

Een eerste kennismaking met Mathematica

Marcel Vonk

November 2013

Inhoud

1	Over Mathematica	1
2	Notebooks en invoer	2
3	Mathematica als rekenmachine	3
4	Werken met variabelen	5
5	Het gebruik van functies	7
6	Operaties herhalen	10
7	Afgeleiden nemen, integreren en vereenvoudigen	12
8	Grafieken maken	15
9	Zelf Mathematica installeren	18
10	Meer leren	18

1 Over Mathematica

Mathematica is een softwarepakket waarmee je wiskundige berekeningen kunt doen. In Mathematica kun je met grote precisie met getallen rekenen, functies differentiëren of integreren, vergelijkingen oplossen, grafieken tekenen en nog enorm veel meer.

We zullen in deze tutorial alleen het topje van de ijsberg zien van de mogelijkheden van Mathematica. Hopelijk ben je na het doornemen van de tekst voldoende vertrouwd met het programma om, als je dat wilt, zelf verder te onderzoeken wat het programma allemaal te bieden heeft.

Handigheid in het werken met Mathematica krijg je natuurlijk niet door het lezen van een handleiding. Je krijgt die handigheid vooral door het programma veel te gebruiken. Doe dus zo veel mogelijk van de voorbeelden in deze tutorial zelf na en probeer zo nu en dan wat varianten uit. Als we in de handleiding Mathematica de afgeleide van de functie $f(x) = x^2$ laten uitrekenen, probeer dan ook eens de afgeleide van een andere functie te berekenen, verander de gebruikte variabele van x in y , of bedenk hoe je nu de tweede afgeleide (dus: de afgeleide van de afgeleide) zou uitrekenen. Op die manier ontdek je wat er wel en niet kan, en krijg je snel de nodige vaardigheid in het werken met het programma.

2 Notebooks en invoer

De bestanden waarin Mathematica berekeningen opslaat, heten *notebooks*. Als je Mathematica opstart, zie je een welkomtscherm waarin je ervoor kunt kiezen om een nieuw notebook te maken.¹ In het notebook kun je vervolgens direct beginnen met typen. Zie je de getypte tekst niet verschijnen, dan is waarschijnlijk het notebook niet geselecteerd – klik in dat geval in het notebook-venster om het te selecteren.

Laten we meteen maar onze eerste invoer in Mathematica typen: typ in het notebookvenster bijvoorbeeld

2+2

De invoer die je in Mathematica typt komt in *cellen* te staan. Zo'n cel bestaat uit één of meerdere regels, en wordt aangegeven door een blauwe “haak” die je rechts in het notebook ziet staan.

Misschien heb je nadat je 2+2 hebt ingetypt al automatisch op de <enter>-toets gedrukt. Het zal je dan zijn opgevallen dat er weinig spectaculairs gebeurt: de cursor gaat gewoon naar de volgende regel, en je kunt hier verder gaan met typen. Dat is natuurlijk niet wat we willen: we zouden graag zien dat Mathematica de som die we hebben ingetypt, uitrekent. Om dat te doen druk je niet op <enter>, maar op <shift-enter>. Als je dat doet krijg je het resultaat dat je in afbeelding 1 op de volgende bladzijde ziet. In Mathematica 8 (en oudere versies) blijft het daarbij; in Mathematica 9 verschijnt onder de uitkomst ook nog een balkje met allerlei opties. Je kunt de opties in die balk in deze tutorial gewoon negeren. Als je het balkje storend vindt kun je op het kruisje rechts in de balk klikken om het te laten verdwijnen.

Je ziet dat Mathematica de som uitrekent en tegelijk ook de cellen met invoer en uitvoer een naam heeft gegeven: In[1] en Out[1]. De horizontale lijn geeft aan waar nieuwe invoer die je typt zal verschijnen. Je kunt die lijn ergens anders plaatsen (bijvoorbeeld boven 2+2,

¹Dit is zo in Mathematica 8 en 9. In oudere versies van Mathematica krijg je vaak automatisch een nieuw, leeg notebook te zien. Is dit ook niet het geval, dan kun je via het menu File/New/Notebook een nieuw notebook openen.



Afbeelding 1: Het resultaat van een eenvoudige berekening.

of zelfs tussen de cellen met In[1] en Out[1]) door op een andere plek in het notebook te klikken. Wil je Mathematica dezelfde berekening nog eens laten doen, dan kun je weer in de cel met 2+2 klikken en nogmaals op $\langle \text{shift-enter} \rangle$ drukken.

Meerdere regels invoer opgeven kan ook. Je scheidt deze regels door een gewone $\langle \text{enter} \rangle$, en drukt als je klaar bent op $\langle \text{shift-enter} \rangle$; Mathematica rekent dan de regels een voor een door, en geeft de uitvoer voor elke regel. Typ bijvoorbeeld “8 – 3 $\langle \text{enter} \rangle$ 5 + 4 $\langle \text{shift-enter} \rangle$ ” en je krijgt het resultaat

```
In[1] := 8 - 3
        5 + 4
Out[1] = 5
Out[2] = 9
```

Mathematica geeft elke cel met invoer en uitvoer een nummer; de nummers die je in je eigen notebook ziet kunnen natuurlijk verschillen van de nummers hierboven. We zullen in de voorbeelden in deze handleiding steeds de aanduidingen “In[1]” en “Out[1]” gebruiken, om duidelijk onderscheid te maken tussen invoer en uitvoer.

Zoals bij elke computeractiviteit is het verstandig om je notebooks regelmatig te saven. Dat doe je door $\langle \text{ctrl-s} \rangle$ in te typen of in het menu File/Save of File/Save As... te kiezen. Opgeslagen mathematicanotebooks herken je aan de bestandsextensie .nb.

3 Mathematica als rekenmachine

Zoals je in het eerste voorbeeld hebt gezien, kun je Mathematica eenvoudig gebruiken als een rekenmachine. Mathematica is bovendien een heel krachtige rekenmachine: het programma berekent in een fractie van een seconde de duizendste macht van 2. (Dus $2 \times 2 \times 2 \times \dots$ met duizend factoren twee.) Het antwoord is een getal van meer dan 300 cijfers:

```
In[1] := 2^1000
Out[1] = 10 715 086 071 862 673 209 484 250 490 600 018 105 614 048 117 055
        336 074 437 503 883 703 510 511 249 361 224 931 983 788 156 958
```

```
581 275 946 729 175 531 468 251 871 452 856 923 140 435 984 577
574 698 574 803 934 567 774 824 230 985 421 074 605 062 371 141
877 954 182 153 046 474 983 581 941 267 398 767 559 165 543 946
077 062 914 571 196 477 686 542 167 660 429 831 652 624 386 837
205 668 069 376
```

Je ziet dat Mathematica voor de leesbaarheid van het getal zelf om de drie cijfers een kleine spatie invoert. Let op dat deze spaties je niet in verwarring brengen. Voor het vermenigvuldigen van twee getallen voer je in Mathematica namelijk geen maalteken (“ \times ”) in, maar ook eenvoudigweg een spatie. Vergelijk dit met de wiskundige notatie waarin we het maalteken ook vaak weglaten: ab staat bijvoorbeeld voor “ a maal b ”. Zo reken je dus 333×444 uit door in te voeren

```
333 444
```

Bij het intypen hiervan zul je zien dat Mathematica nu zelf een maalteken tussenvoegt, zodat het resultaat eruit komt te zien als

```
In[1] := 333 × 444
Out[1] = 147 852
```

Het “extra” maalteken in de invoer toont Mathematica om verwarring te voorkomen met getallen met veel cijfers, waarin zoals we gezien hebben ook spaties staan. Zo betekent het antwoord van de berekening, 147 852, natuurlijk “honderdzevenenveertigduizend achthonderdtweënvijftig” en niet “honderdzevenenveertig maal achthonderdtweënvijftig”. Kortom:

- Bij het zelf *invoeren* van een vermenigvuldiging gebruik je geen maalteken maar een spatie.
- In *uitvoer* (en de door Mathematica gelezen invoer) wordt wel een maalteken gebruikt, en is de spatie alleen een scheidingsteken.

Mathematica kent ook allerlei wiskundige constanten, zoals het getal π . We kunnen dit getal invoeren als

```
In[1] := Pi
```

al lijkt dit in eerste instantie bijzonder weinig effect te hebben: de uitvoer is

```
Out[1] =  $\pi$ 
```

De reden hiervoor is dat Mathematica dergelijke constanten altijd als *exacte* waarden behandelt, en dus niet vervangt door een numerieke benadering zoals 3,14. Als we willen dat Mathematica wel zo’n benadering gebruikt, moeten we dat expliciet aangeven met behulp van de functie `N[...]`:

```
In[1] := N[Pi]
Out[1] = 3.14159
```

Mathematica geeft numerieke waarden van getallen standaard in zes cijfers. Willen we meer nauwkeurigheid, dan is dat geen enkel probleem: we kunnen daarvoor een tweede argument aan de functie `N[...]` meegeven dat het aantal cijfers aangeeft. Bijvoorbeeld:

```
In[1] := N[Pi,100]
Out[1] = 3.1415926535897932384626433832795028841971693993751058209749445
          92307816406286208998628034825342117068
```

Ook het getal 2^{1000} dat we eerder hebben uitgerekend kunnen we met behulp van `N[...]` numeriek benaderen:

```
In[1] := N[2^1000]
Out[1] = 1.07151 × 10301
```

4 Werken met variabelen

Mathematica is natuurlijk meer dan alleen een rekenmachine. We krijgen al iets meer te zien van de capaciteiten van het programma als we gaan rekenen met variabelen. We kunnen zo'n variabele eenvoudig een waarde toekennen, bijvoorbeeld

```
In[1] := x = 6
```

Mathematica ziet zo'n variabele-toekenning als een (heel eenvoudige) berekening, en geeft dan ook de uitkomst weer:

```
Out[1] = 6
```

Vind je een dergelijke uitvoer overbodig, dan kun je Mathematica altijd vragen om de uitkomst van een berekening weg te laten. Dit doe je door de regel af te sluiten met een puntkomma. De invoer

```
In[1] := x = 6;
```

geeft bijvoorbeeld geen uitvoer. Wil je alsnog de waarde van de variabele weten, dan kun je die eenvoudigweg intypen en op `(shift-enter)` drukken:

```
In[1] := x
Out[1] = 6
```

Opmerking voor mensen met programmeerervaring: In veel programmeertalen is het zo

dat je vóór het gebruik van een variabele moet aangeven om wat voor type variabele het gaat: een geheel getal, een decimaal getal, tekst, enzovoort. In Mathematica is dat in het algemeen niet nodig. Het programma herkent zelf om wat voor soort variabele het gaat. Ook de beperkingen die in andere programmeertalen voor bepaalde variabelen gelden (een geheel getal moet bijvoorbeeld in sommige programmeertalen tussen -32768 en 32767 liggen) gelden in Mathematica niet.

Heb je eenmaal een waarde aan een variabele toegekend, zoals hierboven $x = 6$, dan kun je verder rekenen met die waarde. Bijvoorbeeld:

```
In[1] := x^2 - 3 x + 5
Out[1] = 23
```

Misschien is je bij het invoeren van $x = 6$ iets opgevallen: nadat je op \langle shift-enter \rangle hebt gedrukt, verandert de x van kleur: van blauw naar zwart. (De kleuren kunnen van versie tot versie van Mathematica verschillen.) Is dit je niet opgevallen, probeer het dan nog eens met bijvoorbeeld $y = 6$. Met de kleurverandering geeft Mathematica aan dat het nu de variabele x kent, en dat daaraan een bepaalde waarde is toegekend. Zo kun je altijd snel zien of een variabele die je wilt gebruiken nog niet bestaat; als de naam van de variabele al tijdens het intypen zwart wordt, is dat kennelijk wel het geval. Je voorkomt zo ook dat je per ongeluk variabelenamen gebruikt die Mathematica al voor iets anders in gebruik heeft: als je bijvoorbeeld N typt, zie je dat de N zwart wordt (er is immers een functie met die naam), maar als je X typt blijft de X blauw. Deze laatste naam kun je dus voor een nieuwe variabele gebruiken.²

Een erg nuttig soort variabele is de *list*, oftewel: een lijst van variabelen. In Mathematica wordt zo'n list tussen accolades geschreven:

```
In[1] := lijst = {5, 6, 47, 13}
Out[1] = {5, 6, 47, 13}
```

Je ziet hier ook direct dat de naam van een variabele niet altijd uit één letter hoeft te bestaan. Je kunt de naam zo lang maken als je wilt. Dat komt de leesbaarheid van je invoer alleen maar ten goede – al zijn extreem lange namen natuurlijk in het gebruik niet zo handig.

Net zoals bij een losse variabele kun je nu deze lijst op elk moment weer oproepen door eenvoudigweg de naam in te typen:

```
In[1] := lijst
```

²Het is overigens een goede gewoonte om zelfgekozen variabelenamen *niet* met een hoofdletter te laten beginnen. Alle ingebouwde Mathematica-functies hebben namelijk namen die wel met een hoofdletter beginnen; op die manier ontstaat er nooit verwarring.

```
Out[1] = {5, 6, 47, 13}
```

Wil je werken met een enkel element uit de lijst, dan geef je tussen dubbele rechte haken aan welk element je bedoelt:

```
In[1] := lijst[[3]]
```

```
Out[1] = 47
```

Lijsten van lijsten (zogenaamde “geneste” lijsten) zijn ook mogelijk:

```
In[1] := tabel = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}}
```

```
Out[1] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}}
```

is bijvoorbeeld zo’n geneste lijst. Als alle deellijsten even lang zijn (zoals hier: alle deellijsten bevatten drie getallen) heet zo’n lijst een *matrix*. Je kunt een matrix in een wat leesbaardere vorm laten weergeven met behulp van het commando `MatrixForm`:

```
In[1] := MatrixForm[tabel]
```

geeft een nette tabel als uitvoer:

```
Out[1] = 
$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$$

```

Losse elementen uit een geneste lijst geef je weer aan door tussen dubbele rechte haken aan te geven om welk element het gaat. Hierbij werk je “van buiten naar binnen”; bij een matrix vermeld je bijvoorbeeld eerst de rij, en dan de kolom, gescheiden door een komma:

```
In[1] := tabel[[3, 2]]
```

```
Out[1] = 8
```

Vergeet niet om bij het gebruik van lijsten altijd een *dubbele* rechte haak te typen. De enkele rechte haak is in Mathematica gereserveerd voor het aangeven van functie-invoer – bijvoorbeeld `N[Pi]`.

5 Het gebruik van functies

Mathematica kent een groot aantal ingebouwde functies. Vaak hebben die functies de namen die je zou verwachten: de sinus van een hoek bereken je bijvoorbeeld met de functie `Sin`:

```
In[1] := Sin[Pi/3]
```

```
Out[1] =  $\frac{\sqrt{3}}{2}$ 
```

Let hierbij op dat Mathematica verwacht dat je de hoek in radialen³ opgeeft, en niet in graden.

Ben je even vergeten of je een hoek nu in graden of radialen moet opgeven, dan heeft Mathematica een uitgebreide helpfunctie waarin je dit soort zaken kunt opzoeken. Om die helpfunctie te activeren, klik je ergens op het woord **Sin** (zodat de cursor in of net achter de functienaam staat) en druk je op $\langle F1 \rangle$. Mathematica opent dan een uitgebreide hulppagina over de functie. Je ziet op die pagina inderdaad dat je de hoek in radialen moet opgeven, maar dat je met behulp van de aanduiding **Degree** ook graden kunt gebruiken:

```
In[1] := Sin[60 Degree]
```

```
Out[1] =  $\frac{\sqrt{3}}{2}$ 
```

Merk ten slotte op dat Mathematica het antwoord weer niet benadert: in plaats van de benaderde uitkomst 0,866... wordt de *exacte* uitkomst $\frac{1}{2}\sqrt{3}$ gegeven. Wil je een numerieke waarde, dan gebruik je de functie **N** die we al eerder tegenkwamen:

```
In[1] := N[Sin[60 Degree]]
```

```
Out[1] = 0.866025
```

Je ziet dat het voor Mathematica geen probleem is als we functies binnen functies gebruiken: hier wordt de uitkomst van de functie **Sin** aan de functie **N** doorgegeven.

De uitkomst van een functie hoeft niet altijd één getal te zijn. Neem bijvoorbeeld de functie **FactorInteger**, die een geheel getal ontbindt in priemfactoren⁴:

```
In[1] := FactorInteger[144]
```

Deze functie geeft een uitkomst als een lijst van lijsten:

```
Out[1] = {{2, 4}, {3, 2}}
```

Deze uitkomst kunnen we als volgt lezen: 144 heeft een ontbinding in priemfactoren als $2 \times 2 \times 2 \times 2 \times 3 \times 3$, dus het getal heeft vier factoren 2 en twee factoren 3.

Ook de invoer van een functie kan uit meerdere variabelen bestaan. De machtsfunctie **Power[x,y]** rekt bijvoorbeeld de macht x^y uit:

³Ter herinnering: 360 graden is 2π radialen. We rekenen met de bovenstaande invoer dus de sinus van 60 graden uit.

⁴Ter herinnering: priemfactoren zijn delers van een getal die alleen door zichzelf en door 1 te delen zijn.


```
In[1] := Power[3,4]
```

```
Out[1] = 81
```

Mathematica kent enorm veel functies, maar het komt natuurlijk ook wel eens voor dat je zelf een nieuwe functie wilt definiëren. Dat doe je als volgt:

```
In[1] := f[x_] := x^2 - 3x + 5
```

Let hierbij op het volgende:

- Je definieert een functie door een dubbele punt voor het `:=`-teken te zetten.
- De variabele (hier: `x`) zet je tussen enkele rechte haken, en je voegt er een `_` (underscore) aan toe.
- De rechterkant is een uitdrukking in deze variabele (maar zonder de underscore). Je ziet weer dat Mathematica de variabele herkent, en er een andere kleur aan geeft. Deze nieuwe kleur (groen in de meeste versies van Mathematica) geeft aan dat de variabele alleen *locaal* (dus hier: binnen de definitie van de functie) gebruikt wordt. Daarbuiten mag je `x` nog steeds als gewone variabele gebruiken.
- De definitie van een functie heeft geen uitvoer; het maakt dus niet uit of je achter de definitie een puntkomma zet of niet.

Nadat je de functie gedefinieerd hebt, kun je die op precies dezelfde manier gebruiken als de ingebouwde Mathematica-functies. Bijvoorbeeld:

```
In[1] := f[6]
```

```
Out[1] = 23
```

De variabele `x` in de functiedefinitie is een zogenaamde *placeholder* – dat wil zeggen dat Mathematica de variabele vervangt door datgene wat de gebruiker invoert als hij de functie aanroept. Dat kan dus ook een nog niet gebruikte variabele zijn:

```
In[1] := f[z]
```

```
Out[1] = 5 - 3 z + z^2
```

Je ziet overigens dat Mathematica de volgorde van de termen omdraait; het programma gebruikt bepaalde regels om de termen in een uitkomst te ordenen, wat af en toe leidt tot dit soort ongebruikelijke volgordes.

Doordat de variabele `x` als placeholder fungeert, kunt je ook functies in functies invullen:

```
In[1] := f[Sin[90 Degree]]
```

```
Out[1] = 3
```

6 Operaties herhalen

Soms willen we eenzelfde berekening tien keer herhalen, of willen we een som van twintig soortgelijke termen uitrekenen. Mathematica biedt allerlei hulpmiddelen om zulke herhaalde operaties makkelijk uit te voeren. We behandelen er hier twee: `Table` en `Sum`.

Met behulp van `Table` maken we een lijst (tabel) van waarden. Een lijst van de kwadraten van de getallen 1 t/m 10 maken we bijvoorbeeld als volgt:

```
In[1] := kwadraten = Table[n^2, {n, 1, 10}]
Out[1] = {1, 4, 9, 16, 25, 36, 49, 64, 81, 100}
```

De structuur van de functie `Table` is als volgt:

- De eerste invoer in de functie `Table` (hier: n^2) geeft de uitdrukking waarvan je diverse malen de waarde wilt bepalen. In het algemeen zal deze uitdrukking een functie zijn van een bepaalde variabele (hier: n).
- De tweede invoer in de functie `Table` is een lijst van drie elementen (hier: $\{n, 1, 10\}$). Deze drie elementen zijn achtereenvolgens (1) de naam van de variabele (hier: n), (2) de beginwaarde (hier: 1) en (3) de eindwaarde (hier: 10).

Ook hier zal het je weer opvallen dat de locale variabele n een aparte kleur (meestal blauwgroen) krijgt; de variabele komt dus alleen in de functie `Table` als tel-variabele voor, maar kan daarbuiten nog vrij gebruikt worden.

Soms bevat de uitdrukking die wilt herhalen helemaal geen variabele. Dit is bijvoorbeeld het geval als je een lijst van vijf willekeurige (random) getallen wilt maken. Een random geheel getal tussen de 10 en 20 bepaal je bijvoorbeeld met de functie `RandomInteger`:

```
In[1] := RandomInteger[{10, 20}]
Out[1] = 17
```

De uitvoer – in dit voorbeeld 17 – verschilt natuurlijk van aanroep tot aanroep; je kunt dit testen door opnieuw met de cursor achter (of in) de invoer te gaan staan en nog eens op `<shift-enter>` te drukken. Als we nu een lijst van vijf random getallen tussen de 10 en 20 willen maken, kunnen we dat op precies dezelfde manier doen als hierboven:

```
In[1] := Table[RandomInteger[{10, 20}], {n, 1, 5}]
Out[1] = {10, 15, 16, 20, 12}
```

De uitdrukking `RandomInteger[{10, 20}]` hangt echter helemaal niet van de tel-variabele n af, dus het is wat overbodig om deze variabele te gebruiken. In dit geval kunnen we de uitdrukking dan ook wat korter maken:

```
In[1] := Table[RandomInteger[{10, 20}], {5}]
Out[1] = {14, 17, 17, 19, 12}
```

Oftewel:

- Als de eerste invoer in de functie `Table` (hier: `RandomInteger[{10, 20}]`) niet van een variabele afhangt, is het voldoende om in de lijst in de tweede invoer alleen het aantal herhalingen (hier: `{5}`) te noemen.

Het zal je zijn opgevallen dat we de uitvoer van de allereerste invoer in dit hoofdstuk aan een variabele (`kwadraten`) hebben toegekend. We kunnen deze variabele natuurlijk op elk willekeurig moment weer oproepen:

```
In[1] := kwadraten
Out[1] = {1, 4, 9, 16, 25, 36, 49, 64, 81, 100}
```

of we kunnen losse elementen uit deze lijst gebruiken:

```
In[1] := kwadraten[[7]]
Out[1] = 49
```

Alle elementen uit een lijst optellen kan met behulp van de functie `Total`. De som van de eerste tien kwadraten berekenen we dus met

```
In[1] := Total[kwadraten]
Out[1] = 385
```

Er is echter ook een andere manier om herhaalde uitdrukkingen op te tellen. (Dit is een algemeen verschijnsel: voor bijna elk probleem zijn er in Mathematica vele verschillende oplossingen. Welke je kiest is vaak een kwestie van smaak.) Deze manier maakt gebruik van de somfunctie `Sum`:

```
In[1] := Sum[n^2, {n, 1, 10}]
Out[1] = 385
```

De invoer van `Sum` is hetzelfde als die van `Table`: eerst geef je de uitdrukking die je herhaald wilt optellen; vervolgens geeft je een lijst met (1) de naam van de variabele, (2) de startwaarde en (3) de eindwaarde. Het enige verschil in de berekening is dat nu niet eerst een tabel van de individuele kwadraten wordt gemaakt; Mathematica telt de antwoorden direct op.

7 Afgeleiden nemen, integreren en vereenvoudigen

Met Mathematica kun je eenvoudig de afgeleide van een functie nemen. Aangezien het gebruikelijk is om in eenvoudige functies de variabele x te gebruiken, willen we dat hieronder ook doen. Hierbij ontstaat echter het volgende probleem: we hebben aan de variabele x hierboven al een waarde toegekend! (Mocht je dit niet gedaan hebben, doe dat dan alsnog door bijvoorbeeld $x = 6$ in te typen.)

Er gaat nu iets mis als we de afgeleide van een functie van x willen berekenen. Je doet dit met de functie `D`:

```
In[1] := D[x^2 - 3 x + 5, x]
```

Je zult zien dat Mathematica een foutmelding geeft. Om afgeleiden van een functie van x te kunnen nemen, moeten we daarom eerst de variabele x weer “leegmaken”. Dit doen we met de functie `Clear`:

```
In[1] := Clear[x]
```

De functie `Clear` geeft geen uitvoer, maar je ziet wel dat de variabele x overal in het notebook nu weer blauw wordt – Mathematica herkent de variabele nu dus niet meer.

Als we nu nog eens proberen de afgeleide van $x^2 - 3x + 5$ te nemen lukt dat wel:

```
In[1] := D[x^2 - 3 x + 5, x]
```

```
Out[1] = -3 + 2 x
```

De structuur van de functie `D` is als volgt::

- De eerste invoer van `D` (hier: $x^2 - 3x + 5$) is de functie waarvan je de afgeleide wilt nemen,
- De tweede invoer van `D` (hier: x) is de variabele waarvoor je de afgeleide wilt nemen.

Onbepaalde integralen⁵ bepaal je in Mathematica met behulp van de functie `Integrate`:

```
In[1] := Integrate[2 x + 7, x]
```

```
Out[1] = 7 x + x^2
```

De structuur van `Integrate` is vrijwel hetzelfde als die van `D`:

⁵Ter herinnering: een onbepaalde integraal is het omgekeerde van een afgeleide: een onbepaalde integraal van $f(x)$ is een functie $F(x)$ waarvan de afgeleide weer $f(x)$ is. Zo'n functie is niet uniek bepaald – we kunnen er altijd een willekeurige constante bij optellen. $F(x) = x^2$ en $F(x) = x^2 + 1$ hebben bijvoorbeeld dezelfde afgeleide, en zijn dus allebei onbepaalde integralen van $f(x) = 2x$. Mathematica kiest zelf een waarde (in dit geval: 0) voor deze willekeurige constante.

- De eerste invoer van `Integrate` (hier: $2x + 7$) is de functie waarvan je een onbepaalde integraal wilt berekenen,
- De tweede invoer van `Integrate` (hier: x) is de variabele waarvoor je de onbepaalde integraal wilt berekenen.

Mathematica kan ook onbepaalde integralen uitrekenen van functies waarin nog andere variabelen voorkomen. Bijvoorbeeld:

```
In[1] := Integrate[x^m, x]
```

```
Out[1] =  $\frac{x^{1+m}}{1+m}$ 
```

Ook *bepaalde* integralen⁶ bereken je met `Integrate`:

```
In[1] := Integrate[2 x + 7, {x, 0, 1}]
```

```
Out[1] = 8
```

De structuur van `Integrate` voor bepaalde integralen is als volgt:

- De eerste invoer van `Integrate` (hier: $2x + 7$) is de functie waarvan je de bepaalde integraal wilt berekenen,
- De tweede invoer van `Integrate` (hier: $\{x, 0, 1\}$) is een lijst met (1) de variabele waarvoor je de bepaalde integraal wilt berekenen, (2) de beginwaarde, (3) de eindwaarde.

Ook hier kan Mathematica weer omgaan met uitdrukkingen met extra variabelen, maar dit vergt soms wel wat finesse. Probeer bijvoorbeeld eens:

```
In[1] := Integrate[x^m, {x, 0, 1}]
```

De uitvoer die je krijgt hangt van de versie van Mathematica af: in Mathematica 7 krijg je als resultaat

```
Out[1] = If[Re[m]>-1,  $\frac{1}{1+m}$ , Integrate[x^m, {x, 0, 1}],
  Assumptions -> Re[m] <= -1]
```

en Mathematica 8 vertelt je

```
Out[1] = ConditionalExpression[ $\frac{1}{1+m}$ , Re[m] > -1]
```

De precieze betekenis van deze antwoorden doet er niet heel veel toe; het moge duidelijk zijn dat er hier iets mis gaat. Wat er mis gaat is het volgende: Mathematica herkent dat de

⁶Ter herinnering: met een bepaalde integraal bereken je het oppervlak onder een grafiek. De bepaalde integraal $\int_0^1 (2x + 7)dx$ geeft bijvoorbeeld het oppervlak onder de grafiek van $f(x) = 2x + 7$ tussen $x = 0$ en $x = 1$.

variabele m waarden kan aannemen waarvoor de bepaalde integraal oneindig wordt. Dat is bijvoorbeeld het geval voor $m = -1$. We moeten Mathematica dus vertellen dat we alleen geïnteresseerd zijn in waarden van m waarvoor de integraal wel berekend kan worden. Dat is bijvoorbeeld zo voor alle waarden van m groter dan 0. We kunnen deze aanname met behulp van de optie `Assumptions` aan de functie `Integrate` doorgeven:

```
In[1] := Integrate[x^m, {x, 0, 1}, Assumptions -> m > 0]
Out[1] = 1/(1+m)
```

Het pijltje `->` typ je door een `-` en een `>` achter elkaar te typen; nadat je nog een volgend teken hebt getypt, zie je dat Mathematica deze combinatie vanzelf door een pijltje vervangt.

Nu is de uitkomst wel zoals we die zouden verwachten. We zien hier verder een algemeen patroon in Mathematica: aan veel functies (zoals hier `Integrate`) kun je na de “verplichte” invoer nog een aantal opties doorgeven, gescheiden door komma’s, en van de vorm

`Optie -> waarde`

Let op: je plaatst zulke opties altijd *aan het eind* van de functie, na de gewone argumenten. Welke opties je aan een functie kunt meegeven, achterhaal je door de helpfunctie (cursor op de functienaam zetten en `<F1>` indrukken) te gebruiken. Je vindt dan in de helptekst een kopje “Options”; klikken op het pijltje daarnaast geeft een lijst van opties, en nogmaals klikken op het pijltje naast een bepaalde optie geeft meer informatie en meestal wat voorbeelden.

Ook in dit geval kent Mathematica weer diverse oplossingen voor eenzelfde probleem. Laten we hier een tweede oplossing bespreken die vaak erg nuttig is. We doen nogmaals dezelfde berekening, maar geven het antwoord nu een naam:

```
In[1] := resultaat = Integrate[x^m, {x, 0, 1}]
```

Je krijgt weer dezelfde uitvoer als hiervoor. We kunnen Mathematica nu als volgt de opdracht geven het resultaat te vereenvoudigen:

```
In[1] := Simplify[resultaat, m > 0]
Out[1] = 1/(1+m)
```

De structuur van `Simplify` is als volgt:

- De eerste invoer in `Simplify` (hier: `resultaat`) is de uitdrukking die we willen vereenvoudigen.
- De tweede invoer in `Simplify` (hier: `m > 0`) is een aanname (of een lijst van aannames) die gebruikt mag worden om het resultaat te vereenvoudigen.

De aannames zijn niet altijd nodig; veel uitdrukkingen kan Mathematica ook zonder extra aannames vereenvoudigen. Zo kent Mathematica bijvoorbeeld de bekende relatie $\sin^2(x) + \cos^2(x) = 1$:

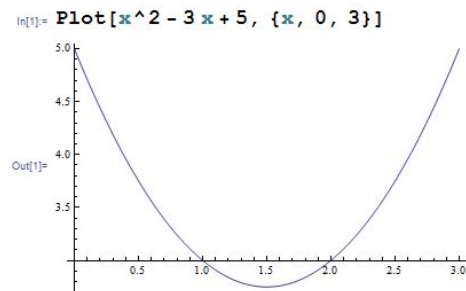
```
In[1] := Simplify[Sin[x]^2 + Cos[x]^2]
```

```
Out[1] = 1
```

8 Grafieken maken

Met behulp van de functie `Plot` kunnen we Mathematica een grafiek van een functie laten maken:

```
In[1] := Plot[x^2 - 3 x + 5, {x, 0, 3}]
```



Afbeelding 2: De grafiek van de functie $f(x) = x^2 - 3x + 5$.

Het resultaat zie je in afbeelding 2. De structuur van `Plot` zal je inmiddels vertrouwd voorkomen:

- De eerste invoer in `Plot` (hier: $x^2 - 3x + 5$) is de functie waarvan we een grafiek willen tekenen.
- De tweede invoer in `Plot` (hier: $\{x, 0, 3\}$) is een lijst met (1) de variabele waarvoor je de grafiek wilt tekenen, (2) de beginwaarde, (3) de eindwaarde.

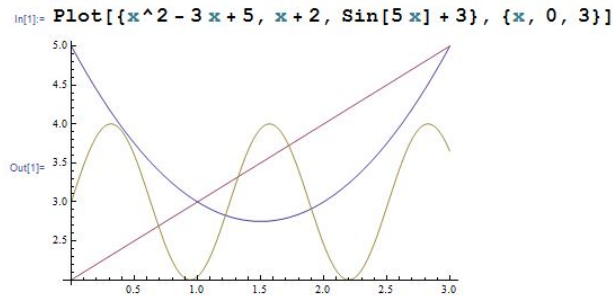
Het zal je misschien opvallen dat Mathematica zelf de nodige keuzes maakt in het tekenen van de grafiek: de plaats van de assen (de x -as wordt hier bijvoorbeeld niet zoals gebruikelijk getekend op de hoogte $y = 0$, maar op de hoogte $y = 3$), het bereik van de y -as, de kleur van de grafiek, enzovoort. Meestal is Mathematica slim genoeg om dergelijke keuzes goed te maken, maar vind je de uitvoer niet mooi, dan kun je vrijwel alles met behulp van opties aanpassen. Bijvoorbeeld:

```
In[1] := Plot[x^2 - 3 x + 5, {x, 0, 3}, AxesOrigin -> {0, 0}]
```

geeft de grafiek weer met de assen op de gebruikelijke plaats. Gebruik weer de helpfunctie om te zien welke opties er allemaal zijn.

Mathematica kan ook meerdere grafieken in één afbeelding tekenen. Hiervoor vervang je simpelweg de functie uit het vorige voorbeeld door een lijst (tussen accolades) van functies. Let op: alle functies moeten daarbij dezelfde variabele gebruiken! Bijvoorbeeld:

```
In[1] := Plot[{x^2 - 3 x + 5, x + 2, Sin[5 x] + 3}, {x, 0, 3}]
```

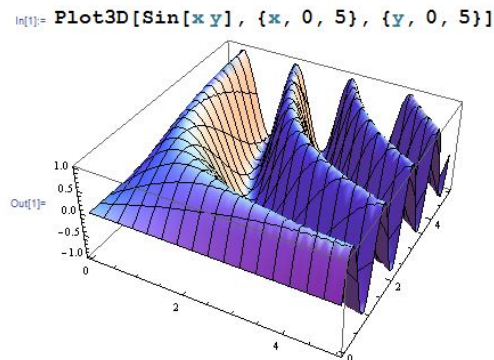


Afbeelding 3: De grafiek van de functies $f(x) = x^2 - 3x + 5$ (blauw), $g(x) = x + 2$ (paars) en $h(x) = \sin(5x) + 3$ (bruin).

Het resultaat zie je in afbeelding 3 – je ziet dat Mathematica elke grafiek een andere kleur geeft, zodat ze goed te onderscheiden zijn. Mathematica vertelt alleen niet welke grafiek bij welke functie hoort; om dat te bepalen moet je dus even nadenken, of eerst de grafieken afzonderlijk tekenen.

Van functies van twee variabelen teken je de grafiek met `Plot3D`:

```
In[1] := Plot3D[Sin[x y], {x, 0, 5}, {y, 0, 5}]
```



Afbeelding 4: De grafiek van de functie $f(x, y) = \sin(xy)$.

Vergeet niet een spatie tussen x en y in `Sin[x y]` te typen, zodat Mathematica begrijpt dat het om twee variabelen gaat, en niet om één variabele met de naam xy ! Het resultaat zie je in afbeelding 4. De structuur van `Plot3D` is als volgt:

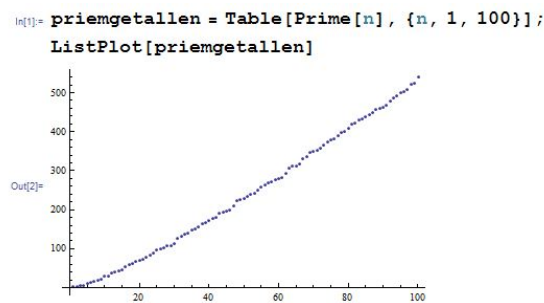
- De eerste invoer in `Plot3D` (hier: `Sin[x y]`) is de functie waarvan we een grafiek willen tekenen.
- De tweede invoer in `Plot3D` (hier: `{x, 0, 5}`) is een lijst met (1) de eerste variabele waarvoor je de grafiek wilt tekenen, (2) de beginwaarde, (3) de eindwaarde.
- De derde invoer in `Plot3D` (hier: `{y, 0, 5}`) is een lijst met (1) de tweede variabele waarvoor je de grafiek wilt tekenen, (2) de beginwaarde, (3) de eindwaarde.

Mathematica kan ook plots maken van lijsten met getallen. Laten we bijvoorbeeld eens een grafiek maken van de eerste 100 priemgetallen. Hiervoor gebruiken we de functie `Prime[n]`, wat het n -de priemgetal geeft. Een lijst van de eerste tien priemgetallen maken we bijvoorbeeld met

```
In[1] := Table[Prime[n], {n, 1, 10}]
Out[1] = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29}
```

Om een grafiek te maken van de eerste 100 priemgetallen gebruiken we vervolgens de functie `ListPlot`:

```
In[1] := priemgetallen = Table[Prime[n], {n, 1, 100}];
ListPlot[priemgetallen]
```



Afbeelding 5: Een grafiek van de eerste 100 priemgetallen.

Het resultaat is te zien in figuur 5. Aan de functie `ListPlot` hoeven we zoals je ziet alleen de lijst door te geven die we willen plotten. Ook hier zijn er natuurlijk weer allerlei opties die we aan de functie kunnen meegeven om de grafiek mooier te maken. Gebruik indien nodig de helpfunctie om een overzicht te zien.

9 Zelf Mathematica installeren

Mathematica is helaas geen goedkoop programma – op dit moment (november 2013) kost een privéversie van het programma €295. Er is ook een studentenversie voor ongeveer de helft van die prijs; voor die versie moet je kunnen aantonen dat je scholier of student bent. Dat kan bijvoorbeeld met een gescande collegekaart of een kopie van je lesrooster. Bijna alle universiteiten en sommige scholen hebben ook een eigen licentienummer, waarmee je Mathematica gratis voor eigen gebruik thuis of op een laptop kunt installeren.

Als geen van die mogelijkheden voor jou geldt, kun je Mathematica in elk geval 30 dagen gratis thuis proberen. Ga daarvoor naar

<http://www.wolfram.com/mathematica/trial/>

en maak een account aan. Nadat je dit gedaan hebt kun je een trial-versie van Mathematica downloaden die 30 dagen lang werkt.

10 Meer leren

Deze handleiding is gebaseerd op de Engelstalige handleiding “Getting started with Mathematica” van Martin Erickson. Handleidingen zoals deze zijn op het internet in overvloed te vinden: google op iets als “introduction mathematica” en je vindt er vanzelf een aantal.

Er zijn ook diverse boeken over Mathematica. Het klassieke boek over het programma is “The Mathematica Book”, geschreven door Stephen Wolfram, de maker van het programma. Dit boek is echter voor het laatst uitgegeven voor Mathematica versie 5 (inmiddels zijn we bij versie 9), en dus niet meer heel erg up-to-date. Het boek is nog steeds wel een goede manier om een inleiding in het programma te krijgen; aan de basisbeginselen is in de afgelopen versies niet veel veranderd.

Wil je meer recente informatie, dan kun je het best de website van Mathematica gebruiken: via de links

<http://www.wolfram.com/support/learn/>

<http://reference.wolfram.com/mathematica/guide/Mathematica.html>

vind je enorm veel informatie in de vorm van tutorials, voorbeelden, video’s, enzovoort.

Tenslotte: de beste manier om Mathematica te leren kennen is natuurlijk door het programma veel te gebruiken. Zoek je een functie die iets doet, dan blijkt vaak dat die functie een heel voor de hand liggende naam heeft. Levert wat proberen toch niet de juiste functie op of is het je niet helemaal duidelijk welke variabelen of opties een bepaalde functie gebruikt, dan biedt de helpfunctie $\langle F1 \rangle$ vaak uitkomst.

Hopelijk heb je na het lezen van deze handleiding in elk geval de smaak van het werken met Mathematica te pakken. Veel succes met het verder ontdekken van het programma!