sig utan semikolon i de flesta fall, men det är en god vana. Ett semikolon avslutar inmatningen, medan ett kolon avslutar inmatningen men hindrar Maple från att skriva ut ett svar. I de flesta fall vill man se svaret, men vissa funktioner skriver ut långa svar, och ibland vill man slippa bekräftelser från självklara rader, och kan då "tysta" Maple med ett kolon på slutet av raden.

Långa inmatningar kan vara bra att bryta upp på flera rader. Detta gör man genom att trycka skift+retur. På så sätt kan man göra komplicerade beräkningar eller inmatningar mer lättölästa. En sådan uppdelad inmatning måste avslutas med semikolon.

Vill man sätta in en ny rad mellan två rader som redan är skrivna göres detta med Insert→Paragraph, eller ctrl-skift-k (Windows), kommando-skift-k (Mac) eller ctrl-meta-k (Unix).

När man har skrivit in en rad och Maple har svarat vill man oftast fortsätta arbeta med resultatet. Har man namngivit svaret (till exempel $ekv := 2x - 5$) kan man använda sig av namnet i efterföljande kommandon (`solve(ekv)`). Har man inte namngivit svaret kan man komma åt det ändå – det senaste svaret får alltid namnet `%`. (Skift-5). Ovanstående kunde alltså ha skrivits $2x - 5$; `solve(%)`; Dock måste man komma ihåg att `%` alltid representerar det senaste svaret, inte nödvändigtvis svaret på raden ovan. Man kan nämligen markera en rad och trycka på retur för att köra raden igen. Kör man då raderna i en annan ordning kommer rader som använder sig av `%` inte komma att ge samma resultat som gången innan.

Maple kör alltså just de rader som markeras och trycks retur på, inte raderna över, och inte raderna under. Vill man köra om hela programmet med nya värden kan man trycka på knappen med tre utropstecken i översta knappraden. Då körs alla rader i ordning, uppifrån och ned. Det är ett bra sätt att försäkra sig om att man har en helt "fräsch" körning.

Maple svarar alltid i exakt form, det vill säga att man inte får ut decimaltal som svar, oavsett hur komplexa uttrycken blir. Vill man ha ett decimalt svar kan man ta till funktionen `evalf()` (Evaluate Floating-point). Till exempel blir `evalf(1/3)` 1.33333...

Tilldelning av namn är ett mycket bra verktyg för att hålla reda på variabler, delsteg, ekvationer och funktioner. För att tilldela ett namn skriver man `namn := uttryck` och trycker på Enter. **Dock kan man ibland vilja "tömma" ett namn.** Man kan ha tilldelat `a` till värdet 5, men senare vill man använda `a` som en okänd variabel. Lösningen är att skriva:

```
a = 'a'
```

För att rensa alla namn och "nollställa" hela körningen använder man textkommandot `restart`.

Maple använder många "ovanliga" tecken på tangentbordet. Beroende på vilken sorts dator man sitter vid är knapptryckningarna olika, så här är en lista över tecknen, vad de betyder och hur man skriver in dem:

Tecken	Windows	Mac	Unix
{ och }	Alt Gr + 7 och 0	Alt + Skift + 8 och 9	Alt Gr + 7 och 0
[och]	Alt Gr + 8 och 9	Alt + 8 och 9	Alt Gr + 8 och 9
^	Skift + ` (bredvid Å)	Skift + ` (bredvid Å)	Skift + ` (bredvid Å)
< och >	Tangenten bredvid Z	Tangenten bredvid Z	Tangenten bredvid Z
(rakstreck)	Alt Gr + <	Alt + 7	Alt Gr + <
Tab	Tangenten bredvid Q	Tangenten bredvid Q	Tangenten bredvid Q

Karet-tecknet `^` kan vara lite kinkigt – det syns inte förrän man skriver in det efterföljande tecknet. Ibland kan det vara värt att försöka med ett mellanslag istället för siffror eller variabler. Fungerar det ändå inte kan du pröva att trycka på Num Lock-knappen (som oftast sitter uppe till vänster på det numeriska tangentbordet) och försöka igen.

Man kan (och bör) kommentera sin kod för att den ska vara lättare att läsa och förstå när man inte arbetat med den ett tag. All text som kommer efter ett `#` (skift + 3) på en rad ignoreras helt av Maple och kallas för kommentar. Ett bra sätt att lösa större uppgifter är att skriva in sin tankegång i kommentarer först, och fylla i koden i efterhand.

Matematiska funktioner och standarduttryck

Att skriva in matematiska uttryck görs ganska naturligt i Maple. I de flesta fall är det bara att skriva in dem som man förväntar sig att de ska skrivas. En del uttryck är dock inte helt självklara. Några tips:

- Multiplikationstecken skrivs med `*`. I många fall kan man dock utelämna tecknet och istället skriva in ett mellanslag, till exempel `x y` eller `(x-1) (x-2)`. (Undantaget är mellan siffror, man kan alltså inte skriva `2 3` för att beräkna $2 \cdot 3$.) Mellan numeriska koefficienter och variabler kan man utelämna tecknet helt, t.ex. `3x`.
- Division görs med snedstreck. Maple 10 sätter all efterföljande inmatning under divisionsstreckets tills man trycker på högerpilen för att komma upp på raden igen.
- Decimalkomma skrivs in med punkt, inte kommatecken. Detta är en amerikansk konvention som skiljer sig från den europeiska. Kommatecknet används för att skilja på element i listor.
- Exponentiering skrivs antingen med `^` eller `**` (karet eller två multiplikationstecken), till exempel `x^2` eller `x**4`. Vilket man föredrar är en smaksak – med kareten kommer exponenten att skrivas in på "rätt plats". Tryck på högerpilen för att komma ner på raden igen.
- Intervall skrivs med två punkter mellan vänster och höger gräns. Plottning och ekvationslösning i ett intervall kan kräva att man anger att en variabel ligger mellan två värden: `x=-3..3`, `y=0..10`.
- Fakulteter skrivs med utropstecken, till exempel `5!` eller `b!`.
- Absolutbelopp skrivs mellan två rakstreck, till exempel `|-x|`, eller som `abs(-x)`.

- Index skrivs efter ett lågstreck, till exempel x_1 , v_{kon} , eller e_x . Var noggrann med att trycka högerpil efter indexet, eventuella semikolon eller likhetstecken i indexet kommer att ge felet "invalid subscript".
- Ekvationer kännetecknas av att det finns ett likhetstecken med i uttrycket, till exempel $2x = 5y$.
- Det finns två sätt att skriva in **egna funktioner**. Antingen nöjer man sig med ett vanligt uttryck, som $f := 3t + 9$, eller så skriver man $f := t \rightarrow 3t + 9$. I de allra flesta fall klarar man sig bättre med den "korta" formen än den längre, av ganska abstrakta, tekniska skäl.¹
- Man kan spara några knapptryckningar genom att skriva in början på funktionsnamn eller symbolnamn och trycka på ctrl-mellanslag (Windows), Escape (Mac) eller ctrl-skift-mellanslag (UNIX). Maple kommer att försöka fylla i resten åt dig. Detta är också ett bra sätt att undvika skrivfel.
- Mängder skrivs in mellan "måsvingar", till exempel $\{2, 3, 4\}$ eller $\{2x, x^2\}$. Många funktioner kan ta emot flera uttryck och arbeta med dem samtidigt. Grafritningsfunktioner ritlar till exempel upp alla funktionerna i mängden i samma graf, ekvationslösningssystemet tolkar mängden som ett ekvationssystem och löser det.

Alla kända matematiska standardfunktioner finns i Maple. Dessa är oftast uttryckta som i matteböcker, till exempel $\sin()$, $\arcsin()$, $\cos()$, $\arccos()$, $\tan()$, $\arctan()$ och så vidare. Man använder dem genom att skriva argumentet till funktionen inom parentes, till exempel $\sin(x)$, $\arcsin(x^3)$ eller $\arccos(0)$. Var uppmärksam på att det inte går att skriva $\sin x$ eller $\sin x$ som i vissa böcker – detta tolkas som en variabel vid namn $\sin x$ respektive en variabel \sin multiplicerad med x .

Vissa funktioner är inte helt uppenbara, till exempel exponentialfunktionen e : den skrivs som $\exp(x)$, vilket tolkas som e^x . Kvadratrotsdragning görs med funktionen \sqrt{x} . Tredje- och fjärderötter (eller högre) görs lättast genom exponentiering: $x^{(1/3)}$.

Ett mycket vanligt fel är att råka skriva in funktionerna med ett mellanslag mellan namnet och första parentesen. Då kommer Maple inte att behandla uttrycket alls, utan bara att skriva ut samma sak som svar. Sinus för x heter alltså $\sin(x)$ men inte $\sin (x)$ – det andra är direkt fel. Detta kan vara mycket förvirrande för den som har programmerat förut, och kan vara ett mycket subtilt problem.

Maple känner också till de vanligaste konstanterna. π skrivs som Pi (notera stor bokstav i Pi), oändligheten som infinity . Behöver man konstanten e skriver man $\exp(1)$, vilket ju är e .

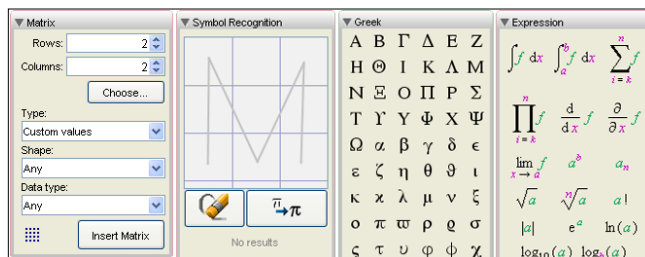
Vill man använda sig av grekiska bokstäver i en uträkning är det i de flesta fall bara att skriva in dem. Maple känner igen α (α), β (β), λ (λ), ω (ω), ϕ (ϕ) och så vidare. För de stora bokstäverna inleder man bara med stor bokstav: Δ (Δ), Ω (Ω), Γ (Γ) och Φ (Φ) är vanliga. Dock är vissa namn upptagna, till exempel χ och π . Vill man ändå mata in den bokstaven kan man ta till den grafiska inmatningen som beskrivs nedan.

¹ Den långa formen har fördelen att man kan skriva $f(4)$ för att beräkna värdet i en viss punkt, men den går till exempel inte att plotta med $\text{plot}...$ -funktionen från popupmenyn.

Grafisk inmatning

Maple 10 ger som sagt mycket bättre stöd för “textfri” inmatning än tidigare versioner. De flesta operationer som t.ex. integrering, ekvationslösning eller derivering kan göras genom att högerklicka på ett uttryck och välja ett lämpligt menyval. Kapitlen om de olika matematiska områdena har tabeller över vilka kommandon som motsvaras av vilket menyval.

Därtill har Maple något som kallas för *paletter*. På vänster sida finns ett antal paletter. De intressantaste är *Expression*, *Common Symbols*, *Matrix* och *Greek*. Om man faller ut *Expression* kan man sätta in färdiga mallar för integraler, summor, differentier, rotuttryck, absolutvärden, n över k , styckvisa funktioner... När mallen är insatt kan man antingen klicka på symbolen man vill ändra, eller trycka på tab-tangenten (till vänster om Q) och shift+tab för att bläddra sig igenom de olika mallvärdena och ändra dem som man vill. Paletten *Matrix* beskrivs närmare i kapitlet om Algebra.



En av paletterna heter *Symbol Recognition*. Även om den inte är ofantligt användbar är den ändå ett finurligt verktyg, och kan komma till pass om man vet vilket tecken man vill skriva in, men inte vad det heter. Genom att rita med musen i ritfältet kan man skissa vad man är ute efter, och sedan klicka på den högra knappen (med två π -tecken). Maple gissar vilka symboler du kan ha menat² och föreslår två-tre symboler. Är den inte med bland förslagen kan man trycka på *More* för att få fler förslag. Klicka på symbolen för att sätta in den i arbetsbladet.

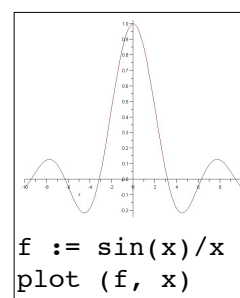
Paket

Maple håller inte all funktionalitet inladdad direkt när man börjar använda det. Det går nämligen att bygga ut Maple med samlingar av funktioner och kommandon som kallas för paket. För de matematiska grundkurserna finns bara på ett fåtal användbara paket: *LinearAlgebra*, *CurveFitting* och *plot*. Innan man kan använda någon av funktionerna i paketet måste man ladda in det. Det görs med kommandot `with()`, till exempel `with(LinearAlgebra)`: Notera kolonet efter `with`. Detta är en rent estetisk åtgärd, så att `with`-kommandot inte skriver ut alla funktioner som hämtas in. Det kan vara ganska mycket utmatning, så det kan vara bekvämt att tysta Maple. Dock kan det vara intressant att se vilka funktioner som faktiskt finns, och bli inspirerad att pröva ett annat sätt att ta sig igenom en uppgift.

Enkel grafitning

Medan man arbetar med matematik för hand rör man sig ofta i ett ganska abstrakt område – det kan vara svårt att se ett funktionsuttryck framför sig, eller hur en derivata förhåller sig till sin funktion. Att rita upp en graf för hand är visserligen ett utmärkt sätt att öva sig i att tolka funktionsuttryck, men det är inte alltid helt trivialt och kanske inte direkt något man gör efter varje löst analysuppgift. Datorer är däremot bra på att göra enkla saker snabbt, och att rita grafer är inget undantag.

På engelska säger man “plotting” för grafitning, och så gör även Maple. Den enklaste funktionen som beskrivs här heter helt enkelt `plot`. Den räcker antagligen till de flesta tillämpningar, och är lätt att använda. I bästa fall kommer man nämligen undan med ett vanligt `plot(f, x)` där f är uttryck av en variabel. (För mer avancerade plottar eller plottar av flera variabler, se avsnittet *Plotting*).

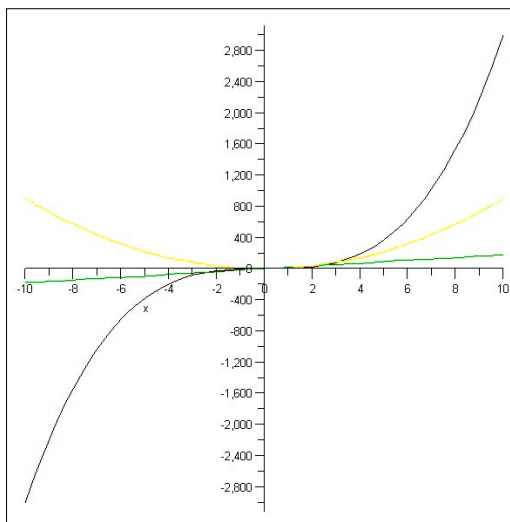


² Med hjälp av minsta kvadratmetoden, om någon är intresserad.

Vill man rita flera funktioner i samma plot anger man dem som en mängd. Ett bra exempel när man börjar med analys är att rita upp en funktion och dess första- och andraderivata:

```
f := 3x^3 + sin(2x);           # Definiera ett uttryck
fprim := diff(f, x);          # Beräkna dess derivata
fbis := diff(fprim, x);       # Beräkna andraderivatan
plot({f, fprim, fbis}, x);
```

(Bekymra dig inte om diff ännu – det är ett kommando som deriverar en funktion.)



Är det alltså bara att mata in en funktion i Maple, plotta den, rita av på papper och lämna in? Nej, man måste själv beräkna bra intervall att rita grafen över. Maple kan göra kvalificerade gissningar, men de är inte att lita på om funktionen inte är trivial. Istället rekommenderas att man tänker på hur funktionen uppför sig: är den periodisk? hur många rötter bör den ha, och syns de alla på bilden? kan andraderivatan orsaka en lokal max- eller minpunkt? Man kan då lägga till nya gränsvärden i plot-anropet. Man använder då inmatningsformen för intervall: $x=-3..3$ för ett intervall $[-3, 3]$ för x .

```
plot(x^2, x = -3..3);          # Begränsa x-axeln till -3..3
plot(x^2, x = -3..3, y = -3..3); # Begränsa även y-axeln till -3..3
```

ALGEBRA

Ekvationslösning

En ekvation i Maple är ett uttryck som har ett vänsterled och ett högerled, precis som vanligt. Man skriver att ett visst uttryck är lika med ett visst annat uttryck:

```
2 * x = 6;  
2x + 4 = x / 2;  
a := 5;  
2 * x = 4 * a;
```

Ekvationer kan innehålla tidigare namngivna variabler (i ovanstående exempel `a := 5`), vilket kan vara mycket praktiskt om man vill pröva sig fram i en ekvation, eller om man har inlämningsuppgifter där ingående variabler till exempel är ditt personnummer. I så fall kan man namnge variabler som så:

```
a := 2; b := 5; c := 2;  
a^2 - sin(b * Pi) = c;
```

...och på så sätt kunna skriva in `a`, `b`, och `c` i alla ekvationer i hela uppgiften, istället för att behöva skriva in värdena överallt. Om man arbetar tillsammans med någon annan kan man på så sätt bara byta ut värdena i variablerna och köra om programmet, istället för att behöva skriva om allting.

Vill man tilldela ekvationen ett namn gör man som vid vanlig tilldelning:

```
ekv := 2 * x = 6;
```

Här syns tydligt varför Maple envisas med att använda `:=` för tilldelning och `=` för likhet. Om man inte skiljde på dem skulle Maple ha svårt att tolka ovanstående uttryck.

För att få Maple att lösa ekvationer använder man kommandot `solve`. För att kunna göra det på enklaste möjliga sätt behövs bara en ekvation. Den kan ges direkt i anropet till `solve` eller som en separat ekvation. Anropet löser funktionen algebraiskt:

```
solve(x^2 - x - 3 / 4 = 0);  
-1/2, 3/2  
f := x^2 - x - 3 / 4;  
solve(f = 0);  
-1/2, 3/2
```

Ger man inget högerled till `solve` utgår lösaren från att man söker rötterna till funktionen, dvs $x = 0$. Vill man komma åt en särskild lösning att fortsätta arbeta med väljs den i menyn `Select Element`.

Vissa funktioner går dock inte att lösa algebraiskt. Ekvationen $\sin(x) = x/2$ har ingen algebraisk lösning, och Maple svarar därför med uttrycket `RootOf`:

```
solve(sin(x)=x/2);  
RootOf(_Z-2sin(_Z))
```

`RootOf` är en speciell symbol i Maple som helt enkelt representerar alla lösningar av ett uttryck. I de flesta fall innebär det att man har stött på något som inte går att lösa algebraiskt, och att man istället måste lösa det numeriskt. Använd istället kommandot `fsolve`:

```
fsolve(sin(x)=x/2);  
0.
```

`fsolve` löser en ekvation numeriskt. Maple svarar att lösningen är `0.` - decimalpunkten visar att svaret är numeriskt, och alltså inte nödvändigtvis exakt. Ritar man upp funktionerna $\sin(x)$ och $x/2$ ser man dock att det finns två lös-

ningar till. `fsolve` börjar nämligen leta efter lösningen i $x = 0$, och hittar den direkt. För att komma runt problemet måste man ge `fsolve` lite hjälp, genom att avgränsa vilka områden den ska leta i. Dessa områden hittar man lätt med hjälp av en plot (Se kapitlet Plottning).

För att lösa numeriskt i ett visst intervall lägger man till intervallet efter ekvationen.

`fsolve(sin(x) = x/2, x = -3..-1)` instruerar `fsolve` att söka i intervallet $[-3..-1]$. Intervall skrivs som vanligt med den vänstra gränsen, två punkter och den högra gränsen. För att hitta alla lösningar till $\sin(x) = x/2$ måste man alltså söka alla tre lösningar separat:

```
fsolve(sin(x) = x/2, x=-5..-1);
-1.895494267
fsolve(sin(x) = x/2, x=-1..1);
0
fsolve(sin(x) = x/2, x=1..5);
1.895494267
```

Detta är ett exempel på att Maple *inte* ersätter matematiskt kunnande - bara beräkningar! Man måste alltid kontrollera sina svar och att de verkar rimliga!

Ekvationer i flera variabler kan naturligtvis också lösas i en av dem. I många uppgifter måste man uttrycka en okänd variabel i en annan, som till exempel $2 * a + 3 = 5 * x$. Ber man Maple göra detta med `solve` eller `fsolve` får man ut den ena variabeln löst med avseende på den andra, och den andra variabeln löst med avseende "på sig själv". Ett exempel är $x = 3a$, $a = a$. Pröva! För att få ett mer användbart svar kan man be Maple att lösa med avseende på en viss variabel. I exemplet nedan löses med avseende på a :

```
solve(2 * a + 3 = 5 * x, a);
-3/2 + 5*x/2
```

(Formen är precis densamma för att ge ett lösningsintervall, fast utan intervall)

Ekvationssystem

För att skriva in ett ekvationssystem skriver man de ingående ekvationerna mellan "måsvingar", till exempel $\{x+y=1, x-y=0\}$. Om man redan har en uppsättning ekvationer som man vill sätta in i ett system kan man bara skriva in ekvationernas namn istället: $\{ekv1, ekv2, ekv3\}$.

```
solve({3x + 2y = 3, x - y = -4});
{x=-1, y=3}
e1 := x + y + z = 1:
e2 := 3 * x + y = 3:
e3 := x - 2 * y - z = 0:
solve({e1, e2, e3});
y=3/5, z=-2/5, x=4/5
```

Det går naturligtvis också bra att blanda formerna och skriva till exempel `solve({e1, e2, x - 2 * y - z = 0});` istället för den sista raden i exemplet ovan. Man kan självklart tilldela ett namn till ett ekvationssystem:

```
systemet := {2*x=4, x-y=3};
solve(systemet);
```

Tänk dock på att namnet `system` är upptaget av Maple. Det går alltså inte att döpa ett ekvationssystem till `system`. Dock går `systemet` bra.

Om systemet saknar lösningar svarar Maple med "inga lösningar" – Maple svarar alltså inte alls. Prova att lösa ekvationssystemet $3x + 3 = 3x -$ inget svar ges. Om systemet däremot har ett oändligt antal lösningar får man ett svar om hur variablerna är relaterade till varandra.

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

Detta skiljer sig något från den form som oftast lärs ut i kurserna i linjär algebra, där lösningarna istället beror på en "fri variabel", exempelvis t . För att övergå på den formen inser man att det bara är att ersätta x med t . För att göra ersättningen med Maple, se `subs()` nedan.

Förenkling, utveckling, faktorisering och substituering

Sitter man med en ekvation och vill forma om den (till exempel flytta högerledet till vänster sida, dividera bägge sidor med $2x$, eller utveckla en kvadrat) kan man göra det med Maples hjälp. Detta är bra sätt att kontrollera sina egna beräkningar om man kört fast i handräkning: man matar bara in samma steg man själv gjort i Maple, tills Maples svar skiljer sig från det man förväntade sig.

Manipulate Equation är ett miniprogram inuti Maple som man kan använda för att interaktivt förändra eller lösa ekvationer. Det startas via menyn `Tools` → `Assistants` → `Equation Builder`. **Hittar man inte Manipulate Equation** behöver man uppdatera Maple. Detta görs genom att besöka <http://www.maplesoft.com> och klicka sig fram till `Customer Support` → `Downloads and Service Packs` och ladda ner den senaste uppdateringen (i skrivande stund version 10.06).

Överst i fönstret ser man ekvationen och vilka steg som gjorts, vilket även visas i rutan `History`. I den undre hälften av fönstret (`Operations`) finns alla operationer man kan tänkas vilka göra på sin ekvition.

- **Addition:** används för att lägga till eller dra ifrån på bägge sidor av ekvationen. Menyn ger förslag på uppenbara termer att addera. Man kan också välja `Group terms on left/right side` för att sätta höger/vänster led lika med noll. För att dra ifrån från ekvationen adderar man en negativ term.
- **Multiplication:** används för att multiplicera bägge led med en faktor. Menyn ger förslag på uppenbara faktorer. Man kan också välja `Clear denominators` för att korsmultiplicera bort alla nämnare i uttrycket.
- **Power:** används för att upphöja bägge sidor i en viss exponent. Det finns snabbknappar för kvadrering och kvadratrottagning. `Exponentiate both sides using base n` skriver om bägge led som n upphöjt till ledet.

- **Apply function:** används för att sätta in bägge led i en funktion, till exempel att ta naturliga logaritmen av bägge led eller att ta arccos av bägge led.
- **Simplify:** Maple har en mängd inbyggda förenklingar, till exempel, parentesutveckling, faktorisering/utbrytning och normalisering av kvoter.

Förenkling görs med `Simplify`. Det dock är inte alltid Maples uppfattning om vad som är "enklast" är det man väntade sig, och då är det läge att göra om det för hand. Dock är det ett jättebra sätt att få hjälp med förenklingsuppgifter i boken. Har man fått ett svar som inte stämmer med facit kan man som vanligt mata in sina egna steg i `Simplify` ett och ett och se efter var Maples svar börjar skilja sig från `Simplify(facits svar)`. Då har man hittat stället där man gjorde fel.

$$\cos(x) * (\cos(x) + \sin(x)^2/\cos(x)) \rightarrow 1$$

Utveckling görs med `Expand`. Maple omvandlar uttrycket till ett vanligt polynom.

$$(2*x - 4)^2 + (3*x + 3) \rightarrow 4x^2 - 13x + 19$$

Faktorisering och utbrytning görs med `Factor`. Uttrycket omvandlas till en produkt av summor (så gott det går).

$$6 * x^2 + 18 * x - 24 \rightarrow 6(x+4)(x-1)$$

Reduktion av kvoter görs i de flesta fall automatiskt, men det går att tvinga Maple om det missar en kvot. Man använder då `Normal`.

$$144/36 \rightarrow 4$$

När ekvationen är färdigarbetad sätter man tillbaka resultatet med knappen `Return` `Result` längst ner.

För att substituera ett uttryck med ett annat kan man använda textkommandot `subs`. Den går att använda dels för att ersätta deluttryck, men också för att sätta in värden i en ekvation utan att behöva skapa en helt ny variabel. Funktionen är lika lätt att använda som `simplify`, `factor` och `expand` men kan göra lite mer. Man kan ersätta hela uttryck med andra, något som är användbart till exempel när man arbetar med knepigare integraler.

```
subs(x = 2, x^2 + x + 1)
7
subs(a = sin(x), cos^2 + a^2);
1
```

Som tidigare nämnts kan man använda `subs` för att sätta in värden i en ekvation, vilket kan vara bekvämt om man redan har skrivit ekvationen. Istället för:

```
ekv:=x^2 - 4 = 2*a;          # Skriv in ekvationen
ekv:= x^2 - 4 = 2a
a:=4;                       # Just det, värdet ska in i a också...
ekv:=x^2 - 4 = 2*a;        # Skriv in ekvationen igen...
ekv:=x^2 - 4 = 8
```

...kan man komma undan med:

```
ekv:=x^2 - 4 = 2*a;          # Skriv in ekvationen
ekv:= x^2 - 4 = 2a
subs(a=4, %);               # Just det, värdet ska in i a också...
ekv:=x^2 - 4 = 8
```

Alla dessa operationer finns både som textkommandon och menyval. Är man kvick på tangentbordet eller bara vill göra en snabb förenkling utan att behöva starta `Manipulate Equation` kan de vara till nytta.

Funktion	Popup-menyval	Kommando
Förenkling av f	Simplifications→Symbolic	<code>simplify(f)</code>
Utveckling av f	Expand	<code>expand(f)</code>

Funktion	Popup-menyal	Kommando
Faktorisering av f	Factor eller Complete Square	factor(f)
Korsmultiplikation av f	Cross-Multiply	Finns ej
Substitution av s till t i f	Finns ej	subs(s = t, f)

“Kan man få Maple att skriva ut vilka steg som görs, för att se hur förenklingen går till?” En vanlig fråga från studenter, och svaret är "ja, men nej". Man *kan* få Maple att skriva ut stegen, men stegen som Maple gör är inte desamma som människor skulle göra, och svaret är näst intill oläsligt. Utskriften av lösningsvägen till `simplify(cos(x) * (cos(x) + sin(x)^2/cos(x)))` är 13 sidor tolvpunkterstext. Det går att få hjälp av utskriften, genom att se vad mellanresultaten är, men det är knappast värt besväret. Tillvägagångssättet är dock enkelt: `printlevel := 100;` ställer in Maple på att noggrant logga allt som händer (i allt du gör). Skriv `printlevel := 0;` för att återgå till det vanliga läget.

Komplexa tal

Komplexa tal uttrycks i Maple precis som vanligt, men man använder I (stora i) för den imaginära enheten.

```
2 + 4I + 2 - I + 3 * (5 + 9I);
19 + 30I;
a := 5: b := 3: c := a + b * I;
5 + 3I;
```

De vanliga operationerna på komplexa tal är att hämta ut real- och imaginärdelen av talet, konjugera, övergång till polär form, samt att beräkna beloppet och argumentet.

Funktion	Popup-menyal	Kommando
Realdelen av Z	Real Part	Re(Z);
Imaginärdelen av Z	Imaginary Part	Im(Z);
Konjugatet av Z	Conjugate	conjugate(Z);
Beloppet av Z	Modulus	abs(Z);
Argumentet av Z	Argument	argument(Z);
Z till polär form	Conversions → Polar	polar(Z);

Komplexa ekvationer löses som vanligt. Mata in en komplex ekvation och gör menyalvalet `Solve`.

LINJÄR ALGEBRA

Maple har mycket bra stöd för att arbeta med linjär algebra. Det finns ett äldre paket som heter `linalg`, och ett nyare som heter `LinearAlgebra`. Det nyare paketet kan samma saker som `linalg`, men det är definitivt lättare att arbeta med. Felmeddelandena är bättre, och matrisaddition och multiplikation kan göras med de vanliga `+`, `-`, `*` (komponentvis multiplikation) och `.` (skalärprodukt, matrismultiplikation). Den här handledningen går endast igenom `LinearAlgebra`.

För att kunna använda `LinearAlgebra`-paketets textkommandon måste man ha utfört `with(LinearAlgebra)`: någons i arbetsbladet. Använder man menyvalen behövs det inte.

Skapa matriser och vektorer

För att skriva in matriser och vektorer i Maple bör man använda `Matrix`-paletten. Fäll ut den genom att klicka på den knappen till vänster.³ Fyll i matrisens storlek i rutorna `Rows` och `Columns`, eller använd knappen `Choose` för att snabbt välja matrisens dimensioner. Vill man själv fylla i matrisens värden väljer man `Custom values` från menyn `Type`. I annat fall kan man skapa nollmatriser och identitetsmatriser från den menyn. Från `Shape` kan man välja en viss form på matrisen, det vill säga diagonaliserad, symmetrisk eller liknande. Menyn `Data type` är i stort sett oväsentlig. Klicka sedan på `Insert Matrix` för att sätta in matrisen i arbetsfönstret. (Om knappen är grå måste du klicka i textfönstret för att visa var du vill ha matrisen.) Om man valde `Custom values` som `Type` kan man nu bläddra igenom elementen med `tab`-knappen och `skift+tab`. Vill man tilldela den färdiga matrisen ett namn görs det lättast med popupmenyvalet `Assign to a Name`.

Matrisoperationer

När man har konstruerat matriserna kan man börja arbeta med dem. Addition och multiplikation görs med `+`-tecknet och `.`-tecknet (plus och punkt).

$$\begin{bmatrix} 1 & 2 \\ -1 & 0 \\ 3 & 1 \\ -1 & 3 \end{bmatrix} \rightarrow A := \begin{bmatrix} 1 & 2 \\ -1 & 0 \\ 3 & 1 \\ -1 & 3 \end{bmatrix}$$
$$A + B \quad \begin{bmatrix} 4 & 3 \\ -2 & 3 \end{bmatrix}$$
$$A \cdot B \quad \begin{bmatrix} 1 & 7 \\ -3 & -1 \end{bmatrix}$$

Gausselimination och radoperationer

Maple kan utföra Gausselimination såväl som enstaka radoperationer. För att göra en komplett lösning av ett system använder man popupmenyvalet `Solvers and Forms` \rightarrow `LU Decomposition` \rightarrow `RREF`, vilket är en tungvrickare. `RREF` står för `Reduced-Row Echelon Form`, vilket motsvarar Gauss-Jordans metod. Detta förutsätter dock att matrisen är byggd så att högraste kolumnen utgörs av ekvationernas högerled, det vill säga att matrisen är systemets totalmatris.

$$C \rightarrow \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 5 \end{bmatrix}$$
$$\rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 7 \\ 0 & 1 & -2 \end{bmatrix}$$

`RREF` matar ut en lista med olika resultat, där de tre första är delresultat, och den fjärde är den reducerade matrisen. För att komma åt den högerklickar man på resultatlistan och väljer `Select Element` \rightarrow 4.

³ Man kan skriva in matriser för hand, men det är tidsödande och lätt att göra fel. Se ?Matrix.

Dock är det inte det minsta ovanligt att man har ett ekvationssystem i en matris, och en kolonnvektor med lösningsvärden. Ekvationen skrivs då $Ax = b$. Här saknas ett menyval för lösningen (`solve` går bet på matrisen) så man måste ta till textkommandot `LinearSolve`. `LinearSolve` är en del av paketet `LinearAlgebra`, så först måste man ladda paketet med `with(LinearAlgebra):`. Sedan löses ekvationen med `LinearSolve(A, b)`.

Determinanter, inverser, egenvärden och egenvektorer

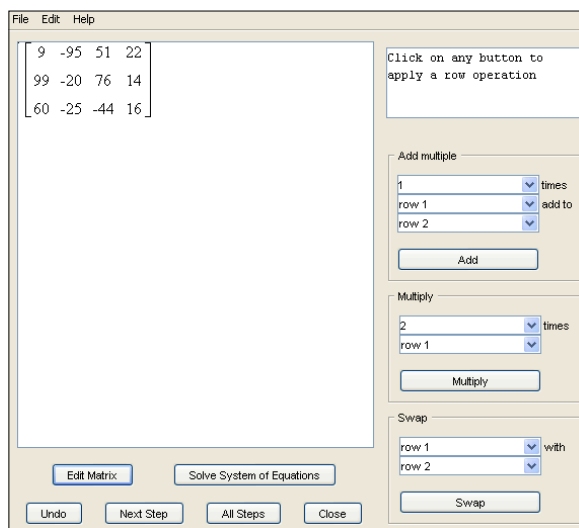
Maple kan beräkna determinanter, inverser och egenvärden och egenvektorer via popupmenyn.

Funktion	Popup-menyal	Kommando
Determinanten för A	Standard Operations → Determinant	Determinant(A)
Invertering av A	Standard Operations → Inverse	MatrixInverse(A)
Transponering av A	Standard Operations → Transpose	Transpose(A)
Hitta A:s egenvärden	Eigenvalues, etc → Eigenvalues	Eigenvalues(A)
Hitta A:s egenvektorer	Eigenvalues, etc → Eigenvectors	Eigenvectors(A)

Det finns ett par hakar med ovanstående funktioner. Får man ett långt felmeddelande efter att ha försökt beräkna en matris determinant beror det i de flesta fall på att matrisen inte är kvadratisk. Får man ett felmeddelande efter att ha beräknat matrisens invers som talar om *singular matrix* saknar matrisen invers. Värt att påpeka är att Maple inte klagar om den matrisen som ska inverteras inte är kvadratisk. Då får man istället ut en pseudoinvers, något som inte ingår i de inledande kurserna. Resultatet från `Eigenvectors` innehåller både egenvärdena och egenvektorerna insatta i en `mat→ris`. Vill man arbeta vidare med egenvektorerna använder man menyvalet `Select Element → 2`.

Vill man öva på Gausselimination kan man använda studentstödet `Gauss-Jordan Elimination`. Det finns i huvudmenyn under `Tools → Tutors → Linear Algebra → Gauss-Jordan Elimination`. Man får ett system med tre okända, och ganska höga värden.

Här kan man sedan träna på strategier för att lösa system genom att välja `Add multiple` (lägg multipel av en rad till en annan), `Multiply` (multiplicera en rad med en konstant) eller `Swap` (byt plats på två rader). Har man en färdig matris man vill öva på klickar man på `Edit Matrix` – detta fönster kan också användas för att se det linjära systemets ekvationer.



ANALYS

Att arbeta med matematisk analys i Maple görs på samma sätt som andra uppgifter – man matar in ekvationer och uttryck och använder Maples inbyggda funktioner för att derivera, integrera och lösa differentialekvationer. Maple kan också hantera flervariabeluttryck, differentialekvationer och göra serieutvecklingar.

Derivator

En funktion deriveras genom att den matas in i Maple på vanligt sätt, och sedan använder man kommandot `Differentiate` eller `diff`. För att bara beräkna en enkel derivata rekommenderas att använda menyvalet `Differentiate`, men för uppgifter som är svårare än så behöver man använda `diff`.

Skriv in en funktion, låt säga $x^3 - 2x^2 + (1/x) + 3$ eller $2 * \cos(a) - \tan(a)$. Använd menyvalet `Differentiate` för att derivera uttrycket.

Textkommandot `diff` kan göra mer än menyvalet `Differentiate`. `diff(f, x)` deriverar f med avseende på x , där x är någon av de ingående variablerna. Man kan också få den n :te derivatan direkt, genom att lägga till fler variabler: `diff(f, x, x, x)`; ger tredjederivatan av f .

Partiella derivator kan göras genom att lista i vilken ordning man vill derivera uttryckets variabler. Givet funktionen för konens volym kan man derivera med avseende på höjden eller radien.

```
V := (r^2 * h * Pi)/3
diff(V, r)
diff(V, h)
                2/3rhπ      # (dV/dr)
                1/3r^2*π    # (dV/dh)
```

Andraderivatorna kan fås direkt på samma sätt som för envariabelfunktion, men man kan också lägga till andra variabler. `diff(V, r, h)` motsvarar andraderivatan av volymen med avseende på radien och höjden.

Funktion	Popup-menyal	Kommando
Derivatan av f m.a.p x	Differentiate → x	<code>diff(f, x)</code>
Andraderivatan av f med avseende på x	Upprepa ovanstående	<code>diff(f, x, x)</code> ;
Partiell andraderivata av $f(x,y)$ med avseende på x följt av y	Upprepa ovanstående med y som variabel	<code>diff(f, x, y)</code> ;

Integraler

En funktion integreras på samma sätt som den deriveras - i det enklaste fallet. Dock finns integrationsintervall, integrationskonstanter och generaliserade integraler att tänka på.

Den enklaste integrationen att utföra i Maple är en obestämd integral. Då kan man nämligen använda menyvalet `Integrate` eller kommandot `int(f, x)` för att integrera f med avseende på x .

```
int(2x + 5)
                x^2 + 5x
```

En sak som är viktig att komma ihåg när man ber Maple att beräkna en obestämd integral är att integrationskonstanten inte skrivs ut. Glöm alltså inte att skriva $x^2 + 5x + C$ i ditt svar.

För bestämda integraler kan man använda menyvalet `Definite Integral` eller textkommandot `int`. Man anger då vilken funktion som ska integreras, med avseende på vilken variabel, och över vilket intervall. Menyvarianten används genom att konstruera den bestämda integralen innan man beräknar den. Välj från popupmenyn kommandot `Constructions → Definite Integral`, och mata in gränserna. Använd sedan popupmenyvalet `Evaluate` på resultatet. Textkommandot används som vanligt, men med integrationsgränserna (till exempel $[0, 5]$) angivna som nedan.

```
f := x^2+5;
int(f, x=0..5)
```

200/3

Funktion	Popup-menyal	Kommando
Obestämd integral av f	Integrate	int(f, x)
Bestämd integral av f på intervall [a, b]	Constructions → Definite Integral, följt av Evaluate	int(f, x=a..b)

Styckvis definierade funktioner

Att definiera styckvisa funktioner i Maple är lätt att göra med en mall, men knepigt att göra med text. Mallen finns längst ner på Expression-paletten och heter `piecewise`. Använd tab-tangenten för att bläddra mellan de olika delarna av funktionen. Det går bra att sätta in en ny styckvis funktion i den förra för att bygga upp större uttryck. Vill man ha snyggare styckvisa funktioner med mer än två villkor kan man dock använda textkommandot `piecewise`. Funktionen tar emot ett villkor, ett uttryck, ett villkor, ett uttryck och så vidare. Till exempel:

```
piecewise(x < 0, abs(sin(x)), x > 0, sqrt(x));
```

Styckvisa funktioner kan deriveras och integreras precis som vanligt. Om funktionen är diskontinuerlig rapporterar Maple att derivatan är `undefined` i den punkten.

Gränsvärden

Att beräkna gränsvärden för hand är en uppgift som kräver en del eftertanke, erfarenhet och kunskap om vilka substitutioner som kan och bör göras. Det är naturligtvis svårt att ersätta den sortens färdigheter med ett datorprogram, och man får inte så mycket hjälp av Maple – Maple ger ett färdigt svar och ingen information om hur man måste resonera. Icke desto mindre går det att använda Maple för att kontrollera sina svar och prova sig fram.

Gränsvärdesberäkningar i Maple görs antingen genom att välja `Constructions → Limit` från menyn, mata in vilket värde variabeln går mot, och sedan ta `Evaluate` på resultatet. Man kan också använda textkommandot `limit`:

```
limit(sin(x), x = 0);
```

1

För gränsvärden där den oberoende variabeln går mot oändligheten kan man ange `infinity` eller `-infinity`, både i menyvarianten och textkommandot.

```
limit(1+1/x, x=infinity);
```

1

Behöver man beräkna vänster- eller högergränsvärden måste man använda textkommandot, och ange `left` eller `right` som tredje argument:

```
limit(arctan(1/x), x=0, left);
```

$-\pi/2$

```
limit(arctan(1/x), x=0, right);
```

$\pi/2$

Gränsvärden i flera variabler kan också lösas. Textkommandot `limit` måste användas, och man anger helt enkelt en lista av värden som de ingående variablerna ska gå mot:

```
limit(x+3/y, {x=0, y=3})
```

1

Differentialekvationer

För att lösa differentialekvationer med Maple behöver man veta hur man matar in både ekvationer och derivator. En differentialekvation är, matematiskt sett, en ekvation som innehåller funktionens derivata, och svårare än så är det inte i Maple heller. $y'' - 6y' - 7y = 0$ skrivs som `diff(y(x), x, x) - 6 * diff(y(x)) - 7y(x) = 0`

Notera att funktionen måste skrivas som $y(x)$ eller liknande - den beroende variabeln måste skrivas uttryckligen för att Maple ska veta vilken/vilka variabler som ingår i den okända funktionen. Var noga med detta, för om Maple inte kan tolka hela uttrycket kommer du inte att få några svar från lösaren.

Lösning av ordinära differentialekvationer görs med menyvalet `Solve DE` eller `dsolve`. `dsolve` tar in en differentialekvation (som vanligt kan du mata in namnet på en tilldelad ekvation, eller ekvationen direkt) och vilken funktion och vilken variabel som ska lösas för:

```
DE := diff(y(x), x) = x;          # Ekvationen y' = x
dsolve(DE, y(x));
y(x) = 1/2 x^2 + _C1
```

Notera att svaret innehåller konstanten `_C1`. Alla svar som börjar med ett lågstreck '_' är så kallade "placeholders" - värden som kan anses som okända konstanter. Detta är precis som det svar du skulle fått om du räknat för hand - en okänd konstant C. Om det finns fler okända värden i svaret numreras de `_C1`, `_C2`, `_C3` och så vidare, istället för sättet som oftast används vid handräkning: C, D, E...

Får man bara tillbaka det man skrev in när man matade in ekvationen (svaret blir till exempel `dsolve(ekvationen)`) är det vanligast för att man glömt att skriva in vilken funktion man vill lösa med avseende på: `dsolve(diffekv)` blir fel, `dsolve(diffekv, y(x))` blir rätt. En annan vanlig anledning är att man glömt att skriva $y(x)$ överallt där funktionen y förekommer, t.ex. på funktionen själv. Se även kapitlet Felsökning.

En ordinär differentialekvation löses precis som ovan:

```
ode := diff(y(x), x) + y(x) = 2; # Ekvationen y' + y = 2
dsolve(%, y(x));
y(x) = 2 + exp(-x) _C1
```

Notera att konstanten `_C1` nu är en koefficient till e^{-x} i partikulärlösningen, precis som sig bör.

Högre ordningens differentialekvationer löses genom att skriva in högre ordningens derivator i ekvationen. Utöver det är det inget konstigt - under förutsättning att ekvationen kan lösas analytiskt.

```
ode := diff(y(x), x, x) = 2 * cos(x) # Ekvationen y'' = 2cos(x)
dsolve(ode, y(x));
y(x) = -2 * cos(x) + _C1 * x + _C2
```

Med högre ordningens ekvationer får man fler konstanter, i det här fallet `_C1` och `_C2`.

Begynnelsevärdesproblem går att lösa på ungefär samma sätt. Istället för att sätta in differentialekvationen och vilken funktion man vill lösa den i, skickar man differentialekvationen och en serie begynnelsevärden, och sedan vilken funktion man vill lösa den i. Ekvationen och villkoren sätts mellan måsvingar. I praktiken skickar man en lista med differentialekvationen först, och villkoren efteråt.

```
ode := diff(y(x), x) + y(x) = 2; # y' + y = 2
bv := y(0)=4;
dsolve({ode, bv}, y(x));
y(x) = 2 + 2* exp(-x)
```

Högre ordningens differentialekvationer kräver begynnelsevärden derivator. Dessa måste ges som ett värde på någon av derivatorna (till exempel $y(0) = 1$ och $y'(0) = 0$). Här kan man inte använda `diff` för att uttrycka derivatan, utan

måste använda operatoren D . Detta på grund av att man inte kan uttrycka att derivatan i en viss punkt ska ha ett värde med hjälp av diff . (Det går inte att skriva $\text{diff}(y(0), x) = 1$ eller $\text{diff}(y(x), 0) = 1$).

Första ordningens derivata i $x = 0$ uttrycks istället som $D(y)(0)$, ("D, av y, av noll") och ett begynnelsevillkoret $y'(0) = 2$ uttrycks som $D(y)(0) = 2$;

```
ode := diff(y(x), x, x) = y(x)    # Ekvationen y'' = y
bv := y(0)=1, D(y)(0)=0          # y(0)=1, y'(0)=0
dsolve({ode, bv}, y(x))
y(x) = 1/2*exp(x)+1/2*exp(-x)
```

Andra- och tredjederivator uttryckta i $D()$ skrivs (enklast) genom att kedja ihop D -operatorer. Till exempel kan begynnelsevärdet att $y''(0) = 3$ skrivas $D(D(y))(0) = 3$. (Kedja alltså ihop D -operatorer tills du har rätt derivata, sätt in vilken funktion som avses, i vilken punkt, och sist likhetstecken och värdet.

```
ode := y(x)+(diff(y(x), x, x)) = 0    # Ekvationen y + y'' = 0
bv := y(0) = 0, D(D(y))(0) = 0        # y(0) = 0, y''(0) = 0
dsolve({ode, bv}, y(x))
y(x) = _C1*sin(x)
```

Randvärdesproblem löses dessbättre på samma sätt. Mata bara in ett värdet för villkoret i en annan punkt:

```
ode := diff(y(x), x) = 1;
rv := y(3) = 5;
dsolve({ode, rv}, y(x));
y(x) = x + 2
```

Partiella differentialekvationer

Det är inte alltid Maple kan lösa ekvationerna, och svaren kan vara svårlästa. Icke desto mindre använder man sig av samma arbetsgång: mata in ekvationen tillsammans med eventuella villkor i funktionen pdsolve :

```
pde := u(x, y) = diff(u(x, y), x) + diff(u(x, y), y)
pdsolve(pde, u(x,y))
u(x, y) = _F1(y - x) * exp(x)
```

...där $_F1$ är en godtycklig funktion på samma sätt som $_C1$ är en godtycklig konstant.

Att separera variablerna är i de flesta fall en bättre taktik än att hoppas på att pdsolve ska göra jobbet tillräckligt bra. Skriv in ekvationen, och använd $\text{subs}()$ (se kapitlet Algebra) för att substituera funktionen $u(x, y)$ med $X(x)Y(y)$:

```
pde := diff(u(x, y), x, x) = 4*(diff(u(x, y), y)); # Ekvationen u''xx = 4u'y
subs(u(x,y) = X(x) * Y(y), pde);
```

Resultatet kan vara svårläst på grund av att differentialnotationen skapar en mängd parenteser och d/dx -operatorer, men det är dels korrekt, och dels en bra övning. Använd sedan $\text{Manipulate Equation}$ för att separera variablerna på ett enkelt sätt.

När variablerna väl är separerade är det en smal sak att skriva om den partiella differential ekvationen till två ordinära differentialekvationer som bägge är lika med λ :

```
ode1 := lhs(pde) = lambda;
ode2 := rhs(pde) = lambda;
```

System av differentialekvationer

Att lösa system av differentialekvationer görs i stort sett på samma sätt som lösning av vanliga ekvationssystem: definiera systemet och lös det med `dsolve`:

```
systemet := {diff(x(t), t) = -x(t), diff(y(t), t) = x(t)}
dsolve (systemet);
{x(t) = _C2*exp(-t), y(t) = -_C2*exp(-t)+_C1}
```

McLaurin- och Taylorutvecklingar

Serieutvecklingar är ett ganska abstrakt begrepp som kan vara svårt att se framför sig. Maples utmärkta kapacitet att snabbt plotta funktioner och serieutvecklingar gör det till ett bra verktyg för att ta till sig vad som faktiskt händer vid utvecklingen.

Taylorutvecklingen av en godtycklig funktion $f(x)$ fås genom textkommandot `taylor`. Kommandot tar emot en funktion, en punkt på axeln att utveckla kring, och vilken grad på polynomet man söker.

```
f := 2 * sin(3x)^2;
taylor (f, x = Pi, 5); # Utveckla f(x) kring  $\pi$ , till ett femtegradspolynom
```

Maple saknar ett separat kommando för McLaurinutvecklingar, men eftersom McLaurinutvecklingar enbart är ett specialfall (som är lättare vid handräkning) av Taylorutvecklingar beräknar man McLaurinutvecklingar som en Taylorutveckling kring $x=0$.

```
f := 2 * sin(3x)^2;
taylor(f, x = 0, 5);
```

Flervariabelutvecklingar görs via kommandot `mtaylor` (multivariate Taylor), på samma sätt. Utvecklingspunkten ges som ett talpar, till exempel $\{x=0, y=3\}$:

```
mtaylor(sin(x^2 + y^2), {x = 1, y = 1}, 2)
```

Vill man studera Taylorutvecklingar närmare rekommenderas studentstödet `Taylor Approximation`. Det finns i menyn `Tools`→`Tutors`→`Single variable`→`Taylor Approximation`. Här kan man skriva in ett polynom ("`f(x) =` "), utvecklingspunkt ("`x =` ") och grad på polynomet ("`Order =` "). En plot visas omedelbart där originalfunktionen är röd och polynomet blått. Genom att flytta utvecklingspunkten och/eller ändra polynomets grad är det lättare att få en intuitiv känsla för vad Taylorpolynomet faktiskt representerar. Klickar man på `Animate` ökas graden succesivt så att man ser hur approximationen förbättras.

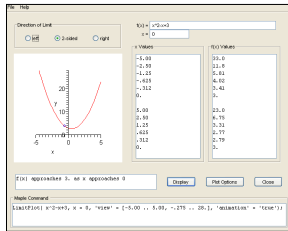
Ett bra tips är att gå in under `Plot Options`, kryssa i "Enable user-defined ranges" och skriva in en bredare x-axel. På så sätt får man en bättre överblick över vart polynomet tar vägen när det inte längre kan hålla sig till funktionen. Ett bra första exempel är att pröva `sin(x)` kring $x=0$, och graden 10. Ställ in att x-axeln ska täcka intervallet $[-10, 10]$ och klicka på `Animate`.

STUDIEHJÄLP I MAPLE

Maple innehåller många bra interaktiva miniprogram ("Tutors") för den som vill träna på vissa begrepp, eller vill fördjupa sin förståelse. Några av dessa miniprogram har demonstrerats i kapitlen om linjär algebra och analys, men det finns många fler. Programmen finns under menyn Tools→Tutors. Mer information om hur de används finns under Help-menyn i varje Tutor.

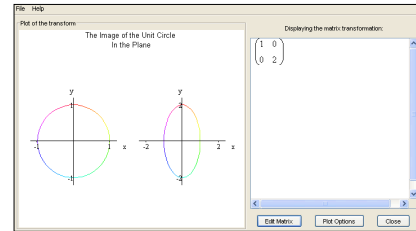
Detta är en lista på några Tutors som kan vara till hjälp i de grundläggande mattekurserna, men titta igenom menyn då och då och se om något kan vara relevant för tillfället.

Studiehjälp för inledande matematikkurser



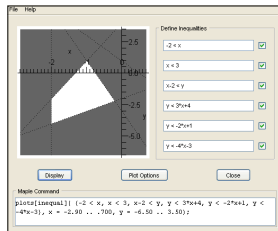
Limits: experimentera med gränsvärden. Man kan skriva in en funktion och en gränspunkt, och se hur funktionen går mot ett visst gränsvärde genom en animerad plot. Värdetabeller för X och Y skrivs upp för närmare studier.

Single-variable→Limits



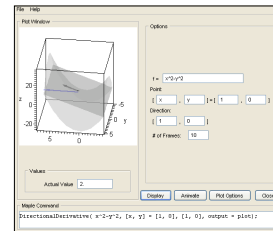
Linear Transform: experimentera med linjära transformationer. Skriv in en transformationsmatrix med Edit Matrix-knappen, och se hur det påverkar en cirkel eller ett klot. Börja med en 2x2-matrix för att få en tydligare bild.

Linear Algebra→Linear Transform



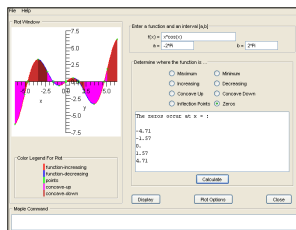
Linear Inequalities: plottar upp resultatet av upp till sex olikheter (endast linjära, tyvärr) och förtydligar precis vilka områden som blir kvar. Skriv in dem en och en och se hur ytan påverkas.

Precalculus→Linear inequalities



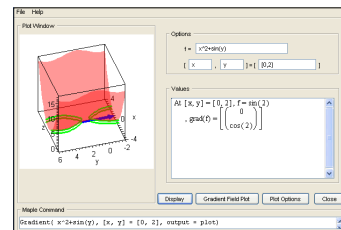
Gradient: illustrerar gradienten på en yta. Man skriver in en funktion av två variabler och evaluerar gradienten i en viss punkt. Den ritas ut med en nivåkurva för funktionens y-värde.

Multi-variable→Gradient



Curve Analysis: plottar en funktion och visar en komplett kurvanalys med markeringar var den är ökande, minkande, har rötter, extrempunkter och inflexionspunkter. Man kan även numeriskt räkna ut i vilka intervall den beter sig på ett visst sätt.

Single-variable→Curve Analysis



Directional Derivative: ritat snabbt upp riktningderivator och tangentplan för en given yta. Funktionen ritas i rött med ett lila tangentplan. Man anger sedan i vilken riktning man vill derivera (grön pil) och derivatans storlek ges som en blå pil.

Multi-variable→Directional Derivative

PLOTTNING

Även om man kan rita upp de flesta grafer man behöver med det vanliga `plot()`-kommandot vill man ofta rita upp mer komplexa samband, till exempel lösningskurvor till differentialekvationer, plan i rymden som lösning till linjära ekvationssystem, parabler i R^3 eller bara ha mer kontroll över färg och text – eller kanske animera någon parameter i lösningen?

Plot Builder

Vill man ha mer kontroll eller slippa mata in gränserna för hand har Maple ett inbyggt miniprogram för att justera plottar. Starta `Plot Builder` via menyn `Tools`→`Assistants`→`Plot Builder`. Ett fönster kommer upp där man kan fylla i de funktioner som ska plottas. I detta fönster måste de skrivas in på tydligast möjliga sätt, alltså till exempel $2*x$ istället för $2x$. Man kan även fylla i namn på namngivna funktioner, eller uttrycka deras derivator som `diff(f, x)` eller liknande. När allt är inmatat klickar man på `Done`. (Det går att återvända till detta fönster om man vill justera något.)

Ett nytt fönster dyker upp med förslag på vilken typ av plot man kan vara intresserad av. (I de flesta funktioner av en variabel fall är 2-D `Plot` rätt, de andra alternativen är för komplexvärda funktioner). Man kan även ställa in värdena för alla ingående variabler här.

För ännu mer kontroll kan man klicka på knappen `Options`. Ytterligare ett fönster kommer upp med alla möjliga inställningar.

Funktioner, variabler och rubriker ställs in i den översta delen av fönstret. Från knappen `Function Selection` kan man återvända till funktionsredigeraren.

Om du tänker plotta mer än en funktion kan du göra separata inställningar för varje funktion genom att välja en funktion från rullgardinsmenyn överst.

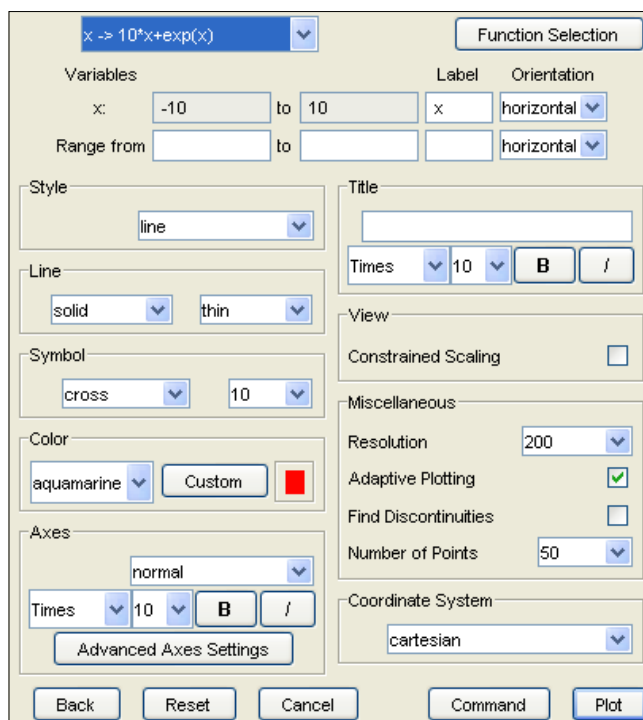
Från rutorna `Variables` på `Options`-fönstret kan man skriva in vilket intervall varje variabel ska plottas över (rutorna på var sida om ordet 'to') och vilken rubrik axeln ska ha (rutan för `Label`). Grafens vertikala axel ställs in med rutorna för `Range` (om man inte lutar på Maples omdöme). Man kan också byta plats på axlarna så att de går horisontellt eller vertikalt.

Ritningsstilen kan ställas om från heldragen linje till punktplot med menyn för `Style`. (Plottar man i 3D får man en uppsjö med olika val här, läs avsnittet om Plottning i 3D för mer information) Väljer man en punktplot kan man från menyn `Symbol` välja hur punkterna ska se ut. Från menyn `Color` kan man antingen välja en förvald färg eller välja en annan färg. (Den här rutan betar sig lite underligt om man har fler funktioner i sin ritning: rullgardinsmenyn kommer inte ihåg vilken färg som hör till vilken graf. Den mins ändå färgen, det är bara menyn som inte återställs ordentligt.)

Figurens axlar kan också ställas in i minutiös detalj, med typsnitt och textstorlek för rubriken, och olika ritningsstilar. Ett bra tips för 3D-plottar är att välja `Boxed` för axlarna, eftersom det blir tydligare var intervallen börjar och slutar.

Hela ritningen kan ges en titel från rutan `Title` tillsammans med typsnitt och textstorlek. Rutan `Constrained scaling` innebär alla axlar i plotten måste ha samma skala. Maple skalar gärna om axlarna så att plotten ser "intressantare" ut (så att rutan fylls ut bättre) men ibland vill man ha grafen i ett kvadratisk koordinatsystem.

Rita slutligen upp plotten med knappen `Plot`.

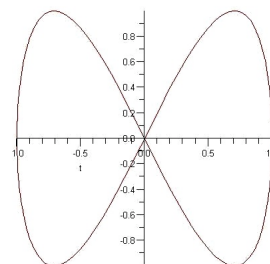


Man kan zooma in i och flytta plotten genom att välja en annan Manipulator. Högerklicka på plotten och välj Manipulators. Scale används för att zooma in och ut ur plotten, och Pan används för att flytta mittpunkten åt olika håll. Med Point Probe kan man se koordinaterna för musen i plottens koordinatsystem i övre vänstra hörnet.

Plotning av parametriska funktioner ($\mathbb{R} \rightarrow \mathbb{R}^n$)

En parametrisk plot görs genom att ange de beroende funktionerna som en lista, d.v.s. inom raka parenteser: $[\cos(t), \sin(2*t)]$.

Använd sedan Plot Builder som vanligt. Plot Builder kommer att föreslå typen Parametric Plot, och man kan ställa in värdet för parametern i Builder-fönstret (dock inte för funktionerna separat).



Animation och interaktion ($\mathbb{R}^2 \rightarrow \mathbb{R}$)

Är det fråga om en förändring över tid eller en funktion som beror på något som man i verkligheten skulle kunna kontrollera, kan det vara vettigt att antingen animera en av variablerna eller kunna kontrollera den själv. Maple kan göra bägge: antingen ändra en variabls värde över tiden, eller koppla värdet till en "slider".

Om en funktion har mer än en oberoende variabel (till exempel $u(x, y)$ eller $h(\text{vinkel, kraft})$) kommer Plot Builder att erbjuda Interactive plot with 1 parameter och Animation som plottningstyper under Select Plot Type.

Animation görs genom att välja Animation som Plot Type. I rutorna längst ner kan man välja vilken variabel som ska användas till vilket syfte: någon av axlarna, samt animationsparameter. Man kan ställa in vilket intervall som parametern ska animeras över. Prova till exempel att animera $\sin(v) * k^2$ över området 0 till $2*\text{Pi}$. När du markerar den resulterande plotten får du nya kontroller högst upp i fönstret – vanliga bandspelarkontroller. Genom att klicka på uppspelningskontrollen (▶) kan du välja mellan att bara spela upp animationen en gång eller att loopa den. Se även avsnittet Spara plottar som bild för att se hur man kan spara animationen som en GIF-animation som kan användas på hemsidor.

Givetvis går det att animera parametriska plottar också – se bara till att välja rätt variabel som parameter och rätt variabel som ska animeras.

Interaktion görs genom att välja Interactive with 1 parameter. (Har du fler oberoende variabler kan du välja att få fler av dem som interaktiva.) Ställ in plotten som vanligt. Innan den sätts in i dokumentet får man upp ett fönster där man kan finjustera värdena innan de sätts in i dokumentet.

Plotning i 3D

Ett sista sätt att plotta funktioner med mer än en oberoende variabel är att rita upp ett objekt i rymden. Detta är mycket användbart för att visualisera lösningar i linjär geometri och algebra, men även för att rita upp ytor för funktioner av två variabler. Värt att förtydliga är att om funktionen ges som en ekvation måste man flytta över högerledet för att sätta ekvationen lika med noll, och bara ge vänsterledet. Detta görs enklast genom att högerklicka på högerledet och välja Move To Left, och sedan högerklicka på resultatet och välja Left-hand Side, och sedan plotta det resultatet.

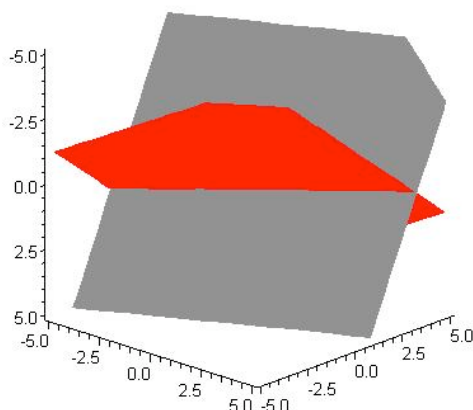
Funktionsytor av två variabler plottas som vanligt: definiera funktionen (som ett uttryck, inte funktion) och högerklicka på den, och välj Plots \rightarrow Plot Builder. 3D Implicit Plot är nästan alltid den plot-typ man söker (detta ger en yta där värdet av funktionen ges på z-axeln). Utöver det är 3D-plottar lika lättskapade som 2D-plottar. Prova att plotta funktionen $\sin(x^2 + y^2)/x$, och titta på avsnittet om att förtydliga 3D-plottar nedan.

Utöver inzoomning och förflyttning kan en 3D-plot också roteras genom att man klickar och drar plotten. Detta är grundläget för Manipulatorn, men du kan välja ett annat läge från popupmenyn Manipulator. Systemets axlar ritas heller inte ut automatiskt, utan måste slås på via menyn Axes \rightarrow Boxed (rekommenderas, men Framed eller Normal fungerar naturligtvis också).

Ekvationssystem tolkade som plan i rummen plottas också på samma sätt som ekvationssystem i 2D, med samma detaljer som för ytor av två variabler. Ett exempel från linjär algebra visar hur det går till i praktiken:

```
e1 := 2x - y + z - 2      # Ekvation 1: 2x - y + z = 2
e2 := 3x - 2y - 4z      # Ekvation 2: 3x - 2y - 4z = 0
esys := {e1, e2}        # Definiera ekvationssystemet
```

Plotta resultatet. (Planen på bilden har färgats om, se avsnittet om att förtydliga plottar nedan.) Linjen som utgör lösningen till systemet syns tydligt där planen skär varandra.



För att göra en 3D-plot tydligare kan man vara behjälpt av att göra ytorna slätare. Från Options-knappen i Plot Builder-fönstret kan man ställa in Grid Size (från rutan Miscellaneous) till något högre än förinställningen 8x8x8.

15x15x15 är lagom, högre än så kan göra att plotten tar lång tid att beräkna, även på en snabb dator. När plotten väl är uppritad finns det ett och annat att justera för att göra den tydligare, särskilt inzoomning och rotation med Manipulátorn. När modellen syns tydligt kan man som tidigare förklarats slå på uppritning av axlarna via menyn Axes. Från menyn Styles kan man ändra på vilket sätt ytorna ritas upp. Grundinställningen är att de färgas och överläggs med ett rutmönster. Detta rutmönster är utmärkt för krökta ytor, men för plan blir de lätt distraherande. En bra style för plan är Patch (without grid). Man kan också ändra färgläggningen av figuren eller för enstaka funktioner i den via menyn Color.

Om man gör ändringar i uppritningen av figuren genom att klicka på dess bakgrund tillämpas ändringarna på alla funktioner i grafen. Om man däremot klickar på en av ytorna ändras bara den ytan.

Spara plottar som bild

Alla plottar går att exportera som bildfiler genom att använda menyn Export. Högerklicka, välj Export och sedan GIF eller JPG. Spara filen. Den kan sedan importeras eller läggas in i Word- eller LaTeX-dokument, och de flesta andra ordbehandlare. De andra formaten är knappast användbara för studenter. Exporterar man en animerad plot som GIF blir resultatet en GIF-animation som kan läggas in på hemsidor.

Exempel: animera hyperboloider

En uppsättning specialfall av krökta andragsytor är enmantlade hyperboloider, elliptiska dubbelkoner och tvåmantlade hyperboloider. De har alla ungefär samma ekvation, tre termer $(x/a)^2$, $(y/b)^2$ och $(z/c)^2$. Två av termerna har samma tecken och en term har motsatt tecken. Det som skiljer de tre kropparna åt är högerledet i ekvationen: den enmantlade hyperboloiden har högerledet 1, den tvåmantlade har högerledet -1 och dubbelkonen har högerledet 0. Genom att plotta upp en sådan kropp, och variera högerledet mellan -1 och 1 kan man animera förhållandet mellan de tre kropparna.

Man sätter a, b och c till ett, för att få en enkel, cylindrisk kropp. Skapa först ekvationen, sätt högerledet till k och flytta det till vänstersidan: $x^2 + y^2 - z^2 + k$

Högerklicka, starta Plot Builder och välj 3D Implicit Plot. Koppla sedan de tre variablerna x, y och z till Parameter 1-3 och ställ in deras gränser till -3..3. Välj k till Animation Parameter, och välj gränserna till -1 och 1. Klicka sedan på Options, och ställ in Grid Size till 15x15x15. Plotta resultatet. (Detta kan ta ett tag.)

Resultatet bör bli en dubbelmantlad hyperboloid. Klicka på Play-knappen för att se den bli en enmantlad hyperboloid via en dubbelkon. Via fler inställningar kan man dock få animationen mer putsad. Ställ in uppspelningen till Oscillate och Continuous (Menyn Animate). Detta får uppspelningen att loopa fram och tillbaka. Uppe i kontrollraden finns en ruta som heter FPS (Frames per second). Ställ in den till ett högre värde, till exempel 24 (samma som film på bio). Ställ in Style på Patch (without grid), och Lighting till Light Setup 1. Spela upp animationen igen.

MAPLES HJÄLPAVSNITT

Öppna Maples hjälpsnitt

Maples hjälpsnitt öppnas antingen via menyn **Help** eller genom att trycka på **F1**. Gäller det instruktioner för hur ett visst kommando används kan man skriva ett frågetecken före kommandonamnet och exekvera det, till exempel `?diff`, `?solve` eller `?ZeroMatrix`.

Beroende på hur man öppnar hjälpen ser man först och främst ett gult fönster till höger med själva hjälptexten, och antingen sökresultat eller innehållsförteckning till vänster. Innehållsförteckningen går att bläddra igenom på vanligt sätt, och sökresultaten dyker upp om man söker på ett visst begrepp eller kommando.

Om Maples hjälpsnitt

Maples hjälpsnitt är inte bara dokumentation av själva programmet, eller dokumentation av kommandona. Maple innehåller även en mycket genomgående uppslagsbok om matematiska termer och begrepp. Många av förklaringarna är på en ganska hög nivå, men man kan alltid läsa dem och få en ny synvinkel på ett begrepp. Dessa små artiklar har *Definition* efter sitt artikelnamn, till exempel *Taylorpolynomial, Definition*.

Understruken text är länkar, precis som på en hemsida. Artiklar som är mer specifika djupdykningar i en viss aspekt av ett ämne har ett snedstreck och ett namn efter ämnesnamnet, till exempel *dsolve/ODE* för mer information om hur kommandot *dsolve* löser ordinära differentialekvationer.

Hur man läser kommandohjälpen

Hjälptexten för ett Maple-kommando är indelad i sju delar:

- **Rubriken:** kommandots namn och en kort beskrivning av vad det gör
- **Calling Sequences:** anropssekvenser, en lista på alla olika sätt man kan ge kommandot. Ett kommando tar (nästan) alltid en mängd olika *argument*, men vissa av dem är frivilliga. Kommandot *dsolve* (lösning av differentialekvationer) har till exempel följande anropssekvenser:

<code>dsolve(ODE)</code>	Lös en ordinär differentialekvation
<code>dsolve(ODE, y(x), options)</code>	Lös för en viss funktion, eventuellt med tillägg
<code>dsolve({ODE, ICs}, y(x), options)</code>	Lös med begynnelsevillkor (här kallade ICs)

En viktig sak att tänka på är att argumentens "namn" (*ODE*, *y(x)*, *options*) bara är förslag på namn, det går utmärkt att anropa *dsolve* med `dsolve(min_diffekvation, x(t))`

- **Parameters:** en närmare beskrivning av vad alla de olika argumenten till kommandot betyder och gör. De flesta kommandon har en parameter kallad "options" (den sista parametern). Den används för att göra noggranna inställningar i hur kommandot arbetar, men om man inte gör mycket komplicerade uppgifter kan man bortse från den helt.
- **Description:** en närmare beskrivning av vad kommandot faktiskt gör, och en grov överblick över hur det går till. I många fall finns länkar till mer noggranna beskrivningar över olika aspekter på kommandot.
- **Examples:** den intressanta delen. I de flesta fall kan man lista ut hur ett kommando genom att titta på exempel och utmatning. De första exemplen brukar vara mycket triviala, och lätta att titta igenom.

SIMULERING

Utöver de matematiska programmeringsverktygen har Maple en uppsjö kommandon och mekanismer för mer generell programmering. Att lära ut generell programmering är ett alldeles för stort ämne för den här handledningen, men ett exempel på simulering med Monte Carlo-metoden är möjligt att gå igenom.

Inledande programmering

Programmering fungerar i stort på samma sätt som vanlig Maple-användning, med ett par undantag: man måste använda textkommandona hela tiden, och att skriva in block med hjälp av skift-retur är vanligare än att avsluta rader med retur. Detta beror på att skift-retur avslutar en rad inom en *execution group*, vilket är ett programblock som Maple uppfattar som en helhet. Programmet måste skrivas inom en *execution group*.

För att sätta in en ny execution group, vilket behövs för att få korrekt utmatning från ett program, använder man menyn `Insert`→`Execution Group`→`Before Cursor`.

De verktyg som “programmering” som disciplin ger, utöver Maples vanliga färdigheter är i kort möjligheten att upprepa en del av programmet ett stort antal gånger med bara två extra rader kod, utskrift av text, samt möjligheten att utföra olika rader beroende på ett visst villkor.

För att skriva ut text använder man kommandot `print` som följer: `print("Valfri text");` Ett bra användningsområde är att kontrollera att programkörningen faktiskt kommer till en viss punkt, genom att skriva ut ett meddelande.

Flödeskontroll

För att upprepa en del av ett program ett stort antal gånger sätter man en så kallad *for*-slinga runt det:

```
for n from 1 to 20 do
  a := i^2;
  print(a);
end do
```

Variabeln `n` i raden är en så kallad räknare – det är en variabel som räknas upp från 1 till 20 (i det här fallet) varje gång blocket inom slingan körs. Räknaren måste ha ett intervall att räkna igenom (som tyvärr inte kan skrivas på den vanliga intervallformen) som skrivs som `from a to b`, där `a` och `b` är intervallets gränser. Raden avslutas med `do`, för att markera att blocket börjar. Blocket måste avslutas med `end do` och ett semikolon (eller ett kolon för att tysta all utmatning från slingan).

I själva blocket kan man utföra vilka vanliga Maple-beräkningar som helst, under förutsättning att det görs med textkommandon. Vill man räkna antal av någonting genom att öka värdet på en variabel görs det lätt genom att skriva `värde := värde + 1;` eller liknande.

För att öka läsbarheten i koden är det en god vana att slå in några mellanslag i början på raderna i varje block. Detta gör att knepiga avsnitt som dubbla slingor blir mer lättlästa:

```
for x from 1 to 20 do
  for y from 1 to 20 do
    a := x*y;
    print(a);
  end do
end do
```

För att hantera villkor i programmet använder man ett så kallat *if*-block:

```
tal := 11;
if tal < 20 then
  print("Under tjugo");
end if;
```

Blocket måste avslutas med `end if` på samma sätt som `for`-blocket måste avslutas med `end do`. Man kan kombinera flera villkor till ett enda block genom `else` (alla andra fall) eller `elif` (om inte det tidigare villkoret var sant, testa ett annat):

```
tal := 11;
if tal < 20 then
  print("Under tjugoo");
elif tal > 20 then
  print("Över tjugoo");
else
  print("Exakt tjugoo");
end if;
```

Det går också bra att lägga ett `if`-block inuti ett annat, och att lägga ett `if`-block i en `for`-slinga. Exemplet nedan skriver ut alla primtal mellan 1 och 100:

```
for tal from 1 to 100 do
  if isprime(tal) then
    print(tal);
  end if
end do
```

`if`-villkoret måste vara ett värde, vilket kan vara lite förvirrande. Exempelvis är `int(sqrt(n)) = n` ett värde som fungerar med `if`. Om villkoret inte är ett värde får man felet `Error, cannot determine if this expression is true or false`. Det enklaste sättet att komma runt det är att lägga ett `evalf()` runt uttrycket som inte fungerar, till exempel:

```
if evalf(exp(3)) < 1000 then print("Sant") end if
```

Att lära sig programmera är inte svårt, men kräver tid och övning. Att bara läsa exempel leder oftast ingen vart, så för att bilden ska klarna rekommenderas att arbeta igenom exemplet nedan i Maple.

Monte Carlo-metoden. Approximera π .

Monte Carlo-metoden är en statistisk metod för att göra beräkningar som beror på slumpmässiga händelser. Programmering i Maple lämpar sig bra för att göra stora antal upprepade tester, vilket är precis vad Monte Carlo-metoden bygger på.

Ett lite längre exempel, Buffons nålproblem, går ut på följande: om man tappar en nål på ett plankolv finns det en viss sannolikhet att den korsar mellanrummet mellan två plankor. Denna sannolikhet går att beräkna, och är lika med ett uttryck innehållande π . Istället för att ha en ekvation för sannolikheten som beror på π , löser man ekvationen för π istället. Man har nu ett uttryck för π som beror på sannolikheten för att nålen ska korsa ett av mellanrummen. Tappar man nu tiotusen nålar på golvet och räknar efter hur många nålar som faktiskt korsar ett av mellanrummen kan man sätta in den i uttrycket för π och få en hyggligt bra approximation.

Uttrycket för sannolikheten är ganska komplicerat⁴, men det finns en enklare version: att kasta pil på en piltavla. Antag att man har en piltavla på en vägg, som utgörs av en cirkel med radien r inskriven i en kvadrat.

```
area_cirkel := r^2 * Pi
area_kvadrat := 4r^2
```



Cirkelns area utgör en viss del av kvadraten, en kvot: cirkelns area delad med kvadratens area:

```
ekv := area_cirkel = kvot * area_kvadrat ( $r^2 \cdot \pi = k \cdot 4r^2$ )
```

Man kan nu lösa ekvationen med avseende på π . Eftersom π redan har ett numeriskt värde i Maple ersätter vi π med `s`:

⁴ Läs gärna mer på http://en.wikipedia.org/wiki/Buffon's_needle

$$\text{subs}(\text{Pi} = \text{s}, \text{ekv}) (r^2 \cdot \text{s} = k \cdot 4r^2)$$

Lös ekvationen med avseende på s. Svaret är att $s = 4 \cdot \text{kvot}$, alltså att π är $4 \cdot \text{kvoten}$ mellan cirkelns area och kvadratens area. Man kan nu approximera den kvoten genom att kasta pil på tavlan. Har man kastat 1000 pilar och 600 av dem hamnade i cirkeln kan man sluta sig till att kvoten är 1000/600, alltså 0.6. Det är ett hästjobb att kasta tusen pilar, men man kan be Maple göra det.

Tanken är att generera ett stort antal slumpmässiga punkter i kvadraten, det vill säga i intervallet $x = [-r..r]$ och $y = [-r..r]$. En punkt ligger på cirkeln om avståndet från mitten är mindre än cirkelns radie, d.v.s. om $\text{sqrt}(x^2+y^2) < r$. Maples slumpgenerator fungerar så att man genererar en procedur – en egen funktion – som skapar tal i ett visst intervall.

```
r := 10000;           # Pröva andra radier - ju större desto noggrannare.
slump := rand(-r .. r); # Kolon för att slippa ful utskrift från rand()
slump()              # Generera slumpstal i intervallet
                    -445
```

Nu måste två saker ordnas: programsnutten måste upprepas några tusental gånger, och kvoten måste beräknas. Uppreppningen görs genom att sätta beräkningen i en for-slinga, och kvoten beräknas genom att räkna hur många pilar som "träffade". Resultatet blir som följer:

```
r := 10000:
träffar := 0:
slump := rand(-r .. r);
for i from 1 to 10000 do
  x := slump():
  y := slump():
  avstånd := evalf(sqrt(x^2 + y^2)):
  if avstånd < r then träffar := träffar + 1: end if # Räkna träffarna
end do:
approxpi := evalf(4 * träffar / i); # Beräkna  $\pi$  som 4*kvoten
```

(Notera att alla rader utom den sista slutar med kolon, inte semikolon. Med semikolon blir mängden utskrift ohanterlig!)

Med 10000 iterationer bör approxpi ligga kring 3.14 ± 0.01 , vilket är en bra approximation med tanke på att beräkningen är helt slumpmässig!

RAPPORTER

Även om man i de flesta fall skriver sina rapporter i Word, OpenOffice eller LaTeX går det faktiskt att producera bra utskrifter även från Maple.

Städa upp arbetsbladet

Det första steget i att göra om ett arbetsblad till en rapport är att ta bort all överflödigt in- och utmatning. Under menyn Edit finns valet `Remove Output → From Worksheet`. Detta tar bort all utmatning, alla plottar och alla felmeddelanden. Gå sedan igenom ditt arbetsblad och ta bort alla kommandon som inte är relevanta, som blev fel eller som bara var till för att testa hypoteser och metoder. Tysta ointressanta rader genom att avsluta rader med kolon istället för semikolon eller inget tecken alls. När dokumentet enbart matar ut intressanta data kan man trycka på !!!-knappen för att köra om hela dokumentet.

Textinsättning, formgivning, LaTeX och utskrift

Det går att sätta in vanlig text i arbetsbladet. Detta fungerar som kommentarer och förklaringar, men det går med lite arbete att göra riktig layout.

För att skjuta in text i ett dokument sätter man markören där man vill ha stycket, och väljer `Insert → Paragraph` från menyn. Detta blir dock inte nödvändigtvis ett textstycke som går att snygga upp särskilt mycket, utan blir bara ett nytt block för matematik om man var i matematikläget innan.

Väljaren Text/Math i övre vänstra hörnet används för att växla mellan Maple-inmatning och vanlig text i ett stycke. Man kan också växla mellan dem med knappen F5 på tangentbordet. I textläget tolkas inget av det man skriver in som Maple-kod, utan som text i vilken ordbehandlare som helst. Man kan ändra typsnitt, storlek och understrykning/fetstil/kursiv stil via kontrollerna ovanför textfönstret.

Mapleutmatning kan kopieras och klistras in i de flesta ordbehandlare. I många fall konverteras dock färdiga ekvationer och uttryck till Maple-kod. Till exempel blir $\frac{2x^2-5}{3}$ till `(2x^2-5)/3`. Om så är fallet är det tyvärr inte mycket att göra åt. I Word kan man återskapa ekvationen med Equation Editor, men skriver man sin rapport i LaTeX ser det betydligt ljusare ut. Maple kan nämligen konvertera all utmatning till motsvarande LaTeX-kod. Detta görs genom att högerklicka på det man vill konvertera och välja `Conversions → LaTeX`. Resultatet kan klippas ut och klistras in i vilket LaTeX-dokument som helst, som vanligt. Man kan även exportera hela arbetsbladet till LaTeX från `File → Export → LaTeX`.

Utskrift sker som vanligt på samma sätt som i alla andra program beroende på vilken sorts dator man använder. Värt att tänka på är dock att minska ner eller förstora plottar så att de blir läsliga (markera plotten och dra i hörnen för att ändra storlek på den). Skriver man ut i svartvitt kan också 3D-grafer vara svårtolkade om de är färgade. Ett första steg för att förtydliga ytorna genom att välja `Style → Patch (with grid)`. Ett andra steg, som ger bättre resultat är att välja färger för ytorna som skiljer sig åt även i svartvitt. Högerklicka på ytorna en och en och tilldela dem färger av olika ljushet från menyn `Color`. Kombinationer av `Yellow`, `Blue` och `Gray` brukar synas bra. Vidare bör man tänka på att animationer naturligtvis inte fungerar på papper, så att man antingen väljer en bildruta som illustrerar förloppet på ett bra sätt, eller sätter in flera plottar med olika bildrutor som visar en större del av förloppet.

FELSÖKNING

Man kör oundvikligen fast förr eller senare. Antingen ger programmet felmeddelanden eller så blir svaren svårtolkade eller direkt fel från en matematisk synvinkel. Blir matematiken fel på grund av felaktig inmatning eller felaktig matematik från användaren finns inte mycket mer att göra än att kontrollera stegen noggrant, plotta mellanresultat och tänka efter ordentligt. Prioritet ett måste vara att hitta *var* felet uppstått, sedan kan man lösa det. Om inget annat hjälper är det läge att ta bort all onödig inmatning (se avsnittet Städa upp arbetsbladet) och sedan göra en fräsch genomkörning med !!!-knappen.

Det blir inte lättare av att Maples felmeddelanden är ganska tekniska och ibland oklara. Därför finns nedan en lista på vanliga felmeddelanden, orsaker och åtgärder.

Vanliga felmeddelanden

- **Empty plot**
Den funktion som skulle plottas ger inga punkter i intervallet. Detta kan antingen bero på att funktionen inte är definierad i intervallet, inte är reellvärd i intervallet, att någon av konstanterna inte har något värde satt, eller att den variabel man försöker plotta med avseende på inte finns med i uttrycket.
- **Error, attempting to assign to `system` which is protected**
Många namn är upptagna i Maple, bland annat namnet system, vilket man ofta vill använda för att döpa ett ekvations-system till. Döp det till ekvsys, systemet eller något annat.
- **Error, cannot determine if this expression is true or false**
En if-sats kan inte jämföra villkor som inte är heltal eller rationella uttryck. Om resultatet av exempelvis `sqrt()` är för komplicerat kan man behöva sätta `evalf()` runt det.
- **Invalid arguments**
Kommandon i Maple är petiga med vilka argument som skickas till dem. Kolla upp hjälpavsnittet för kommandot och läs Calling Sequence och Parameter-avsnitten noggrant. Om inget fel är uppenbart, kontrollera att namnen på variablerna som skickas till kommandot inte har samma namn som något annat i Maple. Gör detta genom att sätta ett frågetecken framför alla parameternamn, ett och ett och se ifall ett hjälpavsnitt dyker upp för det namnet. Isåfall måste parametern döpas om.
- **Invalid subscript**
Att skriva in variabler med index (till exempel `area_cirkel`) kan vara lite lömskt, eftersom man lätt glömmer att trycka på högerpil när indexnamnet är färdigskrivet. Detta kan leda till att man råkar skriva semikolonet nere i indexnamnet, vilket gör det till ett ogiltigt namn. Detta gäller även exponenter.
- **Invalid directional argument**
När man skriver in gränsvärden kan man råka blanda ihop i vilken ordning argumenten ska komma. Rätt anropssekvens är `limit(funktion, gränspunkt, eventuell riktning)`. Giltiga riktningar är 'left' och 'right'.
- **Unable to match delimiters**
Det fattas en parentes, hake eller måsvinge någonstans. Det är lätt hänt att skriva in en höger- eller vänsterparentes för mycket eller för lite när man har många uttryck inom parentes. Kontrollera raden noggrant.
- **... unexpected**
Detta felmeddelande kommer upp om Maple ser ett tecken som inte var väntat. Detta är ett mycket vanligt fel som kan orsakas av ett missat komma, ett komma för mycket, en parentes på fel ställe, en variabel som inte ska vara just där och så vidare. Det enklaste sättet att få bukt med problemet är faktiskt att skriva om raden från början. Ofta är det något litet tecken som smugit sig in, som man inte skriver in andra gången. Om inte det fungerar heller har man missförstått något i formen. Bygg upp uttrycket ett steg i taget, och kontrollera hela tiden med Enter för att se när problemet uppstår.

Vanliga fel

Man har skrivit in en rad som använder någon av Maples inbyggda funktioner, till exempel `diff`, `int`, `isprime`, `cos`, `exp` eller `print`. När man kör raden kommer precis samma rad ut som utmatning, som om Maple struntat i att behandla raden alls. Det är utan motstycke det vanligaste felet. I 90% av fallen har man råkat slå in ett mellanslag mellan funktionens namn och den första parentesen. För studenter som är vana att programmera i andra språk är detta förvirrande, men funktionen heter faktiskt `sin()`, inte `sin ()`.

Om en variabel som tilldelats ett värde inte tas med i beräkningarna, utan bara skickas igenom beräkningen, har man troligen råkat tilldela den ett värde med `=`, inte `:=`. Detta tolkar Maple som en ekvation, inte en tilldelning. Skillnaden är stor. Om beräkningen inte kommer ut som väntat bör detta vara det första man kontrollerar.

Om trigonometriska uttryck inte behandlas i beräkningen kan man ha råkat skriva `sinx` eller `cosx` av gammal vana. Detta tolkar Maple som en variabel som faktiskt heter `sinx` eller `tanx`, inte uttrycket `sin(x)` eller `tan(x)`. Detsamma gäller om man försökt skriva `a*x` som `ax` - det tolkas inte som en faktor, utan som en variabel vid namn `ax`.

Om π och e inte behandlas rätt i beräkningen kan man lätt ha råkat skriva in dem fel. π heter `Pi` i Maple, inte `pi` (godtycklig variabel) eller `PI` (stora π , d.v.s. Π). Behöver man skriva in konstanten e görs det som `exp(1)`.

Om ett väntat menyval inte kommer upp när man högerklickar på en funktion är det för att det man klickar på inte är av den typ man trodde. Till exempel kommer inte Plot Builder upp om man klickar på en funktion definierad som `f:=x->x^2`, eftersom det egentligen är en procedur. Man kan inte lösa ett uttryck, bara ekvationer, och man kan inte derivera ekvationer, bara uttryck. Det finns ingen patentlösning på problemet, utan det är bara att titta efter noga vad det egentligen är man försöker operera på.