

Module 2 Het gebruik van Maple, vervolg

Onderwerp	Nadere detaillering van een aantal zaken van Module 1
Voorkennis	Geen
Expressies	<code> </code> , <code>' '</code> , <code>" "</code> , <code>\</code> , <code>save</code> , <code>read</code> , <code>protect</code> , <code>unprotect</code>

2.1 Decimale getallen

In §1.3 hebben we gezien dat we *getallen* op een voor de hand liggende manier aan Maple kunnen doorgeven: gewoon een rij cijfers, al of niet met een decimale punt erin. Voor heel grote of kleine getallen is er ook een andere manier.

Voorbeeldsessie

```
> 1.4*10^(-7);
1.400000000 10-7
> 1.4e-7;
1.4 10-7
> 34e8;
3.4 109
```

Toelichting

Als het getal 1.4×10^{-7} op de gebruikelijke manier wordt ingevoerd, dan wordt het door Maple 'uitgerekend'; we zien dat aan de nullen die erachter worden gezet. We kunnen dit getal ook, korter, invoeren zoals in de tweede regel is gebeurd. Maple rekent dan niets zelf uit, het geeft evenveel cijfers achter de komma terug als wij hebben gegeven.

Let op dat elk getal dat met de 'e-notatie' is ingevoerd, door Maple automatisch van een decimale punt wordt voorzien. Nullen vóór de decimale punt worden door Maple weggelaten⁶. ◇

! Pas op dat u bij de e-notatie *geen* haakjes mag gebruiken. Kijk maar eens wat `1.4e(-7)` oplevert.

⁶Let daar goed op, want op een Windows-scherm valt zo'n punt vaak niet erg goed op.

2.2 Beheer van execution groups

Soms zal een opdracht zó lang zijn dat hij niet op één regel past. Dat is niet erg, als u gewoon doortypt, dan wordt er als de regel vol is, gewoon op de volgende regel doorgedaan. Bij zo'n lange opdracht zal men echter graag zélf de regelovergangen willen bepalen, al was het maar om uw werkblad een beetje leesbaar te houden. Dat gebeurt met [Shift]-[Enter], dit betekent dus: "ga naar een nieuwe regel, en voer de opdracht(en) nog niet uit."

Zoals bekend mogen er méér opdrachten op één regel worden gegeven. Met [Shift]-[Enter] kunen dus ook verschillende, bij elkaar horende opdrachten op verschillende regels worden gegeven. Als dan op één van die regels op [Enter] wordt gedrukt, dan worden *alle* bij elkaar horende opdrachten achter elkaar uitgevoerd. De resultaten ervan komen in de juiste volgorde er onder elkaar onder te staan. Als u niet alle tussenresultaten wilt zien, dan moet u de betreffende opdrachten met een dubbele punt inplaats van een puntkomma afsluiten.

Welke regels bij elkaar horen (*execution group*) is te zien aan de vierkante haak (`[]`) aan de linkerkant. Als u op [Enter] drukt in een rood Maple-commando springt de cursor, na uitvoering van de opdracht, automatisch naar de volgende execution group, er wordt een nieuwe gemaakt als we onderaan het werkblad zijn.

Onder Edit (Menu bar) vindt u ook *split or join*, waarmee u execution groups kunt *splitsen* en *samenvoegen*. Er wordt gesplitst op de plaats van de cursor; samengevoegd met de execution group onder de execution group waar de cursor in staat.

Als een execution group is ontstaan door een aantal groepen samen te voegen die elk (door gebruik van [Shift]-[Enter]) uit meer regels bestaan, dan zullen sommige regels in deze groep wél een prompt (>-symbool) hebben en andere niet. Dat is niet erg, als u [Enter] geeft op de groep, wordt hij in z'n geheel uitgevoerd.

2.3 Namen van Variabelen

Variabelen kunnen lange namen hebben, die mogen bestaan uit alle (hoofd- en kleine) letters uit het alfabet, alle cijfers, en het teken `_` (underscore). Verder mogen namen niet beginnen met een cijfer.

! Maple maakt onderscheid tussen hoofdletters en kleine letters.

||

We kunnen namen ‘aan elkaar plakken’ met de *concatenatie*-operator `||` (dubbele verticale streep). De naam `x||y` is (op een enkele uitzondering na) hetzelfde als de naam `xy`.

Griekse letters kunnen ook worden gebruikt door ze voluit te typen, of door gebruik te maken van het palet links op het scherm. Zo wordt `alpha`; weergegeven als α enzovoort. Griekse *hoofdletters* krijgt men door de naam met een hoofdletter te typen: `delta` wordt δ en `Delta` wordt Δ .

‘ ‘

Als we ‘verboden symbolen’, zoals spaties, *’s, punten en dergelijke, in namen willen gebruiken kan dat door de hele naam tussen ‘ ‘-tekens (zogenaamde *back-quotes*) te zetten. Deze zitten op de meeste toetsenborden onder dezelfde toets als het symbool `~`.

Voorbeeldsessie

```
> alpha := 6;
                                alpha := 6
> a := x||y;
                                a := xy
> xy - a;
                                0
> seq( x||i, i=1..4 );
                                x1, x2, x3, x4
> ‘rare breuk’ := ‘x/y’ / ‘p*’;
                                rare breuk :=  $\frac{x/y}{p^*}$ 
> x := 4; ‘rare breuk’;
                                x := 4
                                 $\frac{x/y}{p^*}$ 
> ‘x/y’ := 4; ‘rare breuk’;
                                x/y := 4
                                 $\frac{4}{p^*}$ 
> ‘x’;
                                4
> ‘x ‘;
                                x
```

Toelichting


We zien dat xy inderdaad precies hetzelfde is als $x||y$. Deze concatenatie-operator kan bijvoorbeeld worden gebruikt om met een `seq`-opdracht een hele rij namen te genereren. Meer hierover vindt u in Module 8.

De naam `rare breuk` moet tussen back-quotes worden gezet, omdat er een spatie in voorkomt. De variabele `'x/y'` is iets geheel anders dan de breuk x/y , hoewel men dat op het scherm niet kan zien; de waarde ervan heeft dan ook niets te maken met de waarde van x en van y .

Er is één speciaal geval bij het gebruik van de back-quotes: omdat x een toegestane naam is, is `'x'` hetzelfde. Zodra we een extra spatie toevoegen wordt het iets anders. \diamond

2.4 Commentaar invoeren

In §1.6 hebben we al gezien dat het teken `#` kan worden gebruikt om commentaar toe te voegen. Dat is vooral handig voor zeer korte toelichtingen.

Voor uitvoeriger commentaar is de volgende methode beter geschikt. Hiermee is het ook mogelijk om nette formules in de tekst op te nemen. Door aanklikken van de -button in de toolbar wordt een commentaarregel (*Text input*), toegevoegd. Deze is te herkennen aan het ontbreken van de prompt (`>`). U kunt nu tekst intypen. Deze wordt in zwarte letters weergegeven.

Merk op dat de context-bar veranderd is. Helemaal links zitten de knoppen `Text` en `Math`. Hiermee kunt u kiezen tussen het invoeren van gewone tekst of formules. Bij het invoeren van formules kunt u gebruik maken van de paletten aan de linkerkant van het scherm.

Tenslotte is het ook mogelijk het werkblad in te delen in paragrafen (*sections*) en sub-paragrafen, die alle afzonderlijk geopend en gesloten kunnen worden. U kunt het beste zelf eens experimenteren met `Insert Section`.

2.5 Bewaren van resultaten; Bestanden

Classic Worksheet. Een `.mw`-bestand heet een *standard worksheet* (door Maple zelf meestal aangeduid als Maple Worksheet). In het geval dat uw werkblad bijvoorbeeld gebruikt zal worden door iemand anders die misschien een oudere versie van Maple heeft, is het verstandig het (ook) op te slaan als een zogenaamd *classic worksheet*. Deze versie wordt door Maple herkend door de toevoeging `.mws` (in plaats van `.mw`). In dat geval is het van belang eerst *alle* (blauwe) uitvoer uit het werkblad te verwijderen via `Edit` → `Remove Output`. Daarna na `File` → `Save as...` onderaan bij `Files of type` kiezen: `Maple Classic Worksheet`.

Denk er aan dat sommige dingen in de ‘classic’-versie niet werken, bijvoorbeeld de verwijzing met labels (zie §1.6). Ook het gebruik van formules in commentaarregels wordt door de ‘classic’-versie niet herkend. Gebruik deze dus niet als u van plan bent om uw werkblad als Classic Worksheet op te slaan.

Andere mogelijkheden. Het volgende is vooral van belang als u resultaten van ingewikkelde berekeningen wilt bewaren voor later gebruik, of wilt opnemen in een ander werkblad. Het is mogelijk om, in plaats van de hele sessie, ook alleen de *waarden* van een aantal variabelen te bewaren, maar niet de commando’s waarmee deze waarden zijn berekend en de commentaarregels. Dit opslaan gaat met het commando `save`, bijvoorbeeld

`save`

```
save(a,b,c, "c:/userdata/abcwaarde.txt")
```

" "

voor het opslaan van de waarde van de variabelen `a`, `b` en `c`. *Let op de "-tekens (string-quotes)* om de bestandsnaam. Wanneer deze niet worden gebruikt, dan zou bijvoorbeeld de dubbele punt in `c:/userdata/...` door Maple opgevat worden als de afsluiting van de opdracht. Let ook op bestandsnamen die een *pad* bevatten. Hierbij moet (in plaats van de bij Windows gebruikelijke *backslash* (`\`)) de bij Linux gebruikte ‘voorwaartse’ *slash* (`/`) gebruikt worden, óók als u onder Windows werkt. Een andere mogelijkheid is (alléén onder Windows!) om een dubbele backslash (`\\`) te gebruiken.

Als de bestandsnaam eindigt op `.m`, is dit bestand alleen voor Maple leesbaar.

We hebben drie verschillende soorten quotes leren kennen:

- ' (gewone quote) om toekenningen ongedaan te maken, bijvoorbeeld `x := 'x'`,
- ' (back-quote) om variabelen die 'verboden symbolen' bevatten aan te geven,
- " (dubbele quote of string-quote), wordt onder andere gebruikt om bestandsnamen aan te geven.

De op deze manier opgeslagen waarden kunnen bij een latere gelegenheid weer worden ingelezen met het commando

`read` `read("c:/userdata/abcwaarde.txt")`

(dus zonder de namen van de variabelen te vermelden). Na dit opnieuw inlezen hebben alle variabelen weer de waarden die zij hadden op het moment dat de toestand werd opgeslagen. Zo kan dan worden verder gewerkt. Als er veel variabelen zijn opgeslagen, of als de waarden ingewikkelde dingen zijn, bijvoorbeeld (grote) matrices of procedures, verdient het aanbeveling om zo'n `read`-commando met een `:` in plaats van met een `;` af te sluiten.

In plaats van het direct intikken van de opdrachten voor Maple, is het ook mogelijk eerst een bestand te maken waarin deze opdrachten staan. Vervolgens kunnen deze opdrachten aan Maple worden aangeboden door middel van het commando `read`. Een dergelijke werkwijze is eventueel handig als u de betreffende opdrachten niet in uw werkblad wilt hebben.

2.6 Gereserveerde namen

Sommige namen worden door Maple gebruikt als namen van opdrachten. Zo'n naam noemen we een *gereserveerde naam*. Aan een gereserveerde naam kan niet zomaar een (andere) waarde worden toegekend. Maple reageert dan met de mededeling dat deze namen *protected* zijn. Bij een naam als `simplify` is dat wel prettig, want het zou erg onverstandig zijn als we zomaar de opdracht `simplify := 5;` zouden kunnen geven (ga na waarom). Maar soms zijn er gereserveerde namen die horen bij Maple-opdrachten of -constanten die we niet van plan zijn te gebruiken. Een voorbeeld is de naam γ (`gamma`, zie `?gamma` voor de betekenis ervan). We kunnen dan deze naam beschikbaar maken met de opdracht

`unprotect` `unprotect('gamma');`

`protect` We kunnen ook zelf namen reserveren met de opdracht `protect`.

Voorbeeldsessie

```

> restart;
> Pi := 3;
Error, attempting to assign to 'Pi' which is protected
> unprotect(Pi);
> Pi := 3;
                                     3
> protect(g);
g := 5;
Error, attempting to assign to 'g' which is protected
> a := 5;
protect(a);
Error, (in protect) invalid input: setattribute received 5, which is
not valid for its 1st argument, e
> protect('a'); a;
                                     5
> a := 6;
Error, attempting to assign to 'a' which is protected
> unprotect(a);
a := 2;
Error, attempting to assign to 'a' which is protected
> unprotect('a');
a := 2;
                                     2
> restart; # om Pi weer Pi te maken...

```

Toelichting

In het algemeen is het niet verstandig om de protectie van Maple-constanten als `Pi` op te heffen omdat dan de oorspronkelijke waarde ervan niet meer gebruikt kan worden.

Als we zelf een naam willen reserveren, dan zal deze in het algemeen ook een *waarde* hebben; de bedoeling zal immers zijn dat die waarde niet ‘per ongeluk’ veranderd kan worden. We zien dat: `a := 5: protect(a);` niet werkt, en dat komt omdat Maple in het commando `protect(a);` voor `a` *direct* 5 invult, zodat het denkt dat het de ‘naam’ 5 moet reserveren. Om dat te voorkomen moet de te reserveren naam tussen quotes gezet worden. \diamond

