

Maple T.A. Items Maken Deel B

**Copyright © Metha Kamminga
jan 2017**

Maple T.A. Items Maken Deel B

Contents

| | |
|---|----|
| 1 Items maken Deel B | 1 |
| 1.1 Informatie en downloads | 1 |
| 1.2 Formules met Maple T.A. | 1 |
| 1.2.1 Inleiding | 1 |
| 1.2.2 Instellingen van het formule-invulveld | 1 |
| 1.2.2.1 Instelling Maple syntax | 2 |
| 1.2.2.1.1 Text Mode bij Maple syntax | 3 |
| 1.2.2.1.2 Symbol Mode bij Maple syntax | 7 |
| 1.2.2.2 Instelling Formula | 13 |
| 1.2.2.2.1 Text Mode bij Formula | 14 |
| 1.2.2.2.2 Symbol Mode bij de instelling Formula | 17 |
| 1.2.3 Formules voorbereiden | 18 |
| 1.2.3.1 Formules en getallen voorbereiden in het Algorithm | 18 |
| 1.2.3.2 Formules met HTML | 21 |
| 1.2.3.3 Wat is MathML-code | 22 |
| 1.2.3.4 Vier manieren om MathML-code te maken | 22 |
| 1.2.3.5 MathML-code programmeren in rubriek Algorithm | 22 |
| 1.2.3.5.1 Automatische vereenvoudiging tegenhouden | 29 |
| 1.2.3.6 MathML met de Equation Editor | 33 |
| 1.2.3.7 MathML-code met MathType | 36 |
| 1.2.3.8 MathML met Wiris | 37 |
| 1.2.3.9 LaTeX | 38 |
| 1.2.3.9.1 Variabelen, spaties en tekst | 40 |
| 1.2.3.9.2 Superscript en subscript | 41 |
| 1.2.3.9.3 Vectoren en onderstrepingen | 41 |
| 1.2.3.9.4 Vectoren en matrices | 41 |
| 1.2.3.9.5 Functies en commando's cursief en niet cursief | 42 |
| 1.2.3.9.6 Breuken en wortels | 43 |
| 1.2.3.9.7 Ongelijkheden | 43 |
| 1.2.3.9.8 Limieten, sommatie en producten | 43 |
| 1.2.3.9.9 Differentiëren en integreren | 44 |
| 1.2.3.9.10 Haakjes en pijltjes | 45 |
| 1.2.3.9.11 Tekens en Griekse letters | 45 |
| 1.3 Vraagtype Maple-graded | 45 |
| 1.3.1 Algemene structuur van Maple-graded vraagtype binnen de Question Designer | 46 |
| 1.3.2 Algorithme van het Maple-graded vraagveld | 48 |
| 1.3.3 Question Text van het Maple-graded vraagveld | 50 |
| 1.3.4 Answer van het Maple-graded vraagveld | 50 |
| 1.3.5 Grading Code van het Maple-graded vraagveld | 52 |
| 1.3.6 Expression type van het Maple-graded vraagveld | 53 |
| 1.3.7 Maple Repository van het Maple-graded vraagveld | 53 |
| 1.3.8 Plotting Code van het Maple-graded vraagveld | 53 |
| 1.3.9 Custom Previewing van de Maple-graded vraag | 54 |
| 1.3.10 Feedback van de vraag | 55 |
| 1.4 Tips voor de Grading Code | 56 |
| 1.4.1 Getallen overhoren met Maple-graded | 58 |
| 1.4.2 Wanneer zijn formules gelijk aan elkaar? | 58 |
| 1.4.3 Werken met StringTools | 61 |
| 1.4.4 Automatische vereenvoudiging bij Preview tegenhouden | 64 |
| 1.4.5 Custom Previewing Code | 65 |
| 1.4.6 De Analyse van de Response | 65 |

| | |
|---|-----|
| 1.4.7 Coëfficiënten en operanden | 69 |
| 1.4.8 Verzamelingen | 72 |
| 1.4.9 Breuken | 73 |
| 1.4.10 Uitdrukkingen | 79 |
| 1.4.11 Uitdrukkingen met de letter I | 83 |
| 1.4.12 Wortelvormen | 85 |
| 1.4.13 Vergelijkingen | 96 |
| 1.4.14 Stelsels vergelijkingen | 99 |
| 1.4.15 Ongeordende en geordende lijsten | 105 |
| 1.4.16 Ongelijkheden | 106 |
| 1.4.17 Griekse letters | 113 |
| 1.4.18 Differentialen | 114 |
| 1.4.19 Integralen | 117 |
| 1.4.19.1 Onbepaalde integraal | 117 |
| 1.4.19.2 Bepaalde integraal | 119 |
| 1.4.19.3 Meervoudige integralen | 119 |
| 1.4.19.4 De student voert de integraal in | 120 |
| 1.4.20 Matrices en Vectoren | 124 |
| 1.4.20.1 Inleiding | 124 |
| 1.4.20.2 Analyse van de invoer | 125 |
| 1.4.20.3 Voorbeeld | 128 |
| 1.4.20.4 Random matrix | 129 |
| 1.5 Vraagtype Formula | 129 |
| 1.5.1 Formula | 130 |
| 1.5.2 Formula without Simplification | 132 |
| 1.5.3 Formula that matches responses to within +C | 134 |
| 1.5.4 Lijsten en Vectoren met Formula | 134 |
| 1.5.5 Vergelijkingen met Formula | 136 |
| 1.5.6 Chemische formules met Formula | 137 |
| 1.6 Vraagtype Numeric | 138 |
| 1.7 Dynamische figuren | 138 |
| 1.8 Maple-commando's | 138 |
| 1.8.1 Algemene commando's | 138 |
| Index | 143 |

List of Figures

| | |
|---|----|
| Figure 1.1: Instellingen bij het Maple-graded vraagtype | 2 |
| Figure 1.2: Instellingen bij het Maple-graded vraagtype binnen de Question Designer | 2 |
| Figure 1.3: Maple syntax in de Text Mode met Invalid Maple syntax | 3 |
| Figure 1.4: Maple syntax in de Text Mode let op syntax en controleer met Perview | 4 |
| Figure 1.5: Maple syntax in de Text Mode na grading | 5 |
| Figure 1.6: Griekse letters bij de instelling Text Mode | 6 |
| Figure 1.7: Krachtige Preview-functie bij Maple syntax Text entry only | 6 |
| Figure 1.8: Previewfunctie met tegengehouden vereenvoudiging | 7 |
| Figure 1.9: Maple syntax in Symbol Mode met Editor | 8 |
| Figure 1.10: Maple syntax in de Symbol Mode na grading | 9 |
| Figure 1.11: Met of zonder de Editor een Maple-graded vraag | 10 |
| Figure 1.12: Met de formule-Editor aangeboden, geef antwoord ook in 2D | 11 |
| Figure 1.13: Verandering van variabele in de differentiaal | 12 |
| Figure 1.14: Matrix invoeren | 13 |
| Figure 1.15: Instelling Formula bij Maple-graded vraagtype | 13 |
| Figure 1.16: Instellingen voor Formula met mogelijkheid om zelf te kiezen voor Text Mode of Symbol Mode | 14 |
| Figure 1.17: Grading bij Formula met Text Mode | 15 |
| Figure 1.18: De Preview bij de instelling Formula in de Text Mode | 16 |
| Figure 1.19: De Preview bij de instelling Maple syntax in de Text Mode | 16 |
| Figure 1.20: Instelling voor Formula met de formule Editor (Symbol Mode) | 17 |
| Figure 1.21: Grading met gebruik van de Editor (Symbol Mode) bij Formula-instelling | 18 |
| Figure 1.22: Formules in Algorithm | 19 |
| Figure 1.23: Negatieve waarden van de variabelen | 20 |
| Figure 1.24: Gedaante van getallen | 20 |
| Figure 1.25: HTML in de broncode van de tekst van de vraag | 21 |
| Figure 1.26: html-code in het Algorithm voorbereiden | 22 |
| Figure 1.27: MathML-code | 22 |
| Figure 1.28: Formule en MathML voorbereiden in Algorithm | 23 |
| Figure 1.29: Verschillende manieren om MathML-code te maken | 24 |
| Figure 1.30: MathML met subscript en lettercombinaties | 25 |
| Figure 1.31: MathML-Formules met Maple gemaakt | 26 |
| Figure 1.32: MathML coderen met combinaties van methoden | 27 |
| Figure 1.33: MathML voorbereiden in gedeelten | 28 |
| Figure 1.34: Een functie definiëren | 28 |
| Figure 1.35: Automatische vereenvoudiging tegenhouden | 29 |
| Figure 1.36: Gebruik van quotes bij de MathML conversie van Maple | 30 |
| Figure 1.37: MathML en quotes | 30 |
| Figure 1.38: MathML en quotes | 31 |
| Figure 1.39: Imaginaire eenheid | 32 |
| Figure 1.40: Custom PlottingCode | 32 |
| Figure 1.41: Formules met quotes | 33 |
| Figure 1.42: De Equation Editor | 34 |
| Figure 1.43: Vernieuwde Equation Editor in Maple T.A. 10 | 35 |
| Figure 1.44: MathML-code aanpassen in de broncode van de vraag | 36 |
| Figure 1.45: Met MathType de MathML-code maken | 37 |
| Figure 1.46: Formules met Wiris | 38 |
| Figure 1.47: LaTeX met de Equation Editor | 39 |
| Figure 1.48: voorbeeld met LaTeX | 39 |
| Figure 1.49: LateX in het Algorithmme voorbereiden | 40 |
| Figure 1.50: LaTeX met variabelen spaties en tekst | 40 |
| Figure 1.51: LaTeX met superscript en subscript | 41 |

| | |
|--|----|
| Figure 1.52: LaTeX met vectoren en onderstreping | 41 |
| Figure 1.53: Kolomvectoren en matrices met LaTeX | 42 |
| Figure 1.54: LaTeX met functies en commando's | 42 |
| Figure 1.55: LaTeX met cursief en niet-cursief | 43 |
| Figure 1.56: LaTeX met breuken en wortels | 43 |
| Figure 1.57: LaTeX met limieten | 43 |
| Figure 1.58: LaTeX met sommatie en product | 43 |
| Figure 1.59: LaTeX met gecentreerde formule | 44 |
| Figure 1.60: LaTeX met integralen | 44 |
| Figure 1.61: LaTeX met kringintegraal | 44 |
| Figure 1.62: LaTeX en differentiëren | 44 |
| Figure 1.63: partiële differentiatie | 45 |
| Figure 1.64: LaTeX en haakjes | 45 |
| Figure 1.65: LaTeX en tekens en Griekse letters | 45 |
| Figure 1.66: Voorbeeld van een Maple-graded vraag | 46 |
| Figure 1.67: Question Designer vraag | 47 |
| Figure 1.68: Maple-graded vraag aanmaken | 47 |
| Figure 1.69: title of the figure | 48 |
| Figure 1.70: Het algoritme van een Maple graded vraagtype | 49 |
| Figure 1.71: Tekst van de Maple Graded vraag | 50 |
| Figure 1.72: Het correcte antwoord in 2D | 51 |
| Figure 1.73: Grading Code van de Maple-graded vraag | 52 |
| Figure 1.74: Instellingen voor Maple syntax bij Maple-graded vraagtype | 53 |
| Figure 1.75: Plotting Code in de Maple-graded vraag | 54 |
| Figure 1.76: Custom Previewing Code | 55 |
| Figure 1.77: Feedback bij de Maple-graded vraag | 55 |
| Figure 1.78: Grading van een Maple-graded vraag met feedback | 56 |
| Figure 1.79: Instellingen met Maple Syntax | 62 |
| Figure 1.80: Vereenvoudigen | 63 |
| Figure 1.81: Tegenhouden van automatische vereenvoudiging in de Preview door gebruikmaking van de Custom Previewing Code | 65 |
| Figure 1.82: Instellingen voor het antwoord | 66 |
| Figure 1.83: Genereren van de string van de RESPONSE | 66 |
| Figure 1.84: Het zichtbaar maken van de "\$RESPONSE" met Text entry only | 67 |
| Figure 1.85: Het zichtbaar maken van de "\$RESPONSE" met Symbol entry only | 67 |
| Figure 1.86: Het zichtbaar maken van de "\$RESPONSE" met Symbol entry only | 68 |
| Figure 1.87: Het zichtbaar maken van de "\$RESPONSE" met Formula | 68 |
| Figure 1.88: Een polynoom met decimalen matchen | 70 |
| Figure 1.89: Breuken met instelling Text entry only | 74 |
| Figure 1.90: Breuken met instelling Symbol entry only | 75 |
| Figure 1.91: Breuken invoeren in de Editor | 76 |
| Figure 1.92: Breuken invoeren als text bij de instelling Formula | 76 |
| Figure 1.93: Breuken met twee invulvelden voor teller en noemer | 78 |
| Figure 1.94: Haakjes wegwerken | 79 |
| Figure 1.95: Uitdrukking met decimale coëfficiënten | 80 |
| Figure 1.96: Grading code voor de coëfficiënten van een uitdrukking | 81 |
| Figure 1.97: Lineaire uitdrukking met decimalen | 82 |
| Figure 1.98: De letter I niet als imaginaire eenheid | 84 |
| Figure 1.99: Berekeningen met complexe getallen | 84 |
| Figure 1.100: Hoe worden wortels vereenvoudigd | 85 |
| Figure 1.101: De respons en de stringrespons bij verschillende manieren van intikken van de wortels | 86 |
| Figure 1.102: Wortels vereenvoudigen | 87 |
| Figure 1.103: Wortels vereenvoudigen in Algorithm | 88 |
| Figure 1.104: De wortelvorm vereenvoudigen | 89 |

| | |
|--|-----|
| Figure 1.105: title of the figure | 89 |
| Figure 1.106: Splitsen van wortelvormen | 90 |
| Figure 1.107: De operanden van een wortelvorm | 90 |
| Figure 1.108: De analyse van de wortelvorm in de Editor ingevoerd | 91 |
| Figure 1.109: Wortelvorm met Editor | 92 |
| Figure 1.110: Algorithm bij Wortelvorm met Editor | 92 |
| Figure 1.111: Wortelvormen checken met operanden van het antwoord | 93 |
| Figure 1.112: Algorithm van Wortelvormen checken met operanden | 93 |
| Figure 1.113: Wortels vereenvoudigen | 94 |
| Figure 1.114: Wortels met breuken vereenvoudigen | 95 |
| Figure 1.115: Lineair systeem, vergelijking invullen | 96 |
| Figure 1.116: Vergelijkingen met decimalen | 97 |
| Figure 1.117: Vergelijking van een vervormde cirkel | 98 |
| Figure 1.118: Stelsel vergelijkingen | 100 |
| Figure 1.119: Stelsel vergelijkingen met verzameling en met lijst | 100 |
| Figure 1.120: Stelsel vergelijkingen met de variabelen apart | 101 |
| Figure 1.121: Stelsel met meer oplossingen | 102 |
| Figure 1.122: Stelsel vergelijkingen met meer oplossingen Algorithm | 102 |
| Figure 1.123: Stelsel vergelijkingen met meer oplossingen numeriek | 103 |
| Figure 1.124: Algorithm van een vraag over het berekenen van snijpunten van grafieken | 104 |
| Figure 1.125: title of the figure | 105 |
| Figure 1.126: Lineaire ongelijkheid | 107 |
| Figure 1.127: Simpele ongelijkheid | 108 |
| Figure 1.128: title of the figure | 109 |
| Figure 1.129: Kwadratische ongelijkheid | 110 |
| Figure 1.130: Oplossing van ongelijkheid met verzamelingen matchen | 110 |
| Figure 1.131: Ongelijkheid met intervallen | 112 |
| Figure 1.132: Griekse letters invoeren | 113 |
| Figure 1.133: Griekse letters invoegen in het antwoordveld | 114 |
| Figure 1.134: De student tikt een differentiaalvergelijking in | 115 |
| Figure 1.135: Differentiaal symbolisch met text invoer | 116 |
| Figure 1.136: Hogere afgeleiden met het tekstveld | 116 |
| Figure 1.137: Hogere afgeleiden met de Editor | 117 |
| Figure 1.138: Subscript in de Editor | 117 |
| Figure 1.139: Primitiveren, de onbepaalde integraal | 118 |
| Figure 1.140: De bepaalde integraal berekenen | 119 |
| Figure 1.141: Dubbelintegraal berekenen | 120 |
| Figure 1.142: De analyse van het antwoord van de student in de vorm van een integraal in Symbol mode | 120 |
| Figure 1.143: De analyse van het antwoord van de student in de vorm van een integraal in Text mode | 121 |
| Figure 1.144: Invoeren van een integraal | 122 |
| Figure 1.145: Invoeren van een integraal symbolisch | 123 |
| Figure 1.146: Verschil in invoer met Int en int en geen verschil in de Preview | 124 |
| Figure 1.147: Het invoeren van een Matrix door de student in de Editor | 125 |
| Figure 1.148: Invoer van matrices vertaald naar de string | 126 |
| Figure 1.149: Invoer van matrices | 127 |
| Figure 1.150: Matrix invoeren | 128 |
| Figure 1.151: Subtypes van het vraagtype Formula | 130 |
| Figure 1.152: Change Entry Style | 131 |
| Figure 1.153: Vraagtype Formula in de Question Designer | 131 |
| Figure 1.154: Vraagtype Maple-graded met instelling Formula | 132 |
| Figure 1.155: Limieten met wortelvorm | 133 |
| Figure 1.156: Vraagtype Formula met wortels | 133 |
| Figure 1.157: Preview met wortels | 134 |
| Figure 1.158: Question Designer Lijsten en Vectoren | 135 |

| | |
|---|-----|
| Figure 1.159: De subtypen van Formula in de Question Designer | 136 |
| Figure 1.160: Vergelijkingen met het vraagtype Formula | 137 |
| Figure 1.161: Chemische formules | 138 |

1 Items maken Deel B

© Metha Kamminga
Update febr. 2016

1.1 Informatie en downloads

- **Gratis Handleidingen** op de website van Metha Kamminga: <http://www.methakamminga.nl/MapleTA/>
In deze handleidingen gaat het over het maken van toets items on-line.
- **MapleSoft**
Kijk ook eens op <http://www.maplesoft.com/products/mapleta/> voor demo's en question banks.
- **Home page Metha Kamminga**
Meer informatie over cursussen en nieuws over Maple T.A. op de site van Metha Kamminga: <http://www.methakamminga.nl>

1.2 Formules met Maple T.A.

1.2.1 Inleiding

Het Maple T.A.-toetsstelsel is zeer krachtig in het gebruik van formules doordat er elk moment gebruikgemaakt kan worden van het onderliggende computeralgebrasysteem Maple dat eraan gekoppeld is.

Nieuw in versie 10 van Maple T.A. is dat het gebruik van formules nu helemaal onafhankelijk van Java is door de nieuwe programmering gebaseerd op html5.

- Formules kunnen in de tekst van de vraag opgenomen worden, of in de feedback of in de hints en dergelijke, waarbij het computeralgebrasysteem Maple ten dienste staat van het genereren van deze formules.
- Vanaf versie 10 is het aanzien van de formules in de tekst van de vraag aanzienlijk verbeterd.
- Het is zelfs mogelijk om formules te toetsen, waarbij studenten zelf formules kunnen invoeren in de invulvelden. Omdat het bij studenten erg gevoelig ligt met het intikken van formules, waarbij gemakkelijk syntax-fouten kunnen optreden, is het aan de maker van de vraag om zo goed mogelijk de instellingen te doen, zodat de student het systeem als gebruiksvriendelijk ervaart.
Het computeralgebrasysteem Maple komt er ongemerkt aan te pas om grading te verzorgen. De formule van de student wordt vergeleken met de juiste formule. Er zijn vele instellingen mogelijk om verschillende soorten formules te toetsen. Zie daarvoor in paragraaf *Tips voor de Grading Code* (page 56).
- Bij formules is het natuurlijk ook weer interessant om veel gebruik te maken van randomisering waarin het systeem bijzonder sterk is. Zie ook de *Handleiding Randomiseren*.
- Een waarschuwing is hier op zijn plaats. Doordat er zoveel mogelijkheden zijn om instellingen te doen met betrekking tot de manier van invullen door de student, is het belangrijk om nieuwe vragen goed uit te proberen en te kijken of alle instellingen naar wens zijn. In de volgende paragrafen worden er veel oplossingen aangedragen voor situaties die u in de praktijk kunt tegenkomen.
- Ten slotte is het belangrijk dat studenten enige training krijgen in het invoeren van formules. Zie ook op de website het document *Overzicht Formules Invoeren* bij de tips voor studenten.

1.2.2 Instellingen van het formule-invulveld

Op verschillende manieren kan de student een formule als antwoord invoeren in een zogenaamde *Free Response Area* binnen een vraag.

We maken in dit document kennis met verschillende soorten invulvelden en vraagtypen en noemen daarbij de voor- en nadelen ervan. Voor het toetsen van formules staat globaal een aantal van *twee* vraagtypen ter beschikking. Het is het vraagtype *Maple-graded* (page 45) en het vraagtype *Formula* (of *Mathematical Formula*) (page 129).

Deze vraagtypen vindt u ook terug in de *Question Designer*.

De instellingen voor het invoeren van formules bij deze vraagtypen kunnen verschillend zijn.

1.2.2.1 Instelling Maple syntax

Bij het *Maple-graded* vraagtype zijn twee instellingen mogelijk: *Maple syntax* en *Formula* (zie de figuur hieronder). Maak in het algemeen NIET gebruik van de instelling *Formula*!!

Er is een *groot* verschil tussen beide instellingen en dat wordt in de volgende paragrafen uiteengezet.

Select the type of expression you want to accept:

Maple syntax - e.g. diff(2*f(x),x)

Formula - e.g. x^2 sin(x^2)

Maple syntax - e.g. diff(2*f(x),x)

Text/Symbolic entry

Text entry only

Text entry only

Symbolic entry only

Student can choose

Figure 1.1: Instellingen bij het Maple-graded vraagtype

Ook in het vraagtype *Question Designer* kan een invulveld gedefinieerd worden van het type *Maple-graded* en daarbij zijn dezelfde instellingen mogelijk namelijk *Maple syntax* en *Formula*. Ook hier geldt: maak in het algemeen NIET gebruik van *Formula*. Zie de figuur hieronder waar *Maple syntax* geselecteerd is met *Text entry only*.

Edit Response Area

Choose Question Type

- Essay
- Free Body Diagram
- List
- Maple-graded
- Math App
- Mathematical formula
- Multiple Choice
- Numeric
- Sketch

Maple-graded:

Weighting

Answer

Grading Code:

Optional:

Maple Repository:

Plotting Code:

01 diff

differentieer de volgende functie

$$f(t) = \frac{2}{t^2 - 16}$$

\$f

(referenced when grading as \$ANSWER)

```
evalb(simplify(($f)-($RESPONSE)) = 0) and
evalb(0=StringTools[Search]("diff",$RESPONSE)) ;
```

Maple Syntax - e.g. diff(2*f(x),x)

Text entry only

Text entry only

Symbolic entry only

Student can choose

Maple Repository

plot(\$RESPONSE,x=-5..5,y=-100..100,thickness=2)

OK

Cancel

Figure 1.2: Instellingen bij het Maple-graded vraagtype binnen de Question Designer

Bij het vraagtype *Maple-graded* met de instelling *Maple syntax* kan de student de formule intikken met de officiële syntax, waarbij erg veel mogelijk is. Lettercombinaties die als één variabele gezien worden, matrices, differentiaalvergelijkingen, integralen, werken met subscript, werken met functies enzovoort. Het vereist dan wel een goede kennis van de juiste syntax bij de student. Het systeem is daarin nogal streng en bovendien hoofdlettergevoelig, maar het heeft zéér veel voordelen op het gebied van een breed scala van eenvoudige wiskundevraagstukken tot complexe fysische en andere toegepaste vraagstukken.

De student kan op *Preview* (het vergrootglasje) klikken om zijn ingetikte formule te controleren. Zie ook het document op de website *Overzicht Formules Invoeren* bij de *Tips voor studenten*.

TIP: Hier is te zien dat de *Grading Code* erin voorziet dat de student in zijn antwoord niet gebruikmaakt van het Maple-commando `diff`. Ook is te zien dat bij de *Plotting Code* er nog de mogelijkheid aangeboden wordt aan de student om zijn antwoord te visualiseren (met de `P` van `plot`).

1.2.2.1.1 Text Mode bij Maple syntax

Bij het *Maple-graded* vraagtype kan de instelling gezet worden op *Maple syntax* met de mogelijkheid om de invoer te doen in de *Text Mode* (*Text entry only*) óf met behulp van een Editor, de *Symbol Mode* (*Symbolic entry only*). Echter de student kan bij de instelling *Maple syntax* niet zelf kiezen voor *Symbol Mode* of *Text Mode*. Bij de instellingen *Maple syntax* wordt de keuze tussen deze twee namelijk door de bouwer van de vraag afgedwongen bij de instelling *Text/Symbolic entry*.

In de volgende figuur is te zien hoe de *Text Mode* er voor de student uit ziet. De student moet de formule intikken in *Maple syntax* en kan vervolgens op *Preview* klikken.

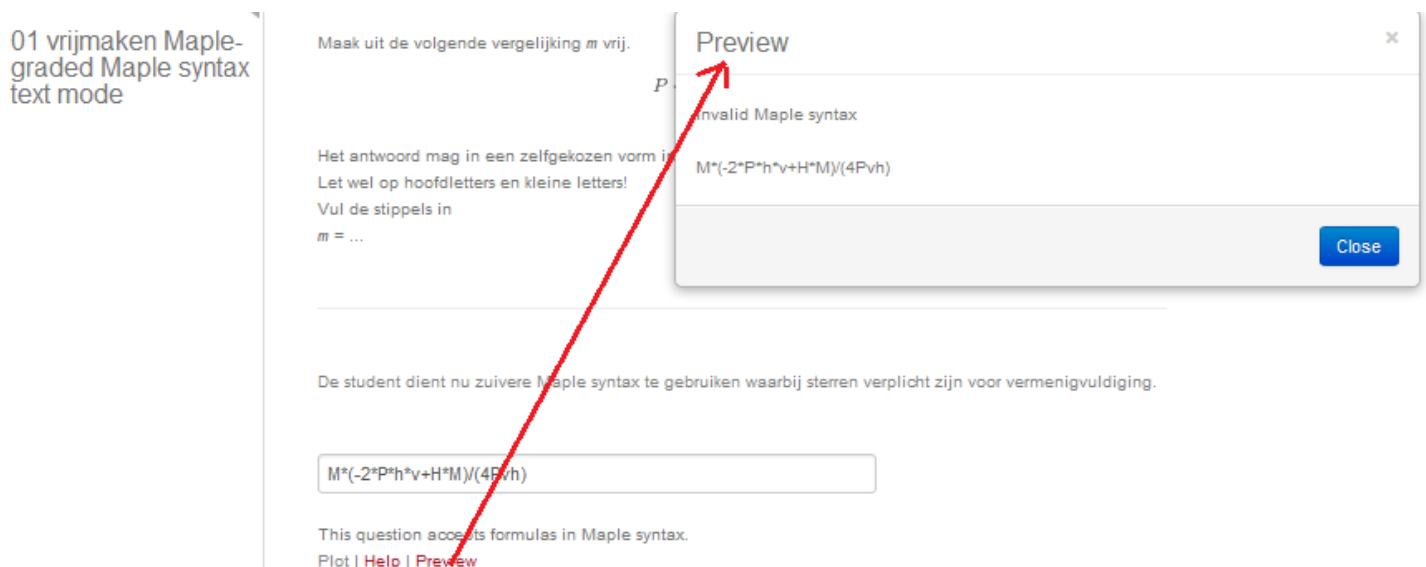


Figure 1.3: Maple syntax in de Text Mode met Invalid Maple syntax

De manier waarop de formule is ingetikt in bovenstaande figuur, *Figure 1.3* (page 3), is niet de juiste en daarvan wordt melding gemaakt bij het klikken op *Preview*. Het is namelijk niet geoorloofd om de 4 vlak voor de letter *P* te tikken als er $4 \times P$ bedoeld wordt. De student had in dit geval $4*P$ moeten typen.

In de volgende figuur is te zien dat het systeem in de *Preview* de letters *Pvh* als één variabele opvat in de noemer van de breuk. Immers in de *Preview* staan er dan ook geen spaties tussen.

Maak uit de volgende vergelijking m vrij.

$$P = \frac{M^2 H}{v(6M + 5m)h}$$

Het antwoord mag in een zelfgekozen vorm ingetikt worden.
 Let wel op hoofdletters en kleine letters!
 Vul de stipfels in
 $m = \dots$

$$\frac{1}{5} \frac{M(-6Pvh + MH)}{Pvh}$$

Close

$$M*(-6*P*v*h+M*H)/(5*Pvh)$$

This question accepts formulas in Maple syntax.

Plot | [Help](#) | [Preview](#)

Figure 1.4: Maple syntax in de Text Mode let op syntax en controleer met Preview

In de figuur hierboven zal het antwoord bij de grading dan ook niet als correct worden opgevat omdat het systeem in de noemer niet een vermenigvuldiging herkent $P v h$. Echter er komt geen syntax error in dit geval. Er hadden sterren tussen de letters getikt moeten worden om het systeem te laten weten dat het wel om een vermenigvuldiging gaat en niet om een variabele bestaande uit een aantal karakters aan elkaar gekoppeld.

Als die sterren er wel tussengezet waren, dan is het antwoord correct, zie *Figure 1.4 (page 4)* en waren de letters in de *Preview* vertaald door letters met spaties ertussen.

TIP: Spaties gelden bij de instelling *Text Mode* niet als vermenigvuldiging. Bij *Maple syntax* mogen er aan weerskanten van operatoren wel spaties getikt worden. Studenten doen dat soms links en rechts van het $+$ -teken of iets dergelijks. Deze worden volkomen genegeerd. Echter voor vermenigvuldiging moet beslist een ster getikt worden. De lettercombinatie $v h$ met een spatie ertussen wordt bij de *Preview* teruggegeven als *Invalid Maple syntax* en dan kan de student de spatie ertussen weghalen of een ster tikken als hij dat bedoeld had.

Let wel op dat bij de Editor de spatie tussen twee letters wél gezien wordt als een vermenigvuldiging! zoals uitgelegd in paragraaf *Symbol Mode (page 7)*.

In onderstaande *Figure 1.5 (page 5)* is na de grading achteraf bij *Your Answer* precies te zien wat de invoer van de student is geweest. In de feedback bij *Comment* kan het juiste antwoord als formule gecommuniceerd worden.

01 vrijmaken Maple-graded Maple syntax text mode

Grade: 1.0

Maak uit de volgende vergelijking m vrij.

$$P = \frac{M^2 H}{v(5M + 3m)h}$$

Het antwoord mag in een zelfgekozen vorm ingetikt worden.

Let wel op hoofdletters en kleine letters!

Vul de stippels in

$m = \dots$



Correct

De student dient nu zuivere Maple syntax te gebruiken waarbij sterren verplicht zijn voor vermenigvuldiging.

Your Answer: $M*(-5*P*h*v+H*M)/(3*P*v*h)$

Correct Answer: $1/3*M*(-5*P*h*v+H*M)/P/v/h$

Comment: Het goede antwoord is $\frac{1}{3} \frac{M(-5Phv+HM)}{Pvh}$

Figure 1.5: Maple syntax in de Text Mode na grading

TIP: In deze vraag is in het ontwerp van de vraag het correcte antwoord ingevuld in de rubriek *Answer*. Het correcte antwoord wordt hier echter in de Feedback (Comment) gegeven zoals dat daar ook geprogrammeerd is. Zie ook figuur *Figure 1.70 (page 49)* in de paragraaf over de algemene structuur van een Maple-graded vraag hoe het goede antwoord in het ontwerp van de vraag ingevuld moet worden.

TIP: Het grote voordeel van deze *Text Mode* bij het *Maple-graded* vraagtype is dat het voor de student volstrekt duidelijk is wat hij heeft ingetikt bij de controle met *Preview*. En ook achteraf bij *Your Answer*. Het toetsen van formules met lettercombinaties eventueel ook met subscript, is allemaal mogelijk. Het vraagtype *Maple-graded* heeft dan nog extra voordelen dat u in de *Grading Code* precies kunt programmeren aan welke eisen het antwoord van de student moet voldoen. Zie meer informatie in de paragraaf *Tips voor de Grading Code (page 56)*.

TIP: Het nadeel van deze instelling is echter dat de invoer in deze situatie aan strenge regels onderhevig is en dat er enige training voor het intikken van formules vereist wordt. De student moet bijvoorbeeld weten dat hij $\exp(x)$ in moet tikken als hij de exponentiële functie bedoelt. In de *Preview* komt dan ook netjes e^x te staan.

Subscript kan ingetikt worden met $x[1]$ en in de *Preview* komt dan te staan x_1 .

Functies zoals $\sin(x)$, $\ln(x)$ en $\exp(x)$ moeten beslist met haakjes ingetikt worden.

Functies in het algemeen worden op deze manier altijd met haakjes herkend. De invoer $f(x)$ wordt dan ook vertaald als zijnde een functie f van x en niet als een vermenigvuldiging $f*x$ of als lettercombinatie fx . In alle gevallen krijgt u géén syntax error want alle manieren zijn geoorloofd binnen de Maple syntax alleen de betekenis van deze drie manieren is verschillend.

TIP: Voor het invullen van Griekse letters kan de *Editor* ook gebruikt worden. Voor de hoofdletters van het Griekse alfabet niet voor alle letters, gebruik eventueel de *tekst entry only* waarbij de student moet weten dat bijvoorbeeld Delta de hoofdletter Δ is. In de *Preview* komt dan ook de echte Δ te staan, zoals in de figuur hieronder.

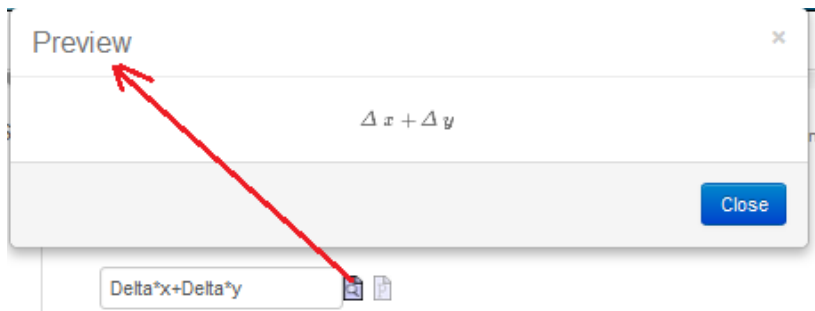


Figure 1.6: Griekse letters bij de instelling Text Mode

De syntax moet bij de studenten dus helemaal helder zijn, maar dat alles hoeft op zich geen nadeel te zijn, want u krijgt er heel veel mogelijkheden voor terug en eventueel gaat u over op de instelling *Symbol entry only* waarbij de student minder afhankelijk is van zijn kennis van de syntax maar waar de student wel degelijk even instructie voor moet krijgen (zie paragraaf (page 7)).

TIP: Een ander nadeel van deze instelling van *Maple syntax Text Mode* is dat de student ook Maple-commando's zou kunnen hanteren om tot het juiste antwoord te komen. Denk bijvoorbeeld aan het oplossen van een vergelijking waarbij u er van uitgaat dat de student de oplossing zelf berekent. Een slimme student kan ook met het commando `solve` werken om toch een grading van 100% te krijgen. Echter dat kunt u ondervangen door te eisen dat er in het antwoord van de student `solve` niet voorkomt. U kunt ook de instelling zetten op *Symbol Mode* (zoals besproken wordt in de volgende paragraaf *Symbol Mode* (page 7)) waar geen commando's geaccepteerd worden. Zie verder bij de paragraaf *Vraagtype Maple-graded* (page 45).

TIP: Nog een nadeel is van de instelling *Text entry only* dat de *Preview*-knop zó krachtig is dat het vaak een vereenvoudigde *Preview* geeft. Dit heeft natuurlijk nadelen als u juist de vereenvoudiging toetst. Als de student bijvoorbeeld intikt: $(3*x+9)/3$, dan is in de *Preview* te zien: $x + 3$.

Of als de student intikt $5/(a/8)$ dan is in de *Preview* te zien $\frac{40}{a}$.

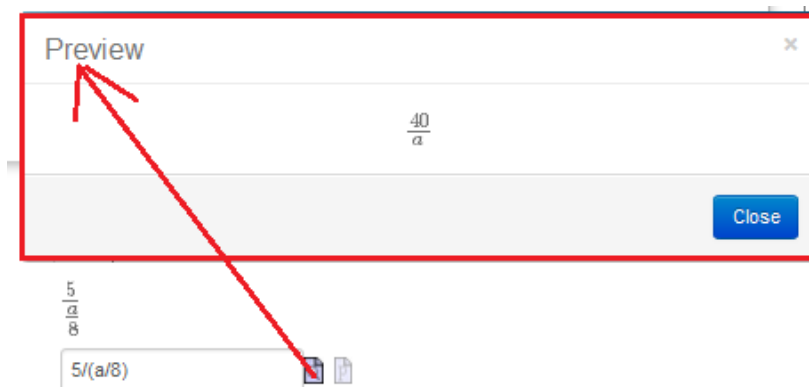


Figure 1.7: Krachtige Preview-functie bij Maple syntax Text entry only

TIP: Een oplossing om te voorkomen dat de student middels de *Preview* de vereenvoudiging voorgezegd krijgt, is de instelling *Symbol Mode* aan te bieden waar de *Preview*-knop niet aanwezig is, omdat die daar ook niet nodig is (zie volgende paragraaf *Symbol Mode* (page 7)).

TIP: Een andere oplossing is te kiezen voor de instelling *Formula* en dus niet voor *Maple syntax*. Bij de instelling *Formula* is de *Preview*-knop niet zo krachtig, maar heeft wel weer andere (grote) nadelen, zie paragraaf *Instelling Formula* (page 13).

TIP: Nog een mogelijkheid voor dit soort gevallen is om in de *Custom Previewing* het volgende te programmeren:

```
use InertForm:-NoSimpl in $RESPONSE: end: printf(InertForm:-ToMathML(%));
```

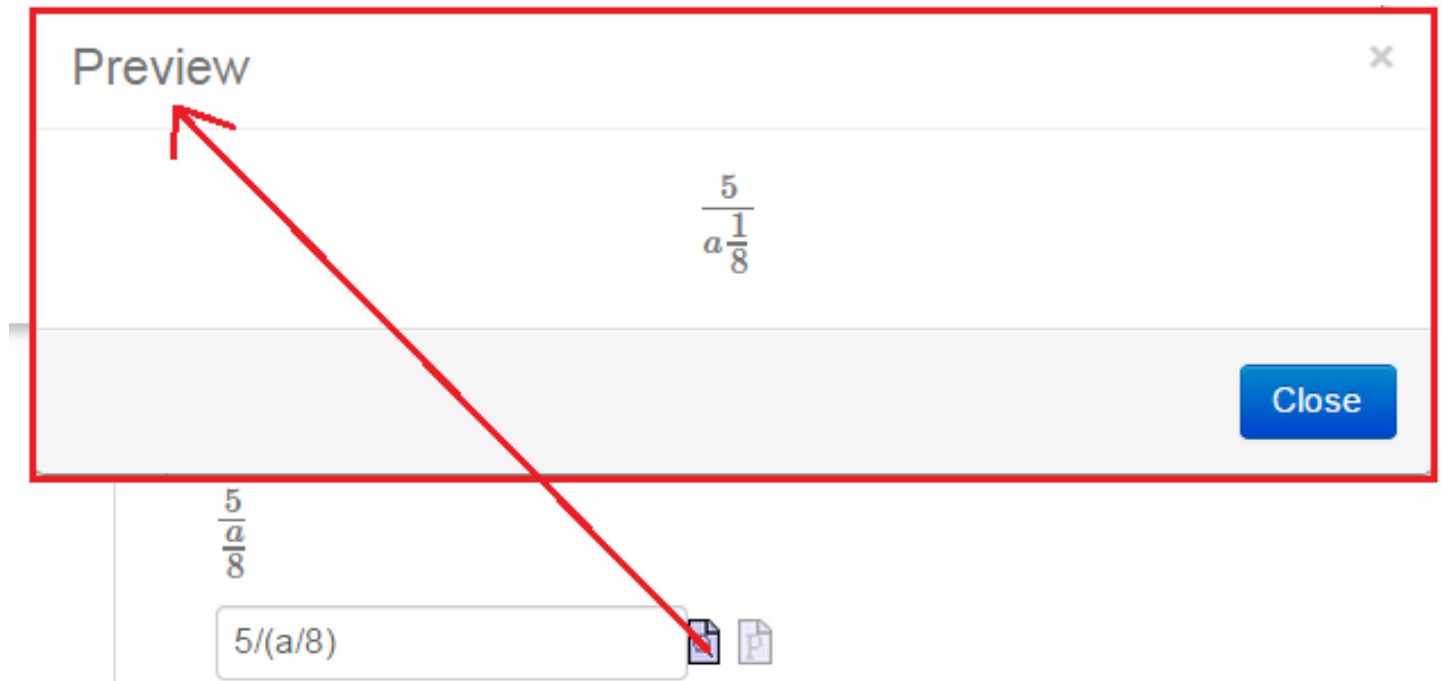


Figure 1.8: Previewfunctie met tegengehouden vereenvoudiging

TIP: In vorige versies van Maple T.A. was die *Preview* nog niet zo krachtig en eventueel moet u in uw oude bestanden bepaalde vragen met *Maple syntax* waarbij *Text entry only* is ingesteld deze veranderen naar *Symbol entry only* of anderszins maatregelen nemen.

Voor overgaan van text naar symbol: in de broncode hoeft u alleen te veranderen: `allow2d=0` vervangen door `allow2d=2`.

1.2.2.1.2 Symbol Mode bij Maple syntax

Het is mogelijk om bij het vraagtype *Maple-graded* met de instelling *Maple syntax* een Editor aan te bieden (*Symbol Mode*). Deze Editor vertaalt de formule op een goede manier naar *Maple syntax* mits goed ingevoerd.

(MapleSoft is al gewaarschuwd maar bij de feedback wordt niet juist weergegeven wat de student had ingevoerd, hoewel de grading wel betrouwbaar is.)

Het aanbieden van de Editor wordt afgedwongen door de bouwer van de vraag. De student kan zelf niet kiezen tussen de *Text Mode* en de *Symbol Mode*.

In onderstaande figuur is een dergelijke vraag te zien zoals de student die voor zich krijgt in een Assignment.

Equation Editor

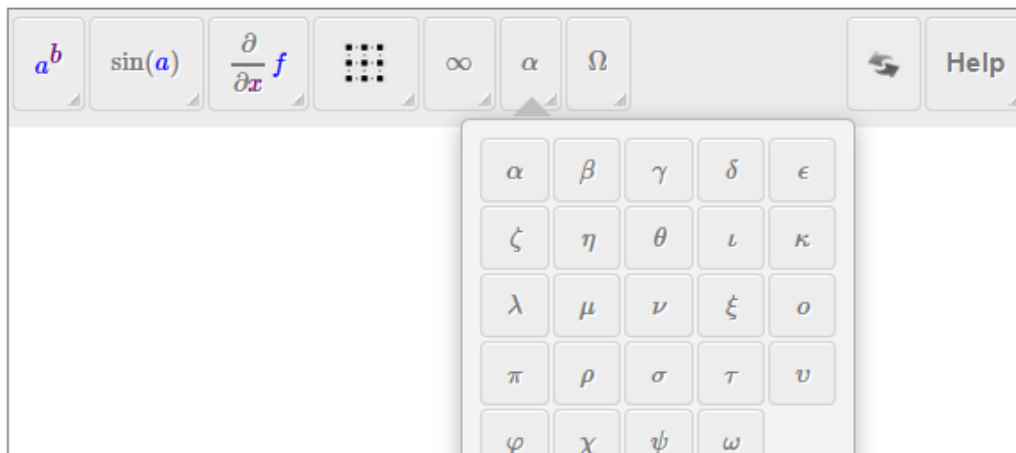


Figure 1.9: Maple syntax in Symbol Mode met Editor

In bovenstaande figuur is de Editor te zien die *Maple syntax* aanbiedt met *Symbol Mode*. Ook subscript is mogelijk en integralen en differentiaal en matrices en zo meer. Voor Griekse letters zijn nog een paar Griekse hoofdletters erbij gekomen!

TIP: De student moet wel degelijk óók leren omgaan met deze *Editor*. Integralen, differentiaal, subscript, matrices, Griekse letters en dergelijk is allemaal mogelijk. Let wel op dat de letters en symbolen alleen separaat gebruikt kunnen worden. De lettercombinatie Δx bijvoorbeeld zal niet werken om te kunnen graden.

Verder moet de student leren dat als zijn formule mislukt en hij wil opnieuw beginnen, dat hij dan het pijltjesteken voor vernieuwen (rechts in de Editor) gebruikt om de Editor weer even schoon te vegen, zodat er geen onzichtbare codes meer in blijven zitten.

TIP: Nog een bijzonderheid is bij de **Griekse letters** dat de letter Zeta (ζ) en de letter Pi (π) opgevat worden als de officiële functie Zeta en de officiële waarde voor Pi = 3.14.... en dus met een hoofdletter in de formule van het correcte antwoord meegenomen moeten worden voor een juiste grading. De ontwerper van de vraag heeft hier rekening mee te houden, maar studenten merken er niets van.

TIP: Voor een vermenigvuldiging moet de student een ster tikken (die op het scherm als een stip wordt gepresenteerd) óf hij moet een spatie tikken als hij een vermenigvuldiging bedoelt. Als de student géén spatie of ster tikt, wordt bijvoorbeeld Pvh ook niet als een vermenigvuldiging gezien, maar wordt deze lettercombinatie als één geheel gezien wat veel voordelen biedt bij het gebruik van formules in de toepassingsfeer. Vermenigvuldiging van matrices en vectoren kan de student doen met de gewone punt van het toetsenbord. (Maar let op dat als u juist de vermenigvuldiging van matrices vraagt, dat de student voor een 100%-grading dus ook de punt kan gebruiken. Zie ook de paragraaf *Tips voor de Grading Code* (page 56).

TIP: Voor het invoeren van een formule zoals $M(a + b)$ moet vlak vóór het haakje openen $M(\dots)$ de student beslist een spatie of een ster tikken, want hier wordt een vermenigvuldiging bedoeld. Echter zonder spatie of ster wordt het niet als vermenigvuldiging opgevat en kunt u zodoende bijvoorbeeld ook functies toetsen.

De uitdrukking $f(x)$ wordt dan niet automatisch vertaald naar een vermenigvuldiging $f*x$, wat ook weer voordelen heeft voor het toetsen van allerlei toepassingen met functies. (Zie paragraaf *Differentiaal* (page 114) waar ook meer informatie over het veranderen van de variabele bij integratie en differentiatie.)

Echter als de student $6P$ zonder spatie tikt, wordt dat in deze Editor weer wél als vermenigvuldiging gezien. In geval de student $P6$ intikt wordt dit juist weer niet als vermenigvuldiging opgevat en $P6$ kan dan ook als één geheel worden gebruikt in een formule. Alle standaardfuncties worden op de bekende manier ingevoerd zoals $\sin(x)$, $\exp(x)$ en $\ln(x)$, met haakjes dus.

In de volgende figuur wordt getoond wat de student te zien krijgt na grading. Bij *Your Answer* ziet de student nu in 2D wat hij letterlijk heeft ingetikt (dus in de Editor waar geen *Preview*-knop meer nodig is).

Grade: 100%

Maak uit de volgende vergelijking m vrij.

$$P = \frac{M^2 H}{v(6 M + 3 m) h}$$



Correct

Het antwoord mag in een zelfgekozen vorm ingetikt worden.

Let wel op hoofdletters en kleine letters!

Vul de stippels in

$m = \dots$

Your Answer: $\frac{M \cdot (-6 \cdot P \cdot v \cdot h + M \cdot H)}{3 \cdot P \cdot v \cdot h}$

Comment: Het goede antwoord is $\frac{1}{3} \frac{M (-6 P v h + M H)}{P v h}$

Figure 1.10: Maple syntax in de Symbol Mode na grading

Wat de student heeft ingetikt is na afloop bij de grading te zien. Hier is bijvoorbeeld te zien dat de student hier en daar een ster getikt heeft wat vertaald wordt met een stip in de Editor. Na de 3 in de noemer is automatisch een spatie ingevoerd ook al heeft de student dat misschien niet gedaan.

TIP: In deze vraag is in het ontwerp van de vraag het correcte antwoord niet ingevuld. Het correcte antwoord is hier echter wel in de Feedback (*Comment*) geprogrammeerd. Zie ook *Figure 1.70 (page 49)* waar en hoe het goede antwoord in het ontwerp van de vraag ingevuld moet worden.

In het volgende voorbeeld zien we in één vraag twee invulvelden waarbij de eerste keer géén Editor aangeboden wordt en bij het tweede gedeelte wel. Let hierbij op dat bijvoorbeeld yz één variabele is bestaande uit een combinatie van karakters. Het betekent dus NIET een vermenigvuldiging $y \cdot z$. De student mag dan ook géén spatie of ster tikken.

04 simplify verschillende instellingen

Vereenvoudig de volgende formules:

$$-8 \, qr + 14 \, yz + 14 + 4 \, qr - 10 \, yz - 8$$

Met *Maple syntax Text entry only*, kan de student op *Preview* klikken en dan kijken wat de vereenv. *Preview*-knop werkt heel krachtig voor vereenvoudiging.

Dezelfde vraag nu maar dan wordt de automatische vereenvoudiging van de *Preview* tegengehouden, overtekken geeft een foute beoordeling:

$$14 \, b - 10 \, qr + 4 \, b - 10 \, qr - 8$$

Equation Editor

| | | | | | | | | | |
|-------|-----------|---------------------------------|---|----------|----------|----------|--|--|------|
| a^b | $\sin(a)$ | $\frac{\partial}{\partial x} f$ | $\begin{matrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{matrix}$ | ∞ | α | Ω | | | Help |
|-------|-----------|---------------------------------|---|----------|----------|----------|--|--|------|

Figure 1.11: Met of zonder de Editor een Maple-graded vraag

Zie ook nog meer uitleg bij dit voorbeeld in *Figure 1.80 (page 63)*.

Nog een derde manier om te voorkomen dat de *Preview* het vereenvoudigde antwoord al geeft is om in de *Custom Previewing* het volgende te programmeren:

```
use InertForm:-NoSimpl in $RESPONSE: end: printf(InertForm:-ToMathML(%));
```

Een goede gewoonte is het als u bij het aanbieden van de Editor ook het goede antwoord in 2D programmeert. Dat kan in de rubriek *Answer* met de Maple-opdracht: `printf(MathML[ExportPresentation](.....));` zoals in de figuur hieronder te zien is.

Edit Response Area

Choose Question Type

- Essay
- Free Body Diagram
- List
- Maple-graded
- Math App
- Mathematical formula
- Multiple Choice
- Numeric
- Sketch

Maple-graded:

Weighting:

Answer

(referenced when grading as \$ANSWER)

Grading Code:

Expression Type:

Text/Symbolic entry:

Optional:

Maple Repository:

Plotting Code:

Custom Previewing Code:

OK Cancel

Figure 1.12: Met de formule-Editor aangeboden, geef antwoord ook in 2D

TIP: Het voordeel van deze Editor (*Symbol Mode*) bij *Maple syntax* is dat de student ook echt de formule tweedimensionaal moet intikken en dat maple-commando's niet als zodanig opgevat worden. De student kan dus niet bijvoorbeeld het commando `solve` gebruiken om toch een goede beoordeling te krijgen bij het oplossen van een vergelijking. Ook wordt er in de Editor geen automatische vereenvoudiging toegepast, zoals wel bij *Preview* van de *Text Mode* gebeurt (zie *Figure 1.11* (page 10)

Zie ook nog meer uitleg bij dit voorbeeld in *Figure 1.80* (page 63).

Het werken met de Editor vereist beslist ook enige training voor de student.

TIP: Spaties rond operatoren geven geen probleem, deze worden genegeerd, net als bij de *Text Mode* van *Maple syntax*. Kijk ook in paragraaf *Differentiaalvergelijkingen* (page 114) waar een differentiaalvergelijking wordt ingevoerd in de Editor en waar $\frac{d}{dx}$ een operator is. Aan weerszijden hiervan mogen eventueel spaties getikt worden die toch genegeerd worden.

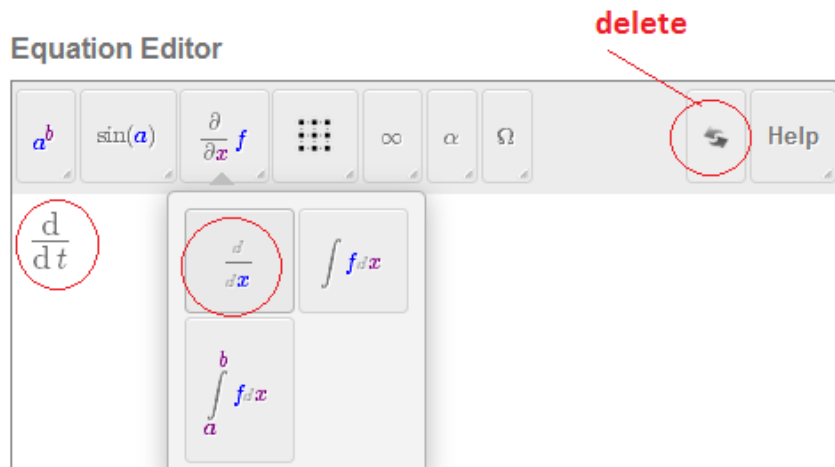


Figure 1.13: Verandering van variabele in de differentiaal

Let op dat als je de variabele wilt veranderen, dat je dan eerst achter de "x" gaat staan in de editor, daar een "t" neerzet en dan pas de x weghalen. De "d" moet niet cursief worden, anders wordt de code niet verstaan door het systeem.

TIP: Wat de student ook moet weten is, dat als hij een antwoord intikt in de Editor en dit weer weg wil hebben, dat dit moet gebeuren met het selecteren van het geheel en dan deleten. Beter is nog met de knop van het dubbele pijltje .

Als hij alleen de backspace-knop gebruikt om iets weg te halen, kan het zijn dat er nog onbedoelde code achterblijft waardoor het kan gebeuren dat de grading niet goed verloopt.

TIP: Bij het overhoren van Matrices en Vectoren moeten studenten goed opgevoed worden met name dat hij op voorhand de juiste dimensies van de Matrix kiest en die invult.

In het volgende voorbeeld moet de student een 3×4 -matrix invoeren.

06 Lineair systeem matrix

Gegeven zijn de volgende drie vergelijkingen met de onbekenden x , y en z .

$$\begin{cases} 11x + 9z + 9y = 11 \\ -9x - 6y + 9z = 7 \\ -7x + 10z + 13y = -5 \end{cases}$$

Geef de aangevulde 3 bij 4 matrix van het lineaire systeem ten opzichte van de onbekenden $[x, y, z]$.

In dit voorbeeld maakt het niet uit of de volgorde van de vergelijkingen verwisseld wordt of dat de vergelijking vereenvoudigd zijn of niet. De coëfficiëntenmatrix aangevuld met de bijbehorende rechterlid-kolom geeft een oplossing van het systeem.

Equation Editor

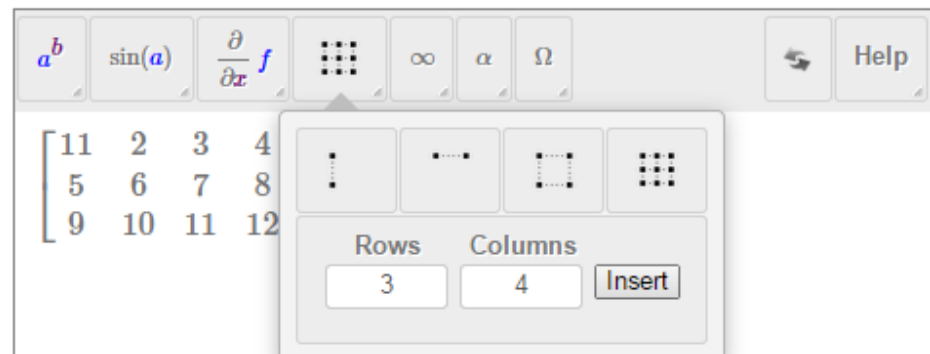


Figure 1.14: Matrix invoeren

TIP: De student kan ook matrices en vectoren met elkaar vermenigvuldigen door er een punt tussen te zetten. Dat is de gewone punt van het toetsenbord. Evenzo ook optellen en aftrekken. Let wel op dat in de *Grading Code* erin voorzien moet worden dat de student dit kan doen!

Zie verder in paragraaf *Matrices en Vectoren* (page 124) voor dit onderwerp wat betreft de *Grading Code*.

Zie ook paragraaf *Differentialen* (page 114) voor het invoeren van differentiaal.

Zie ook paragraaf *Integralen* (page 117).

1.2.2.2 Instelling Formula

Bij het vraagtype *Maple-graded* (en bij het vraagtype *Mathematical Formula* zie paragraaf *Vraagtype Formula* (page 129)), kunt u de instelling voor het intikken van het antwoord op *Formula* zetten. U moet hierbij wel bedenken dat u deze instelling alleen gebruikt in bijzondere omstandigheden want deze instelling kan niet veel formules aan en is in de meeste gevallen zelfs af te raden.

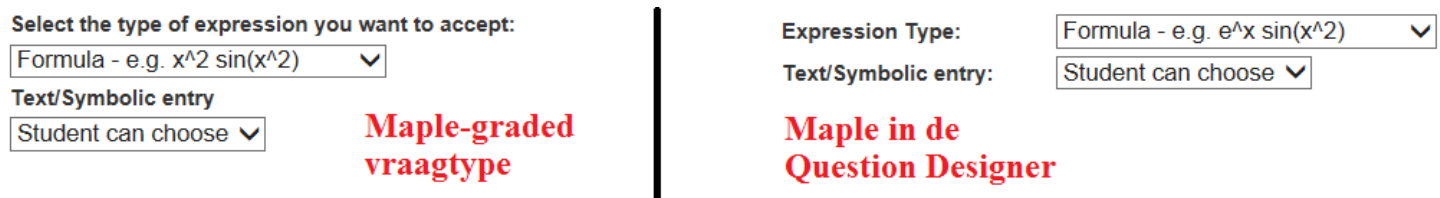


Figure 1.15: Instelling Formula bij Maple-graded vraagtype

Het is daarbij mogelijk dat de student zelf kiest (Student can choose) voor *Text Mode* óf voor *Symbol Mode* met de beschikbare knop in de vraag: *Change Math Entry Mode*.

Het is niet aan te raden om bij de instellingen van *Formula*, gebruik te maken van de Editor met de *Symbol Mode*, omdat er door de verschillende conversies die er dan plaatsvinden, miscommunicatie kan ontstaan. Bovendien wordt daarmee ook niet het goede begrip van de operatoren ontwikkeld bij de studenten en het gaat beslist niet sneller met deze Editor. Echter soms heeft de instelling *Formula* bij *Maple syntax* voordelen, zie paragraaf *Vraagtype Formula* (page 129).

TIP: De Editor die bij de instelling *Formula* wordt aangeboden is niet erg betrouwbaar en veel minder uitgebreid dan de Editor die bij *Maple syntax* wordt aangeboden! Dus ook weer een reden om liever de instelling *Formula* niet te gebruiken.

TIP: Zie voor het vraagtype *Mathematical Formula* in betreffende paragraaf (page 129). Daar kan namelijk voor de instellingen van het type uitdrukking gekozen worden voor *Formula* en daarvoor geldt hetzelfde verhaal als hieronder, maar bij dat vraagtype zijn er nog meer mogelijke instellingen voor andere typen uitdrukkingen, zoals al of niet geordende lijsten of scheikundige formules en dergelijke. Echter de aangeboden Editor van de *Symbol Mode* is zeer beperkt en is aangepast aan de instellingen van dit vraagtype.

1.2.2.2.1 Text Mode bij Formula

Bij de instelling *Text Mode* die de student zelf kiest bij de instelling *Formula*, krijgt hij het volgende te zien:

Question Name: 01 vrijmaken Maple-graded

Maak uit de volgende vergelijking m vrij.

$$P = \frac{M^2 H}{v(3 M + 5 m) h}$$

Het antwoord mag in een zelfgekozen vorm ingetikt worden.
Let wel op hoofdletters en kleine letters!
Vul de stippels in
 $m = \dots$

$M^2(-3P^2v^2h+MH)/(5P^2vh)$

This question accepts numbers or formulas.
Plot ([Help](#)) [Change Math Entry Mode](#) ([Preview](#))

Maple T.A. - Preview Math - Windows I
http://mapleta.can.nl:8080/mapleta/m

Equation Editor Modes: You have a choice of two modes for entering answers to math questions:

Text Mode: In text mode, "x squared divided by y" looks like this:

$$(x^2)/y$$

- Equations are typed in from the keyboard
- Text mode is quick and easy to use in any browser

Symbol Mode: In symbol mode, "x squared divided by y" looks like this:

$$\frac{x^2}{y}$$

- You need to download a tool to enter equations
- This mode is optimized for Internet Explorer (version 5.0 and later) on Windows and may not work well in other browsers.

Change Equation Editor Mode:

☒ Symbol Mode $\frac{x^2}{y}$

☐ Text Mode $(x^2)/y$

Note: Your change will take effect with the next question you view.

Maple-graded in de Question Designer met instelling *Formula*

Change entry mode

Bij de instellingen *Formula* is de *Preview*-knop niet zo krachtig, maar de *Preview* ziet er niet goed uit. Ten eerste komen er veel te veel haakjes en ten tweede worden lettercombinaties als vermenigvuldigingen gezien. Dus niet met de instelling *Formula* werken. Echter de

• The system default mode is **Text** (4.2 Choosing a Math Entry Mode) using syntax similar to a graphing calculator, combined with a **Preview** option, which allows you to see what your

Figure 1.16: Instellingen voor Formula met mogelijkheid om zelf te kiezen voor Text Mode of Symbol Mode

In bovenstaande figuur is te zien dat de student een opdracht krijgt om iets met een formule te doen.

Als de student klikt op *Help*, dan komt er een pop-up scherm met informatie over het intikken van de formule. Er staat dan ook bij: *This question accepts numbers or formulas*. De student kan nu de door hem berekende formule gaan intikken waarbij het helemaal niet erg is als er af en toe sterren of spaties voor vermenigvuldiging vergeten worden. Alle ingetikte letters worden door het systeem als afzonderlijke variabelen beschouwd. Het systeem zet er in feite zelf de sterren tussen. Dat is ook te zien als de student iets in het invulveld tikt en vervolgens op *Preview* klikt, waarmee een pop-up wordt aangeboden met de ingetikte formule 2-dimensionaal. Daarin is inderdaad te zien dat bijvoorbeeld de lettercombinatie MH als $M H$ wordt opgevat, dus als een vermenigvuldiging. In het kader de situatie als u in de *Question Designer* deze instellingen hanteert bij een Maple-graded invulveld met instelling *Formula*.

Als de student na afloop van de toets op *Grade* klikt, dan wordt deze ingetikte formule toch goed gerekend en hij kan bij *Your Answer* kijken wat hij heeft ingetikt zoals te zien is in de volgende figuur.

Grade: 100%

Maak uit de volgende vergelijking m vrij.

$$P = \frac{M^2 H}{v(3M + 5m)h}$$



Correct

Het antwoord mag in een zelfgekozen vorm ingetikt worden.

Let wel op hoofdletters en kleine letters!

Vul de stippels in

$m = \dots$

Your Answer: $M*(-3*P*v*h+MH)/(5*P*v*h)$

Comment: Het goede antwoord is $\frac{1}{5} \frac{M(-3Pvh + MH)}{Pvh}$

Figure 1.17: Grading bij Formula met Text Mode

TIP: In deze vraag is in het ontwerp van de vraag het correcte antwoord niet ingevuld. Het correcte antwoord wordt hier echter wel in de Feedback (Comment) gegeven. Zie ook figuur *Figure 1.70 (page 49)* waar en hoe het goede antwoord in het ontwerp van de vraag ingevuld moet worden.

BELANGRIJKE TIP: De instelling met *Formula* voor het toetsen van formules in een Maple-graded vraag is eigenlijk **absoluut af te raden!** U kunt wel formules met lettercombinaties toetsen, maar de student wordt totaal op het verkeerde been gezet met deze zogenaamd "gebruiksvriendelijke" instelling. De student leert ook het verschil niet te zien tussen f^*x , fx , $f(x)$ en $f x$. Het wordt allemaal op dezelfde manier door het systeem opgevat namelijk als een vermenigvuldiging van f maal x .

Ook de *Preview*-knop geeft absoluut geen betrouwbare weergave van de invoer. Neem bijvoorbeeld de exponentiële functie. Als deze ingetikt wordt als $\exp(x)$, dan wordt deze in de *Preview* vertaald naar $e x p x$ terwijl bij het intikken van $\cos x$ met de *Preview*-knop de formule als volgt laat zien: $\cos(x)$. En $\sin 2x$ wordt opgevat als $\sin(2) x$. Verder wordt een ingetikte veelterm voorzien van allerlei haakjes in de *Preview*. Zie de figuur hier onder.

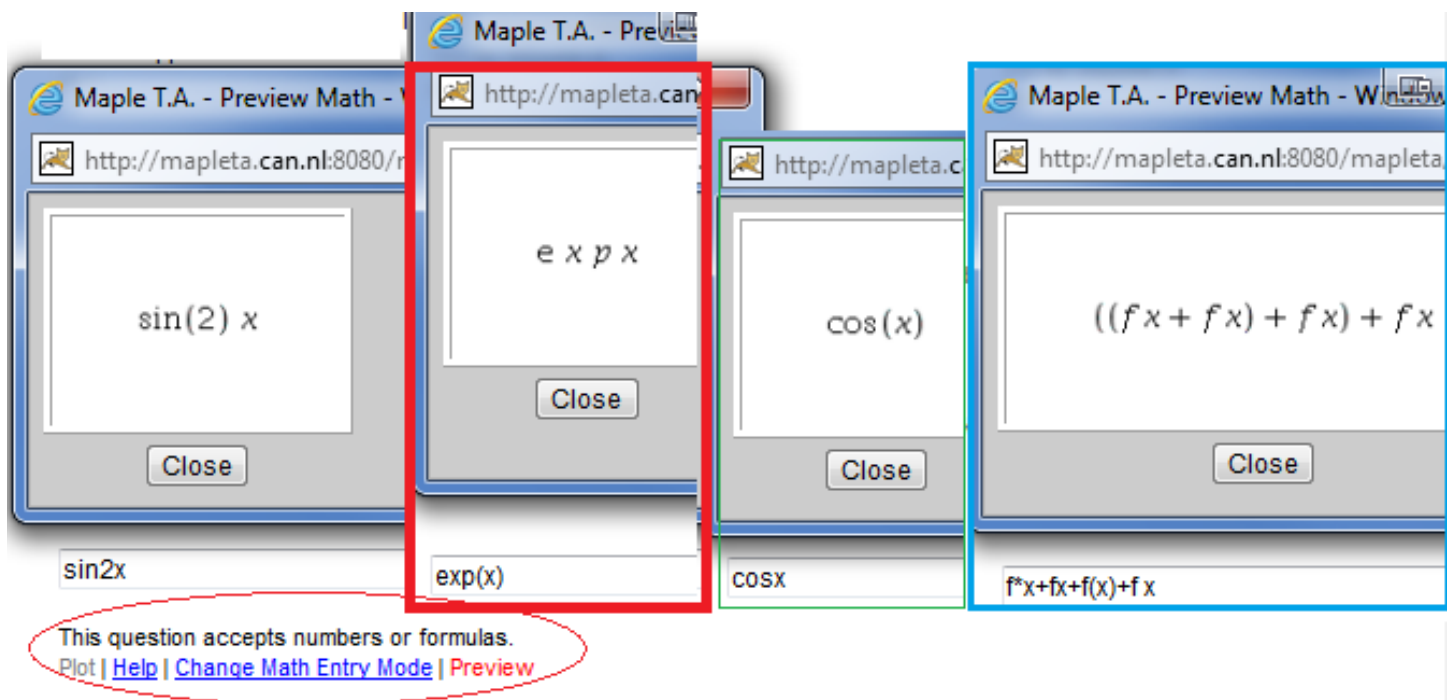


Figure 1.18: De Preview bij de instelling Formula in de Text Mode

TIP: Met de instelling *Maple syntax Text Entry only* (paragraaf *Text Mode* (page 3)) krijgt u een veel betere *Preview* te zien zoals in de figuur hier onder duidelijk gemaakt is. Echter bij functies MOETEN haakjes getikt worden en als de student spaties tikt tussen letters, dan komt er een *Invalid Maple syntax*-boodschap te staan. Ook moet u er zich bewust van zijn dat Maple vereenvoudigt alvorens de *Preview* weer te geven.

In sommige gevallen is deze vereenvoudiging te voorkomen. Zie daarvoor in een van de tips hier onder.

Overigens worden spaties aan weerszijden van operatoren genegeerd bij deze instelling.

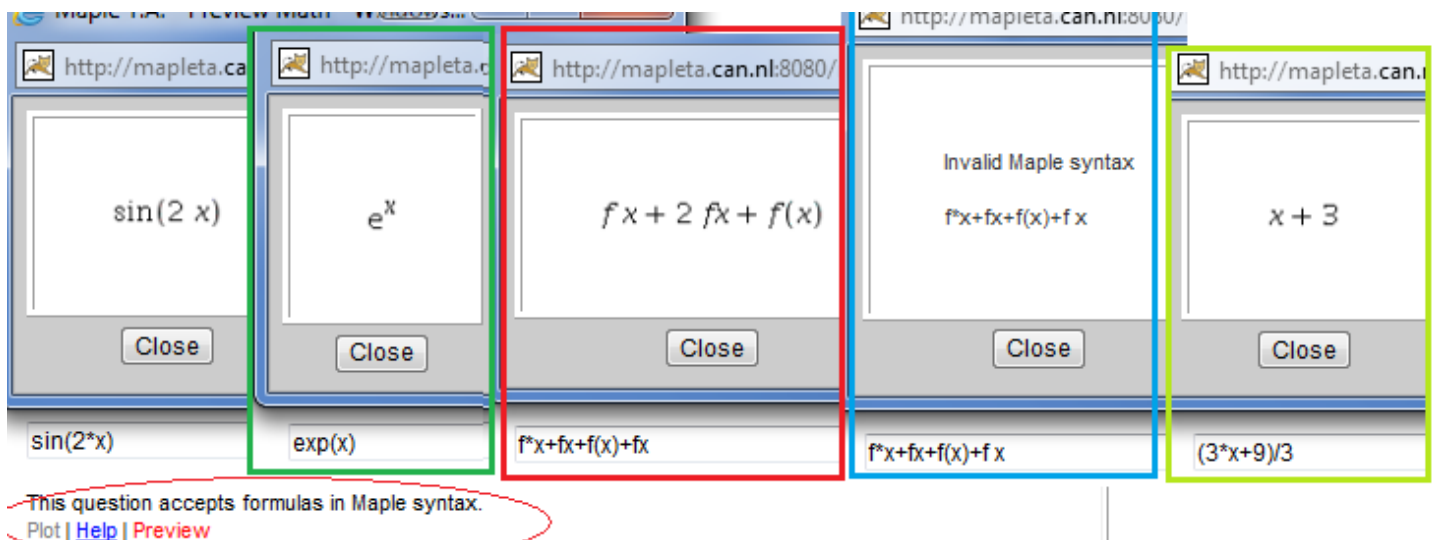


Figure 1.19: De Preview bij de instelling Maple syntax in de Text Mode

TIP: Het voordeel van de instelling *Formula* zou kunnen zijn dat de *Preview*-knop géén automatische vereenvoudiging genereert, zoals wel bij de instelling *Maple syntax* in de *Text Mode* de *Preview*-knop van $(3^*x+9)/3$ bijvoorbeeld $x + 3$ maakt.

U kunt het in het achterhoofd houden voor eventualiteiten waarbij er geen andere mogelijkheden meer zijn.

TIP: Overigens maakt de *Preview* van deze *Formula*-instelling een hopeloos slechte *Preview* in vergelijking met de instelling met *Maple syntax* zoals te zien is in bovenstaande figuren.

TIP: Omdat nu ook mogelijkheden zijn met *Custom Previewing*-instellingen, kunt u in die rubriek ervoor zorgen dat de vereenvoudiging in de *Preview* wordt tegengehouden (in de meeste gevallen). Programmeer dat met:

```
use InertForm:-NoSimpl in $RESPONSE: end: printf(InertForm:-ToMathML(%));
```

1.2.2.2.2 Symbol Mode bij de instelling Formula

Door te klikken op *Change Math Entry Mode* (in de *Question Designer* klikken op het *Sigma*-teken achter het invulveld) krijgt de student een pop-up-schermpje waar de student middels een radiobutton kan kiezen voor *Symbol Mode* of voor *Text Mode*, zie *Figure 1.16* (page 14) waar gekozen is voor *Text Mode*.

Als de student zelf gekozen heeft voor *Symbol Mode*, krijgt hij een Editor aangeboden waarin de formule opgebouwd kan worden. Er worden paletten met knoppen aangeboden waarmee de formule gebouwd kan worden. Deze Editor is aanmerkelijk minder uitgebreid dan de Editor die aangeboden wordt bij *Maple syntax Symbol entry only*.

Ook hier geldt weer dat lettercombinaties door het systeem gezien worden als individuele letters. De student mag spaties tikken tussen de letters, of een ster of helemaal niets. Steeds wordt dat automatisch door het systeem vertaald in vermenigvuldigingen.

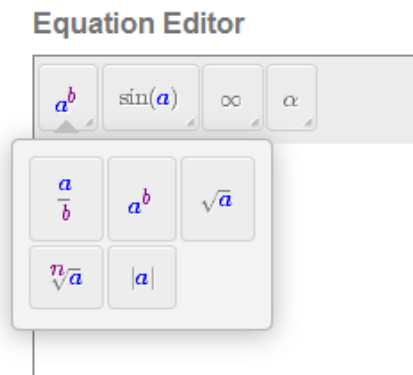


Figure 1.20: Instelling voor Formula met de formule Editor (Symbol Mode)

De student kan op *Help* klikken voor informatie over hoe de formules ingevoerd dienen te worden. Eventueel kan de student zelf kiezen voor de *Text Mode* met *Change Math Entry Mode*. Let op de knop met de dubbele pijl rechts in beeld om het scherm van de editor even schoon te maken als de student opnieuw wil beginnen met het invoeren van de formule.

Als de student de grading opvraagt, krijgt hij bij *Your Answer* te zien (*Figure 1.21* (page 18)) wat hij tweedimensionaal heeft ingevoerd (zonder spaties).

In de *Feedback* (*Comment*) is het juiste antwoord gecommuniceerd waarin wel spaties staan tussen de letters om de vermenigvuldiging aan te geven. Toch wordt het antwoord goedgekeurd. Het houdt in dat ook in deze situatie het systeem het vermenigvuldigingsteken zelf aanvult als er lettercombinaties worden ingevoerd.

Grade: 100%

Maak uit de volgende vergelijking m vrij.

$$P = \frac{M^2 H}{v(2M + 6m)h}$$



Correct

Het antwoord mag in een zelfgekozen vorm ingetikt worden.

Let wel op hoofdletters en kleine letters!

Vul de stippels in

 $m = \dots$

Your Answer:
$$\frac{M(-2Pvh + MH)}{(6Pvh)}$$

Comment: Het goede antwoord is
$$\frac{1}{6} \frac{M(-2Pvh + MH)}{Pvh}$$

Figure 1.21: Grading met gebruik van de Editor (Symbol Mode) bij Formula-instelling

TIP: U moet zich goed bewust zijn van de instellingen met *Formula* bij het vraagtype *Maple-graded*. Het is afhankelijk van wat u testen wilt, maar in het algemeen is deze instelling zeer verwerpelijk!

TIP: Het vraagtype *Formula* heeft ook de mogelijkheid voor de student om tussen *Text Mode* en *Symbol Mode* te kiezen. De Editor van deze *Symbol Mode* is nog een stuk primitiever dan dezelfde instelling bij het vraagtype *Maple-graded*. Zie ook in paragraaf *Vraagtype Formula* (page 129).

1.2.3 Formules voorbereiden

Voor formules op het scherm werken we met HTML-code of MathML-code of vanaf versie 9 zelfs met LaTeX. Deze code zorgt ervoor dat de formules in de Browser op de juiste manier worden weergegeven.

Er is een aantal manieren om de MathML-code of LaTeX-code te genereren en in bepaalde gevallen hoeven we zelfs geen MathML-code te gebruiken, maar gewoon alleen HTML-code of LaTeX-code. Vooral in situaties van kleine formules in de regel is het in feite beter om HTML-code te gebruiken. Zie voor meer informatie en voorbeelden in paragraaf *Formules met HTML* (page 21) en voor LaTeX in paragraaf *LaTeX* (page 38).

U kunt de formules die u gaat gebruiken in de tekst van de vraag, in de hints of in de feedback het beste voorbereiden in de rubriek *Algorithm* waar alle andere variabelen ook voorbereid worden. U hoeft dan alleen maar waar nodig de formule aan te roepen met zijn naam om deze netjes op het scherm te laten verschijnen.

Formules die u op het scherm wilt presenteren moeten voldoen aan de internationale conventies. Dat wil zeggen dat bijvoorbeeld variabelen in de formule cursief zijn en dat er een juiste spatiëring gehanteerd wordt. Zie ook <http://www.methakamminga.nl/MapleTA/formules.pdf>.

In de volgende paragrafen wordt uiteengezet hoe MathML-code gegenereerd kan worden.

TIP: Ook de antwoorden kunnen in de rubriek *Algorithm* voorbereid worden, of dat nu tekst, getallen of formules zijn. Deze antwoorden zijn vanzelfsprekend niet gecodeerd met behulp van MathML. Hoe u daarmee kunt omgaan wordt duidelijk in de voorbeelden bij de bespreking van de verschillende vraagtypen.

1.2.3.1 Formules en getallen voorbereiden in het Algorithm

De formules waarmee gerekend wordt, kunnen alvast voorbereid worden in het *Algorithm*. Maar let op: er zijn verschillende gedaanten van de formules mogelijk. Bekijk de volgende figuur.

```

$a:=range(-10,-2);
$b:=range(-10,10);
$formule1="$a+5*Mk-$b/$a*Q";
$formule2=$a+5*Mk-$b/$a*Q;
$formule3:=maple("$a+5*Mk-($b)/($a)*Q");

```

| Variable | Value |
|----------|--------------------------------------|
| a | -8 |
| b | 6 |
| formule1 | $-8+5Mk-6/-8Q$ |
| formule2 | $-8.0+(((5.0(M))^{(k)})-(-0.75(Q)))$ |
| formule3 | $-8+5Mk+3/4Q$ |

Figure 1.22: Formules in Algorithm

In bovenstaande figuur merken we het volgende op:

De variabele \$formule1 is gemaakt mét quotes zodat Mk als één geheel wordt beschouwd en de deling $6/8$ wordt niet vereenvoudigd. De variabele \$formule2 is gemaakt zonder quotes. Het resultaat is niet echt om aan te zien en bovendien wordt Mk beschouwd als een vermenigvuldiging $M*k$. Ook wordt de deling $6/8$ omgezet in een decimaal getal en ook de andere getalsvariabelen worden als decimale getallen opgevat. Zonder quotes kunnen er ook geen vergelijkingen gedefinieerd worden op deze manier (dus formules met een =-teken).

De variabele \$formule3 is de meest elegante formule om mee verder te rekenen.

Met Maple kunnen er zonder probleem veel verschillende formules gedefinieerd worden en Maple rekt exact. Echter u bent dan wel afhankelijk van de automatische vereenvoudigingen die dan plaatsvinden.

De code voor het uitbesteden van een opdracht aan Maple is steeds

```
maple(".....")
```

De variabele Mk wordt gewoon als één variabele opgevat en de deling $6/8$ wordt automatisch vereenvoudigd, maar niet omgezet in decimalen (tenzij de variabelen \$a en \$b al gedefinieerd waren als decimale getallen). Let wel op de haakjes om de variabelen \$a en \$b als deze eventueel ook negatief kunnen zijn en midden in de formule staan.

TIP: Als Maple er aan te pas komt, let dan op dat variabelen die mogelijk negatief kunnen zijn, tussen haakjes komen te staan. Maple kan namelijk niet omgaan met twee operatoren achter elkaar. Als de randomvariabelen ook negatieve waarden kunnen aannemen, is het verstandig deze in formules waar Maple aan het werk moet, tussen haakjes op te nemen, zoals in het volgende voorbeeld te zien is. Immers Maple verstaat geen dubbele operatoren.

```
$a=switch(rint(2),range(-10,-1),range(1,10));
$b=switch(rint(2),range(-10,-1),range(1,10));
$vraag=maple("($a)/x+($b)");
$vraag1=maple("($a)/x+$b");
```

| Variable | Value |
|----------|---|
| a | -3 |
| b | -8 |
| vraag | -3/x-8 |
| vraag1 | com.maplesoft.server.router.MapleSyntaxException: on line 368, syntax error, `-' unexpected: |

Figure 1.23: Negatieve waarden van de variabelen

Als u deze error-melding krijgt met `` unexpected, dan weet u waar dat dus aan kan liggen.

Het is ook handig om de truc met quotes te kennen als het gaat om grote getallen. Zie het verschil tussen de volgende manieren om een getal te representeren.

```
$a=range(-10,-2);
$b=range(-10,10);
$c=range(3000,4000);
$getal1=$a*$c/$b;
$Getal1="$getal1";
$getal2=maple("$a*$c/($b)");
$getal3=maple("evalf[5]($a*$c/($b))");
$getal4=decimal(1,$getal1);
$getal5="$getal4";
```

| Variable | Value |
|----------|--------------|
| a | -10 |
| b | -7 |
| c | 3,334 |
| getal1 | 4,762.857143 |
| Getal1 | 4762.85714 |
| getal2 | 33340/7 |
| getal3 | 4762.9 |
| getal4 | 4,762.9 |
| getal5 | 4762.9 |

Figure 1.24: Gedaante van getallen

In bovenstaande figuur ziet u hoe u de duizendtal-separator kwijt kunt raken (met Maple het getal genereren of met quotes er omheen). Met het commando `evalf` in Maple kan het aantal significante cijfers gestuurd worden (niet het aantal decimalen).

1.2.3.2 Formules met HTML

Soms is het handig om HTML-code te gebruiken om formules netjes op het scherm te krijgen. Meestal gaat het dan om een kleine formule in de tekst. Zie ook in paragraaf *Latex* (page 38) om kleine formules meteen in te tikken tussen `\(.....\)`.

De meeste dingen zijn met de knoppen te doen die tot uw beschikking staan bij het maken van de tekst van de vraag of de hints of de feedback.

Er zijn knoppen voor cursief, subscript, font, kleur en dergelijke en er is ook nog een knopje *Insert Special Character* waarmee verschillende tekens in de tekst opgenomen kunnen worden (helaas nog geen Griekse letters, maar daar is MapleSoft mee bezig en vanaf versie 11 verwezenlijkt. Bovendien kunt u voor Griekse letters de Equation Editor gebruiken via de Sigma-knop).

TIP: In de paragraaf *Getallen, tekens, links en lettertypen in de presentatie van de vraag in de Handleiding Toets Items Maken Deel A* op de website, staan veel tips over het gebruik van HTML-code.

TIP: Gebruik voor een kleine formule of bijvoorbeeld een enkele Griekse letter in de regel zo weinig mogelijk de *Equation Editor*. Het is inderdaad ook mogelijk om de letter λ in de tekst niet met de *Equation Editor* te maken vooral met de komst van versie 9.5 is de gedaante van de formules die met de Equation Editor gemaakt zijn veel beter. Als u links boven op de knop *Source* klikt, ziet u de HTML-code staan van de tekens die met html gemaakt zijn en daarin kunt u ook weer editen.

Een voorbeeld daarvan is te zien in de volgende figuur waar zelfs ook variabelen uit de rubriek *Algorithm* gebruikt kunnen worden in de HTML-code.

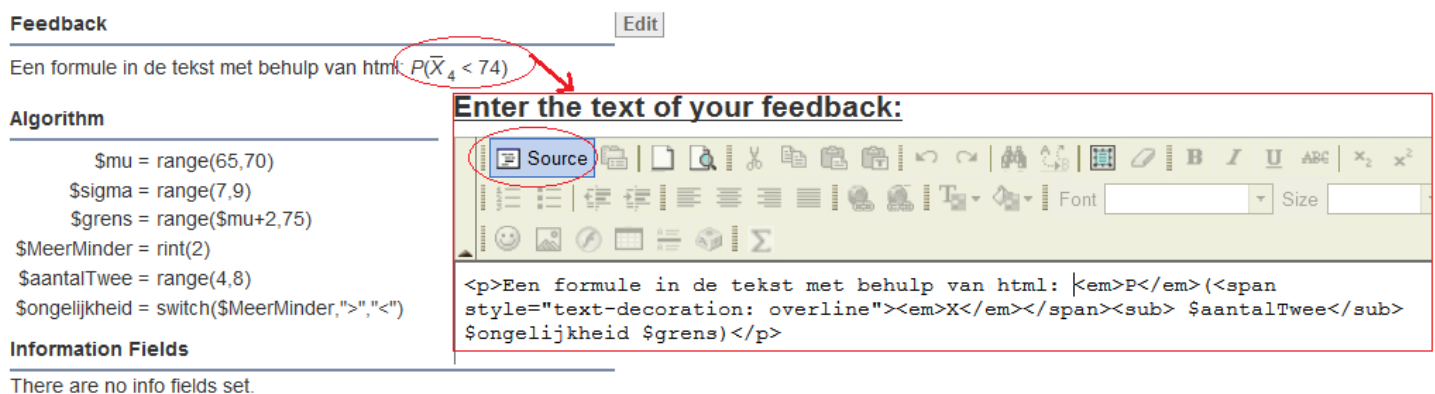


Figure 1.25: HTML in de broncode van de tekst van de vraag

`P(X_{$aantalTwee} $ongelijkheid $grens)`
De formule komt netjes in de regel te staan van de tekst van de feedback.

TIP: Helemaal mooi is als u LaTeX gebruikt. U kunt dan gewoon direct in de tekst van de vraag de LaTeX-code opnemen. Zie paragraaf (page 38)

TIP: Gebruik zo weinig mogelijk de *Equation Editor* en gebruik deze alleen als het op een andere manier niet kan.

Zie paragraaf *MathML met de Equation Editor* (page 33) voor meer informatie over de *Equation Editor* die niet op HTML-code is gebaseerd maar op MathML-code.

Zie voor mogelijkheden voor LaTeX-code in paragraaf LaTeX (page 38).

TIP: In alternatieven voor *Multiple Choice*-vragen kan ook heel goed HTML-code gebruikt worden, maar ook formules zijn te verwezenlijken met de *Sigma*-knop waarmee u in de Equation Editor komt.

TIP: Handig is ook om een formule met HTML-code eventueel voor te bereiden in het tekstvak van de vraag en die HTML-code eventueel (vanuit de broncode) te kopiëren naar zo'n invulveld voor Multiple Choice, waar geen knopjes zijn voor HTML.

TIP: HTML-code is tot op zekere hoogte ook voor te bereiden in het *Algorithm*. Neem dan de hele HTML-code over en zet dat tussen dubbele quotes zoals te zien is in onderstaande figuur.

```
$tekst="<em>x</em><sup>2</sup> + 7 met <strong>vette tekst</strong>";
```

| Variable | Value |
|----------|---------------------------|
| tekst | $x^2 + 7$ met vette tekst |

Figure 1.26: html-code in het Algorithm voorbereiden

1.2.3.3 Wat is MathML-code

Een formule netjes op het scherm krijgen is vaak een lastige zaak, maar met behulp van MathML-code is het vrij eenvoudig. Maple T.A. ondersteunt het gebruik van MathML-code op een elegante manier.

Zelf de code maken is niet nodig.

Een voorbeeld van een gecodeerde formule is de volgende code die door de browser gezien wordt als een echte formule.

```
<math xmlns='http://www.w3.org/1998/Math/MathML'><mrow><mi>P</mi><mo>=
</mo><mfrac><mrow><msup><mi>M</mi><mn>2</mn></msup><mo>&InvisibleTimes;
</mo><mi>H</mi></mrow><mrow><mrow><mi>v</mi><mo>&InvisibleTimes;
</mo><mfrac><mrow><mrow><mi>B</mi><mo>&InvisibleTimes;</mo><mi>M</mi></mrow><mo>+
</mo><mi>C</mi><mo>&InvisibleTimes;
</mo><mi>m</mi></mrow></mrow></mfrac></mrow><mo>&InvisibleTimes;
</mo><mi>h</mi></mrow></mfrac></mrow></math>
```

Figure 1.27: MathML-code

Deze code stelt de volgende formule voor en het systeem vertaalt dus de code naar een presentabel formaat:

$$P = \frac{M^2 H}{v (B M + C m) h}$$

Nogmaals: gelukkig hoeven we een dergelijke code niet te lezen en zeker niet zelf te maken.

Er zijn verschillende manieren om de MathML-code te maken die allemaal hieronder uitgelegd worden.

1.2.3.4 Vier manieren om MathML-code te maken

De MathML-code kan in Maple T.A. op drie manieren gemaakt worden.

- In de rubriek *Algorithm* met behulp van een Maple-commando (meest efficiënte manier).
`$displayvraag=maple("printf(MathML[ExportPresentation]($vraag))");`
 Zie paragraaf *MathML-code programmeren in rubriek Algorithm* (page 22).
- In de rubriek *Algorithm* met behulp van de `mathml`-functie van Maple T.A..
`mathml("$a/x+$b");`
 Zie verder in paragraaf *MathML-code programmeren in rubriek Algorithm* (page 22).
- In de rubriek *Algorithm* met behulp van LaTeX. U kunt deze LaTeX-code tussen quotes in het Algorithm voorbereiden en testen, maar u kunt ook deze code direct in de tekst van de vraag intikken. Zie paragraaf *LaTeX* (page 38) voor meer informatie.
- Met de *Equation Editor* is het vaak wat langdradig werk om een formule te bouwen en deze Editor is niet heel erg uitgebreid, maar het kan soms nodig zijn als het in de rubriek *Algorithme* niet lukt. Zie verder in paragraaf over de Equation Editor (page 33). In deze *Equation Editor* kan eventueel ook MathML-code vanuit een ander programma geplakt worden als u gewend bent in een ander formule-programma te werken dat meer faciliteiten biedt (bijvoorbeeld MathType). Zie daarvoor de paragraaf over MathType (page 36).

1.2.3.5 MathML-code programmeren in rubriek Algorithm

MathML-code kan heel gemakkelijk in de rubriek Algorithm voorbereid worden.

U definieert eerst de formule van de vraag waarmee gerekend zal worden.

Voor de tweedimensionale presentatie van de formule in de vraag, moet de formule van de vraag omgezet worden naar MathML-code. Dit kan alvast in de rubriek *Algorithm* voorbereid en zelfs gecontroleerd worden.

In het veld waar de vraag gesteld wordt (maar ook op elke andere plaats, bijvoorbeeld in de *Feedback* of in *Hints* of in alternatieven van *Multiple choice*-vragen), kan gerefereerd worden naar deze voorbereide code die als variabele is opgeslagen in de rubriek *Algorithm*.

```
$B=range(2,6);
$C=range(2,6);
condition:not(eq($B,$C));
$vraag=maple("P = M^2*H/(v*(( $B)*M+( $C)*m)*h)");
$displayvraag=maple("printf(MathML:-ExportPresentation($vraag))");
$opl=maple("solve($vraag,m)");
$displayopl=maple("printf(MathML:-ExportPresentation($opl))");
```

| Variable | Value |
|--------------|--|
| B | 5 |
| C | 3 |
| vraag | $P = M^2 H / v (5 M + 3 m) h$ |
| displayvraag | $P = \frac{M^2 H}{v (5 M + 3 m) h}$ |
| opl | $1/3 * M * (-5 * P * v * h + M * H) / P / v / h$ |
| displayopl | $\frac{1}{3} \frac{M (-5 P v h + M H)}{P v h}$ |

Figure 1.28: Formule en MathML voorbereiden in Algorithm

In bovenstaande figuur is te zien dat de variabele \$vraag een echte formule (vergelijking) is waarmee gerekend kan worden. De variabele \$displayvraag is in feite de MathML-vertaling naar een code waarmee de browser de formule in 2D kan presenteren, zoals ook te zien is aan het resultaat. In dit geval is de MathML-code steeds gegenereerd door Maple met de opdracht:

```
maple("printf(MathML:-ExportPresentation($vraag))"); of
maple("printf(MathML[ExportPresentation]($vraag))");
```

Bij het genereren van de oplossing \$opl, laten we Maple het werk doen door de vergelijking \$vraag met `solve` op te lossen naar de onbekende m . Het resultaat kunt u weer met behulp van Maple naar MathML coderen tot de variabele \$displayopl en deze kan dan weer gebruikt worden voor bijvoorbeeld de presentatie van de oplossing in de feedback. Steeds is het resultaat binnen het *Algorithm* goed te checken door op de knop *Refresh* te klikken.

Een tweede mogelijkheid is het maken van de MathML-code met behulp van het commando `mathml("...")`;

```
$a=range(2,5);
$b=range(3,5);
$vraag=maple("$a/x+$b");
$displayvraag=mathml("$vraag");
```


Probeer ook eens `$displayvraag = mathml("$vraag","nosimplify");` om ongewenste vereenvoudigingen tegen te houden. Soms is het handig om `mathml("$vraag,1")` te hanteren als u geen ongewenste vereenvoudiging wilt hebben.

LET OP: Met de omzetting van een formule naar MathML-code krijgt u een formule met de bedoeling deze op het scherm te presenteren maar NIET om mee te rekenen! Met de variabele `$vraag` kan binnen het systeem gerekend worden, met de variabele `$displayvraag` (gecodeerde formule) kan dus NIET gerekend worden. Het is dan handig om deze variabelen een naam te geven met display erin voor de herkenbaarheid.

U dient altijd uit te proberen of de vraag ook weergegeven wordt zoals bedoeld is. Dat kan ook gemakkelijk gecontroleerd worden binnen de rubriek *Algorithm* (met de knop *Refresh*). Bekijk eens de volgende mogelijkheden om de formule te coderen naar MathML:

```
$a=range(3,10);
$b=range(4,10);
$test1=maple("printf(MathML[ExportPresentation]($a+$b/$a*A))");
$test2=mathml("$a+$b/$a*A");
$test3=mathml($a+$b/$a*A);
$test4=mathml("$a+$b/$a*A","nosimplify");
```

| Variable | Value |
|----------|---------------------|
| a | 8 |
| b | 5 |
| test1 | $8 + \frac{5}{8} A$ |
| test2 | $8 + \frac{5}{8} A$ |
| test3 | $8 + 0.625 A$ |
| test4 | $8 + \frac{5}{8} A$ |

Figure 1.29: Verschillende manieren om MathML-code te maken

In bovenstaande figuur is te zien dat de MathML-code niet altijd door Maple gemaakt hoeft te worden met het aanroepen van `maple("printf(MathML[ExportPresentation](...))")`. Het programma zelf kan de code ook maken met `mathml("...")` en dat is in de meeste gevallen wel toereikend. Met de extra optie "nosimplify" bij `mathml("...")` wordt de automatische vereenvoudiging tegengehouden en als u bij de opdracht `mathml` de quotes weglaat, komen er zelfs decimale getallen, ook al zijn de oorspronkelijke variabelen `$a` en `$b` geen decimale getallen.

TIP: Hier staan trouwens geen haakjes om de variabelen bij `$a+$b/$a*A` omdat dat in dit geval niet beslist noodzakelijk is. Immers de variabelen `$a` en `$b` zullen steeds positief zijn in bovenstaand voorbeeld, maar als de variabelen ook negatieve waarden kunnen hebben, is het verstandiger om haakjes om de variabelen te plaatsen zoals dat ook al gedaan is in *Figure 1.28 (page 23)*. Zie ook paragraaf *Formules in Algorithm (page 18)*.

TIP: De functie `mathml("...")` van het systeem is minder krachtig dan die van Maple:

```
maple("printf(MathML[ExportPresentation](...))")
```

Om te beginnen kan `mathml("...")` geen karaktercombinaties aan en geen ongelijkheden aan en als er geen quotes gebruikt worden zelfs ook geen vergelijkingen. De opdracht met Maple `maple("printf(MathML[ExportPresentation](...))")` kan veel meer, ook met het oog op lettercombinaties en subscript, vergelijkingen, ongelijkheden integralen en

differentialen. Let dus op bij lettercombinaties: dat kan `mathml (" . . . ")` namelijk beslist niet aan, want dan worden de lettercombinaties als vermenigvuldiging gezien met spaties ertussen. Zie in de volgende figuur:

```
$a:=range(2,5);
$test1:=mathml("$a*Pb+Q");
$test2:=mathml("b P$a+Q");
$test3:=maple("printf(MathML[ExportPresentation]($a*Pb+Q))");
$test4:=maple("printf(MathML[ExportPresentation]($a*P[b]+Q))");
$test5:=maple("printf(MathML[ExportPresentation](b*P[$a]+Q))");
```

| Variable | Value |
|----------|-------------|
| a | 2 |
| test1 | $2 P b + Q$ |
| test2 | $2 b P + Q$ |
| test3 | $2 P b + Q$ |
| test4 | $2 P_b + Q$ |
| test5 | $b P_2 + Q$ |

Figure 1.30: MathML met subscript en lettercombinaties

In de volgende figuur ziet u nog wat voorbeelden van andere formules:

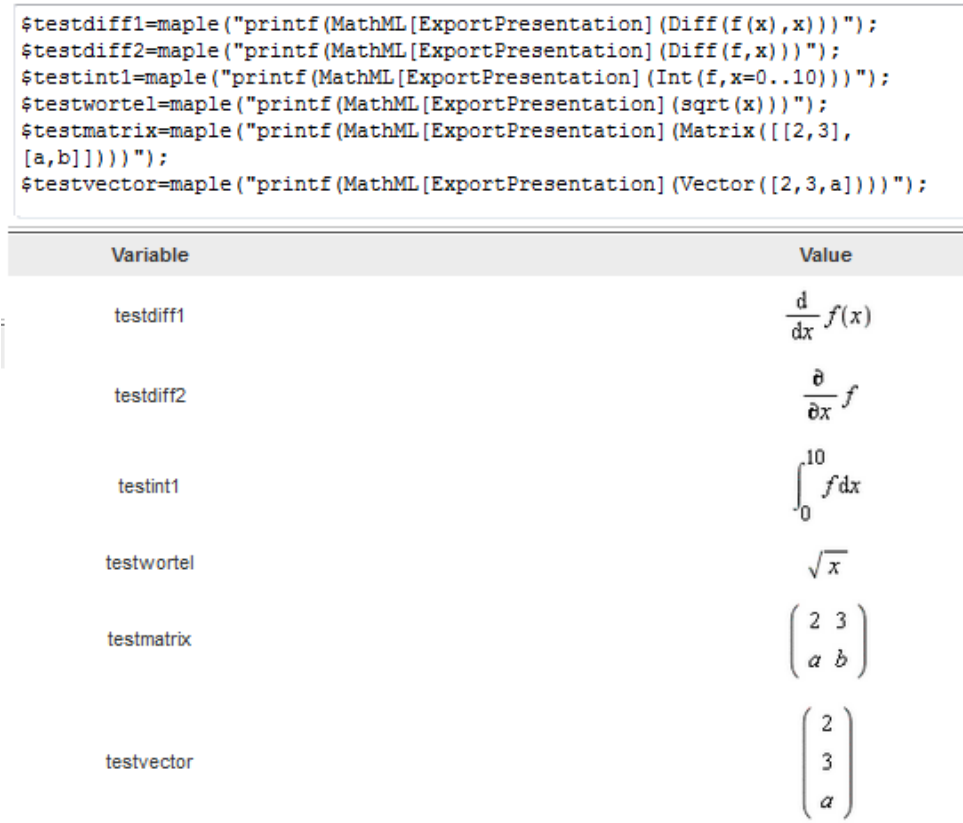


Figure 1.31: MathML-Formules met Maple gemaakt

TIP: Let u op dat het soms verschil uitmaakt als u wel of geen spaties tikt bij de quotes aan weerszijden van het commando met printf: `maple(" printf(MathML[ExportPresentation](...)) ")`

TIP: Soms is een *combinatie* van de functie `mathml (" . . . ")` en `maple ("printf(MathML[ExportPresentation] (. . .))")` een oplossing:

```
$a:=range(2,10);
$b:=range(2,10);
$c:=range(2,10);
$d:=range(2,10);
$test:=maple("printf(MathML[ExportPresentation]((2*x-$c)/($a+$c-$d)))");
$teller:=maple("numer((2*x-$c)/($a+$c-$d))");
$noemer:=maple("denom((2*x-$c)/($a+$c-$d))");
condition:=not(eq($noemer,1));
$breuk:=mathml("($teller)/($noemer)");
```

| Variable | Value |
|----------|------------------------------|
| a | 3 |
| b | 5 |
| c | 2 |
| d | 2 |
| test | $\frac{2}{3}x - \frac{2}{3}$ |
| teller | $2x-2$ |
| noemer | 3 |
| breuk | $\frac{2x-2}{3}$ |

Figure 1.32: MathML coderen met combinaties van methoden

In bovenstaande figuur is het lastig om een bepaalde breukvorm af te dwingen in de MathML-gecodeerde vorm. Maple doet vaak automatische vereenvoudigingen, vooral met getallen. Deze vereenvoudigingen zijn bijna niet tegen te houden!

Er valt soms wel eens wat te proberen met de combinatie van beide methoden voor het MathML-coderen. Steeds is binnen de rubriek *Algorithm* direct het effect te beoordelen.

TIP: Tegenwoordig is er nog een extra mogelijkheid gekomen om de automatische vereenvoudiging tegen te houden. Zie daarvoor paragraaf (page 29).

TIP: Nog een mogelijkheid als de combinatie van twee (of meer) formules in de tekst van de vraag bij de voorbereiding in het *Algorithm* leidt tot ongewenste vormen of vereenvoudigingen, is het handig om eerst twee formules apart te definiëren en in de

uiteindelijke tekst van de vraag door middel van een \times ([AltGr][=]) of met een midden-stip (met de knop voor Special Characters) met elkaar te vermenigvuldigen:

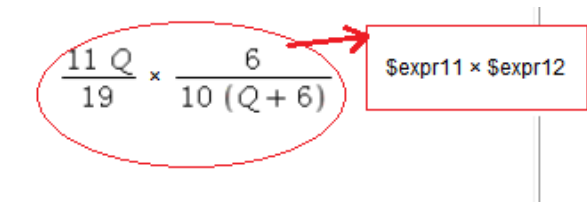


Figure 1.33: MathML voorbereiden in gedeelten

In bovenstaande formule is in de tekst van de vraag een combinatie van twee formules te zien met een \times -teken ertussen. Dit teken kan niet gemaakt worden met de bekende methoden om MathML te coderen in de rubriek *Algorithm*. Op deze manier: met het prepareren van twee formules, zou dat opgelost kunnen worden.

TIP: Als u op bovenstaande manier twee formules wilt presenteren met een minteken ertussen, neem dan het minteken uit de set van de Special Characters. Dit minteken is iets robuuster.

TIP: Heel veel oplossingen voor problemen met het presenteren van formules kunt u bereiken met LaTeX zie paragraaf (page 38).

TIP: Nog een mooie manier om een functie te definiëren en daar in het algorithm gebruik van te maken om snel functiewaarden te berekenen is de volgende:

```
$a=range(1,4);
$b=switch(rint(2),range(-4,-1),range(1,4));
$functie=switch(rint(5),"$a*x^2+($b)","$b*x^2+$a","($a*x+($b))^2","$a*x^2+($b)","x^2");
$f=maple("proc (x) options operator, arrow: $functie end proc");
$fdisplay=maple("printf(MathML[ExportPresentation]($f))");
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

```
$a=range(1,4);
$b=switch(rint(2),range(-4,-1),range(1,4));
$functie=switch(rint(5),"$a*x^2+($b)","$b*x^2+$a","($a*x+($b))^2","$a*x^2+($b)","x^2");
$f=maple("proc (x) options operator, arrow: $functie end proc");
$fdisplay=maple("printf(MathML[ExportPresentation]($f))");
$f0=maple("$f(0)");
$f1=maple("$f(1)");
$f1=maple("$f(-1)");
$f2=maple("$f(2)");
$f2=maple("$f(-2)");
$f3=maple("$f(3)");
$f3=maple("$f(-3)");
```

| Variable | Value |
|----------|--|
| a | 4 |
| b | -2 |
| functie | 4*x^2+(-2) |
| f | proc (x) options operator, arrow; 4*x^2-2 end proc |
| fdisplay | $x \rightarrow 4x^2 - 2$ |

Figure 1.34: Een functie definiëren

1.2.3.5.1 Automatische vereenvoudiging tegenhouden

Er is tegenwoordig in Maple de mogelijkheid om met het pakket InertForm het een en ander te doen.

In veel gevallen heeft u last van de automatische vereenvoudiging van breuken of veeltermen als u de conversie naar MathML met Maple laat uitvoeren. Als u dat op de volgende manier doet binnen het pakket InertForm, wordt de vereenvoudiging in de meeste gevallen tegengehouden. Probeer wel een en ander altijd even uit.

```
$test7:=maple("use InertForm:-NoSimpl in x*y*(x+1)/(x*z*(x+1)): end: printf(InertForm:-ToMathML(%)) ");
```

```
$test7:=maple("use InertForm:-NoSimpl in x*y*(x+1)/(x*z*(x+1)): end: printf(InertForm:-ToMathML(%)) ");
$test8:=maple("use InertForm:-NoSimpl in x+y+2*x: end: printf(InertForm:-ToMathML(%)) ");
$a:=range(2,10);
$b:=range(2,10);
$c:=range(2,10);
$d:=range(2,10);
$test9:=maple("use InertForm:-NoSimpl in (2*x-$c)/($a+$c-$d): end: printf(InertForm:-ToMathML(%))");
```

| Variable | Value |
|----------|---------------------------|
| test7 | $\frac{xy(x+1)}{xz(x+1)}$ |
| test8 | $x + y + 2x$ |
| a | 7 |
| b | 9 |
| c | 7 |
| d | 10 |
| test9 | $\frac{2x-7}{7+7-10}$ |

Figure 1.35: Automatische vereenvoudiging tegenhouden

Ook handig om te weten is de nieuwe manier van MathML conversie binnen Maple als u let op de spaties bij de quotes. Een commando binnen de MathML conversie van Maple zal niet uitgevoerd worden als u de opdracht strak tussen de quotes plaatst. Echter als u een extra spatie ertussen voegt, wordt de opdracht volledig uitgevoerd binnen de quotes.

```
$test6:=maple("printf(MathML[ExportPresentation](a=log[2](x)))");
$test7:=maple(" printf(MathML[ExportPresentation](a=log[2](x))) ");
$test8:=maple(" printf(MathML[ExportPresentation](a=expand((x+2)*(x-2)))) ");
$test9:=maple("printf(MathML[ExportPresentation](a=expand((x+2)*(x-2))))");
```

```
$test6=maple("printf(MathML[ExportPresentation](a=log[2](x)))");
$test7=maple(" printf(MathML[ExportPresentation](a=log[2](x)) ");
$test8=maple(" printf(MathML[ExportPresentation](a=expand((x+2)*(x-2)))) ");
$test9=maple("printf(MathML[ExportPresentation](a=expand((x+2)*(x-2))))");
```

| Variable | Value |
|----------|-------------------------------------|
| test6 | $a = \log_2(x)$ |
| test7 | $a = \frac{\ln(x)}{\ln(2)}$ |
| test8 | $a = x^2 - 4$ |
| test9 | $a = \text{expand}((x + 2)(x - 2))$ |

Figure 1.36: Gebruik van quotes bij de MathML conversie van Maple

Overigens presenteert Maple (en tegenwoordig ook de moderne rekenmachines met een groter display) het grondtal van de logaritme als subscript: $\log_4(20)$.

TIP: Soms zijn backward quotes ook handig om Maple *exact* te laten overnemen wat u wilt. Deze backward quotes zijn ervoor om hetgeen tussen die quotes staat precies zo te laten als het is en er niets mee te doen.

Bijvoorbeeld:

```
$test1=maple("printf(MathML[ExportPresentation](c[n]=5*(``)^6/(5!*n)))");
```

Dit komt er uit te zien als $\frac{5^6}{5!n}$.

```
$test1=maple("printf(MathML[ExportPresentation](c[n]=5*(``)^6/(5!*n)))");
```

| Variable | Value |
|----------|-------------------------|
| test1 | $c_n = \frac{5^6}{5!n}$ |

Figure 1.37: MathML en quotes

Er moet dus voorkomen worden dat 5^6 wordt uitgerekend door Maple.

Met $5*``^6$ (of $5*(``)^6$) wordt het systeem even op het verkeerde been gezet en wordt er in feite een spatie tot de macht 6 verheven. Zo wordt ook voorkomen dat de 6 zowat tegen de 5 aan wordt afgebeeld én de 5 wordt een echte 5 die niet cursief is!

Dat 5^6 niet geëvalueerd wordt, had ook wel met de opdracht `mathml` bereikt kunnen worden, maar dan werd het uitroepteken weer niet verstaan en subscript was dan ook niet mogelijk in bovenstaande voorbeeld. Door gebruik te maken van de quotes stijf tegen de opdracht aan te zetten (`maple("...")`) wordt de evaluatie van deze berekening $5!$ tegen gehouden. Immers $5! = 120$.

Iets dergelijks bereikt u ook in de volgende formule waar `mathml` een resultaat geeft met veel overbodige haakjes en waar het gebruik van Maple de vereenvoudiging tegenhouden kan worden met `InertForm`.

Als uiterste middel kunt u ook nog gebruikmaken van LaTeX. (Zie paragraaf (page 38))

```
$displayq2:=maple("use InertForm:-NoSimpl in $b*x+($c)+($a)*y+($d)+($c)*x +($d)*y: end:
printf(InertForm:-ToMathML(%))");
```

```
$a=1;
$b=switch(rint(2),range(-10,-1),range(1,10));
$c=switch(rint(2),range(-10,-1),range(1,10));
$d=switch(rint(2),range(-10,-1),range(1,10));
$displayq1=mathml("$b*x+($c)+($a)*y+($d)+($c)*x +($d)*y");
$displayq2:=maple("use InertForm:-NoSimpl in $b*x+($c)+($a)*y+($d)+($c)*x +($d)*y: end:
printf(InertForm:-ToMathML(%))");
$test1:=maple("printf(MathML[ExportPresentation](3*x^`+($a)*x))");
```

| Variable | Value |
|-----------|--------------------------------------|
| a | 1 |
| b | 5 |
| c | 10 |
| d | -3 |
| displayq1 | $((((5x + 10) + y) - 3) + 10x) - 3y$ |
| displayq2 | $5x + 10 + 1y - 3 + 10x - 3y$ |
| test1 | $3x + x$ |

Figure 1.38: MathML en quotes

In bovenstaande figuur is te zien dat de opdracht `mathml` bij de variabele `$displayq1` weliswaar de vereenvoudiging niet doet, maar er komen wel allemaal storende en overbodige haakjes in de formule.

Bij de variabele `$displayq2` geven we Maple de opdracht om binnen `InertForm` de MathML-codering te maken. (Let op dat de variabelen tussen haakjes staan, want ze kunnen hier eventueel negatief zijn.) We zijn nu niet meer overgeleverd aan de automatische vereenvoudiging van Maple. Echter een nadeel is dan weer wel dat u in de formule bijvoorbeeld $1y$ ziet staan. Een mogelijkheid is om in de variabelen niet het getal 1 mee te nemen.

Eventueel met nog een andere truc kunnen we Maple laten geloven dat de termen niet gelijksoortig zijn door een van de termen met een "spatie" te vermenigvuldigen. Let dan wel op dat de spaties ergens komt te staan waar hij niet duidelijk storend is. Bijvoorbeeld bij de variabele `$test1` staat de spatie links van het plusteken, zodat dat minder opvalt.

TIP: Om te voorkomen dat bijvoorbeeld "Re" bij Maple opgevat wordt als "het reële deel", en u wilt Re ergens anders voor gebruiken, dan kunt u ook met quotes werken. Ingetikt als ``Re`` (dus inclusief spatie!!!) wordt het dan niet als zodanig opgevat. Echter u ziet wel deze lettercombinatie uiteindelijk cursief verschijnen (zie Figure 1.39 (page 32)). Wilt u dit niet cursief in de formule laten verschijnen, dan is er nog de mogelijkheid om de formule in LaTeX voor te bereiden (vanaf versie 9 zie paragraaf Formules met LaTeX (page 38)).

TIP: Automatisch wordt de hoofdletter I bij Maple altijd opgevat als de imaginaire eenheid en daarmee afgebeeld in roman (dus niet cursief). Echter als u de letter *I* als variabele voor iets anders gebruikt, kunt u Maple elke keer laten weten dat de letter een symbool is (zie Figure 1.39 (page 32)).

```
$antw1 = maple("local I:=`I `:3*P*L/(E*I)");
$antw1display=maple("printf(MathML[ExportPresentation]($antw1))");
$test1=maple("local Re:=`Re `:printf(MathML[ExportPresentation](Re+R[e]))");
```

| Variable | Value |
|--------------|---------------------------------|
| antw1 | $3 \cdot P \cdot L / E \cdot I$ |
| antw1display | $\frac{3 P L}{E I}$ |
| test1 | $R_e + R_e$ |

Figure 1.39: Imaginaire eenheid

U kunt dus de imaginaire eenheid I , van backquotes voorzien inclusief een spatie, en dan wordt deze als een variabele (symbool) opgevat in de MathML-code.

Dus bijvoorbeeld met `$test=maple("local I=`I `:printf(MathML[ExportPresentation](Y=C+I))");`

TIP: Maak dan ook gebruik van de *Custom Previewing Code* om te voorkomen dat als de student *Preview* gebruikt in een Maple-graded vraagtype, dat daar dan weer de I wordt opgevat als imaginaire eenheid.

U vult dan bij *Custom Previewing Code* van het Maple-graded vraagtype het volgende in:

```
local I=`I `:printf(MathML[ExportPresentation]($RESPONSE));
```

22 Test Parameters Imaginarie eenheid

Antwoord moet zijn $3 \cdot \frac{P \cdot L}{E \cdot I}$ maar MapleTA herkent de I als imaginaire unit Als daar niet i gebruikgemaakt wordt van de I Maple op het juiste spoor wordt gezet.

Ook met de **Custom Previewing Code:**

$3 \cdot P \cdot L / (E \cdot I)$

Preview

$$3 \frac{P L}{E I}$$

Close

Figure 1.40: Custom PlottingCode

Zie voor het rekenen en graden met complexe getallen in paragraaf *Uitdrukkingen met de letter I* (page 83).

Nog een voorbeeld waarbij ten behoeve van de feedback onvereenvoudigde formules voorbereid kunnen worden door met InertForm te werken.

```
$a=switch(rint(2),rint(-9,-2),rint(2,9));
$b=switch(rint(2),rint(-9,-2),rint(2,9));
$num1=maple("(x-($a))*(x-($b))");
$question1=maple("Limit(expand($num1)/(x-($a)),x=$a)");
$question1display=maple("printf(MathML[ExportPresentation]($question1))");
$question2=maple("Limit($num1/(x-($a)),x=$a)");
$question2display=maple("printf(MathML[ExportPresentation]($question2))");

$question3display=maple("use InertForm:-NoSimpl in Limit($num1/(x-($a)),x=$a): end:
printf(InertForm:-ToMathML(%))");
```

```
$a=switch(rint(2),rint(-9,-2),rint(2,9));
$b=switch(rint(2),rint(-9,-2),rint(2,9));
$num1=maple("(x-($a))*(x-($b))");
$question1=maple("Limit(expand($num1)/(x-($a)),x=$a)");
$question1display=maple("printf(MathML[ExportPresentation]($question1))");
$question2=maple("Limit($num1/(x-($a)),x=$a)");
$question2display=maple("printf(MathML[ExportPresentation]($question2))");
$question3display=maple("use InertForm:-NoSimpl in Limit($num1/(x-($a)),x=$a): end:
printf(InertForm:-ToMathML(%))");
```

| Variable | Value |
|------------------|--|
| a | -3 |
| b | -9 |
| num1 | $(x+3)*(x+9)$ |
| question1 | $\text{Limit}((x^2+12x+27)/(x+3), x = -3)$ |
| question1display | $\lim_{x \rightarrow -3} \frac{x^2+12x+27}{x+3}$ |
| question2 | $\text{Limit}(x+9, x = -3)$ |
| question2display | $\lim_{x \rightarrow -3} (x+9)$ |
| question3display | $\lim_{x \rightarrow -3} \frac{(x+3)(x+9)}{x+3}$ |

Figure 1.41: Formules met quotes

1.2.3.6 MathML met de Equation Editor

Soms is het vooraf programmeren van de MathML-code in de rubriek *Algorithm* niet voldoende om de formule op het scherm te krijgen zoals u misschien zou willen.

U wilt bijvoorbeeld de lettergrootte aan kunnen passen of misschien geeft Maple de formule niet weer zoals u zou willen zien.

Er zijn dan nog wel wat trucjes met quotes of combinaties met de opdracht `mathml("...")` binnen Maple T.A. zoals in paragraaf *Werken met quotes* (page 29) is besproken.

We hebben altijd nog de mogelijkheid van de *Equation Editor* achter de hand, *Figure 1.42 (page 34)*. Deze is op verschillende manieren te bereiken bijvoorbeeld met de knop met het Sigmateken in het formulier waar de tekst van de vraag ingevuld moet worden (of in de rubriek *Hints* of *Feedback*).

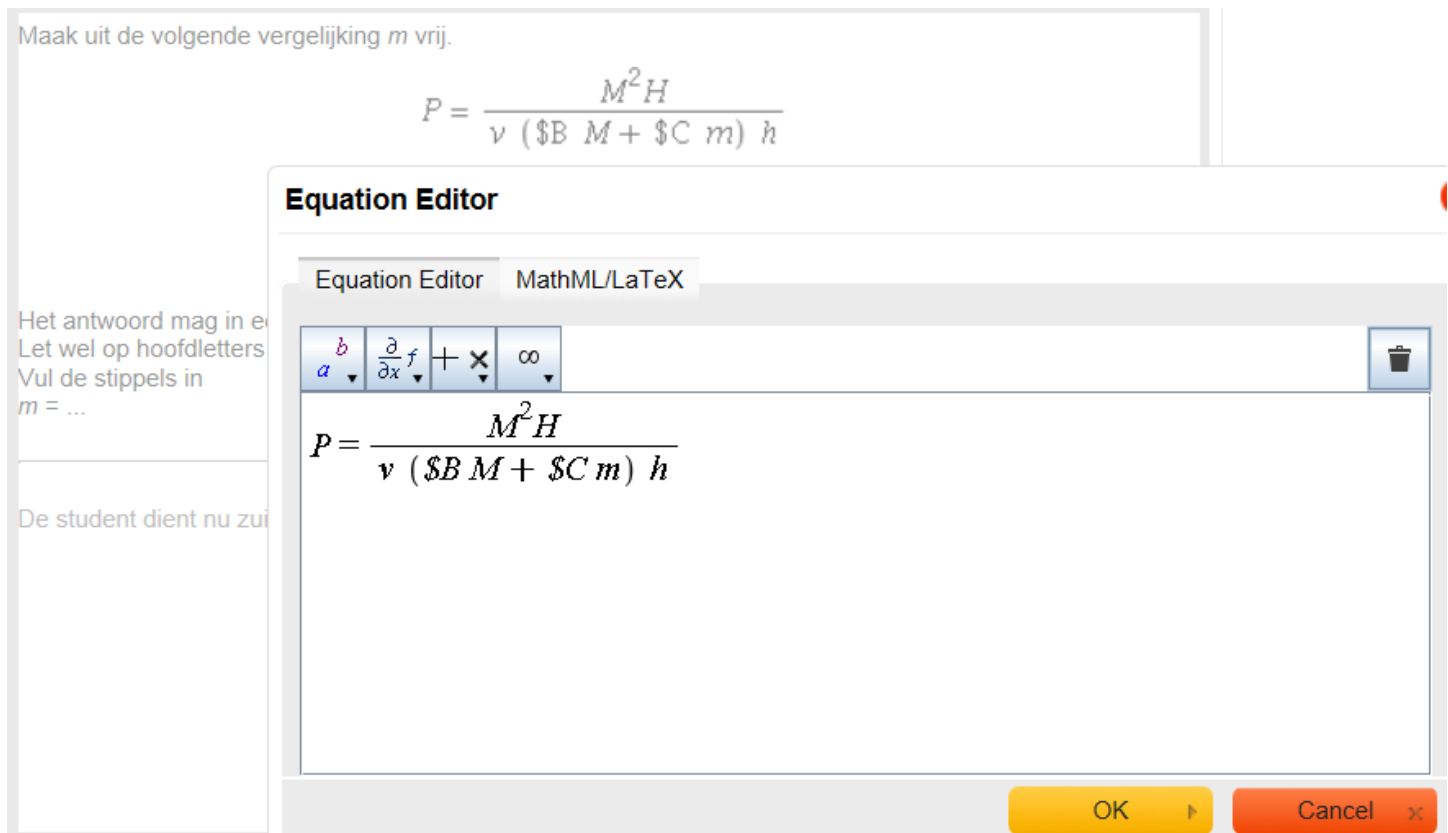


Figure 1.42: De Equation Editor

In bovenstaande figuur is te zien dat er in de formule zelfs ook variabelen (te herkennen aan het dollarteken) opgenomen kunnen worden. Met het klikken op een bestaande formule en vervolgens op de knop van het *Sigmatteken*, kan de formule aangepast worden. Of klikken op de knop van het *Sigmatteken* kan een nieuwe formule aangemaakt worden. Er zijn veel mogelijkheden om speciale tekens met de knoppen in deze Editor te maken.

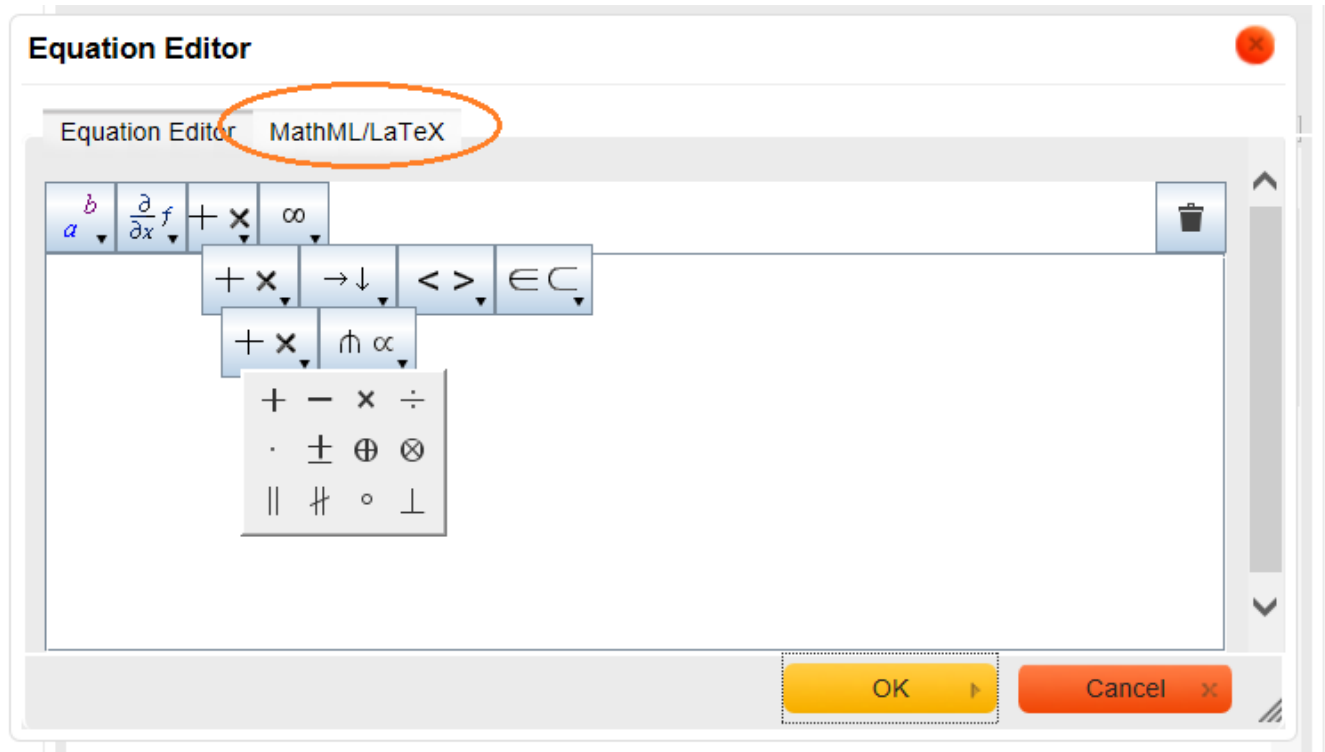


Figure 1.43: Vernieuwde Equation Editor in Maple T.A. 10

TIP: Met het tabblad MathML/LaTeX in de *Equation Editor*, Figure 1.42 (page 34) kan de MathML-code nog aangepast worden bijvoorbeeld om de formule wat groter te maken of deze variabelen niet cursief te maken. Ook is het mogelijk om in dit tabblad de code te plakken die verkregen is uit een ander programma waar MathML-code is aangemaakt bijvoorbeeld in Maple of in MathType.

TIP: Als u aanpassingen doet in deze MathML-code in het tabblad MathML/LaTeX van de formule-Editor, ga dan vanuit dit tabblad van de MathML-code direct op *OK* klikken om de formule te bevestigen en niet eerst weer terug naar de *Equation Editor*, want dan komen er wellicht ongewenste conversies.

TIP: Met het maken van een formule op deze manier met variabelen erin, worden de variabelen als vanzelf cursief weergegeven, zoals $\$B$ en $\$C$ in Figure 1.42 (page 34). Dat is echter niet wenselijk als deze variabelen staan voor getallen, want in de vraag zullen in dit geval voor deze $\$B$ en $\$C$ getallen worden ingevuld en die moeten beslist niet cursief!

Met het tabblad MathML in de *Equation Editor* kan de MathML-code nog aangepast worden, maar de MathML-code die in het tabblad MathML staat van de *Equation Editor* is vrij ingewikkeld. Op de een of andere manier wordt deze code weer omgezet in een eenvoudiger code die te vinden is in de broncode van het item die te bereiken is met *Edit Source* direct nadat op *Edit* geklikt is bij het openen van de vraag. Het is dus veel handiger om de MathML-code aan te passen in de broncode van de vraag. De MathML-code is daar veel schoner en transparanter en er kan bijvoorbeeld aan toegevoegd worden: `fontSize="14"` of cursieve variabelen kunnen weer niet cursief gemaakt worden door de tags `<mi> . . . </mi>` te veranderen in `<mn> . . . </mn>`, zie de volgende figuur. Bijvoorbeeld `<mi>sin</mi>` moet u dan veranderen in `<mn>sin</mn>` en in dit geval bijvoorbeeld `<mi>\$C</mi>` veranderen in `<mn>\$C</mn>`.

`<mi>` betekent mathematical italic en `<mn>` betekent mathematical normal.

Cancel

Save

Edit Question Source

You can view and edit the raw data fields of your question below.

WARNING: If you enter invalid field or value data, you may make your questions unworkable.

```
question=<p>Maak uit de volgende vergelijking <em>m</em> vrij.</p>
<p align="center"><br />
<math xmlns='http://www.w3.org/1998/Math/MathML'><mstyle
fontsize="14"><mrow><mi>P</mi><mo>=</mo><mfrac><mrow><msup><mi>M</mi><mn>2
</mn></msup><mo>&InvisibleTimes;</mo><mi>H</mi></mrow><mrow><mrow><mi>v</mi><mi>v</mi><mo>&InvisibleTimes;</mo><mi>M</mi></mrow><mo>+</mo><mrow><mn>$B</mn><mo>&InvisibleTimes;</mo><mi>M</mi></mrow><mo>+</mo><mrow><mn>$C</mn><mo>&InvisibleTimes;</mo><mi>M</mi></mrow></mrow></mfrac></mrow></mstyle></math></p>
<p><br />
Het antwoord mag in een zelfgekozen vorm ingetikt worden.<br />
Let wel op hoofdletters en kleine letters!<br />
Vul de stipfels in<br />
<em>m</em> = ...</p>
```

Figure 1.44: MathML-code aanpassen in de broncode van de vraag

1.2.3.7 MathML-code met MathType

Soms krijgt u de formule niet goed op het scherm als u alles al geprobeerd hebt in de rubriek *Algorithm* waar de formules kunnen worden voorbereid. En lukt het ook niet met de *Equation Editor*, dan is er nog wel een andere methode om toch de formule die u wilt in de tekst van de vraag of in de hints of de feedback te krijgen.

Als u gewend bent het programma MathType te hanteren, hét programma om geavanceerde formules te bouwen, kunt u daarmee MathML-code genereren en dan de code overbrengen naar de plek waar u de formule wilt hebben.

Het programma MathType heeft een functionaliteit waarbij gemakkelijk de MathML-code te genereren is.

Deze code is ook vrij schoon en gemakkelijk te overzien in de Editor.

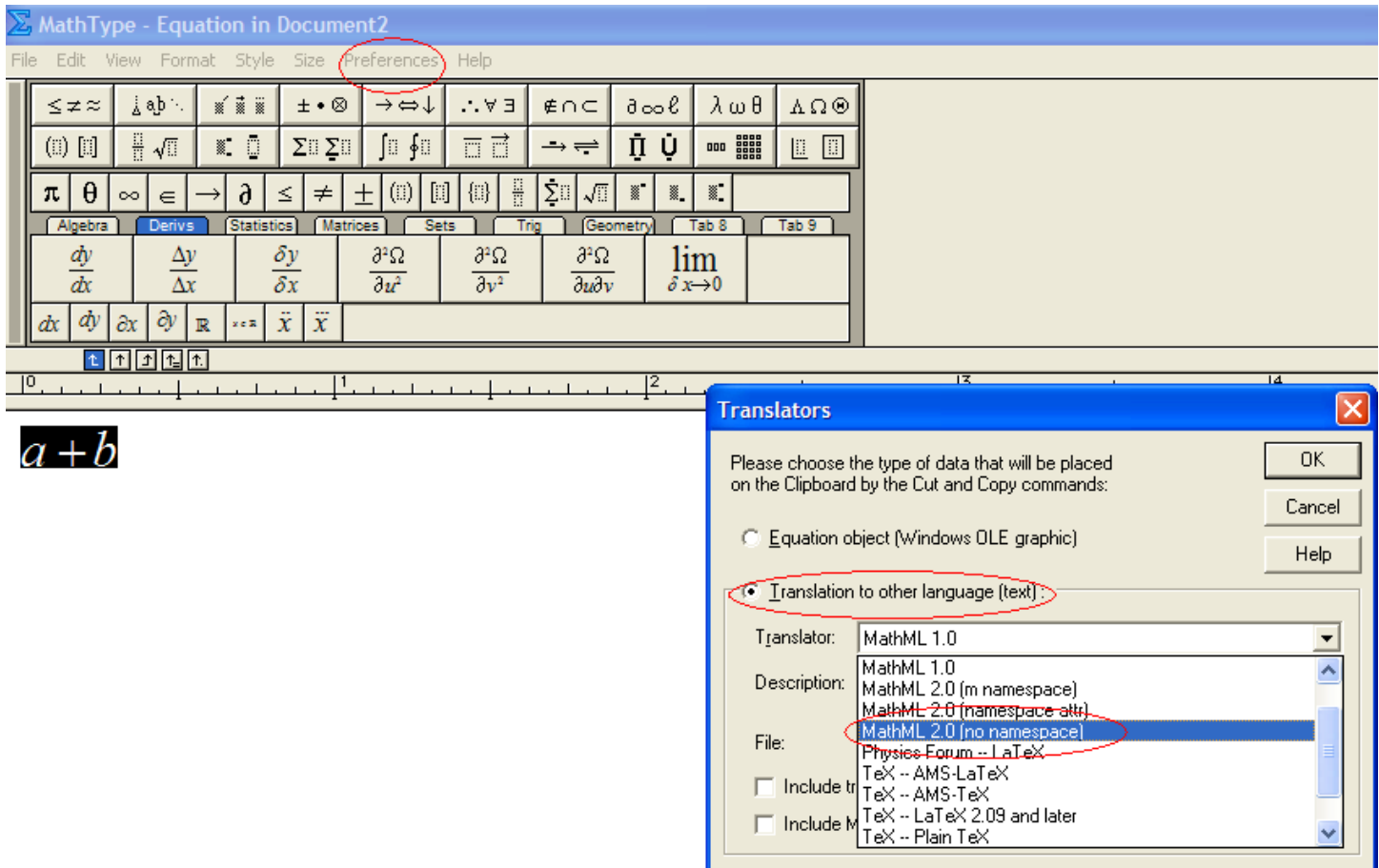


Figure 1.45: Met MathType de MathML-code maken

Start het programma MathType en ga naar *Preferences* en kies voor *Translators*.

Zorg ervoor dat de volgende instellingen gedaan worden: zie *Figure 1.45* (page 37).

Met de radiobutton kiest u voor *Translation to other language* en kies dan voor de *Translator: MathML -- WebEQ compatible of MathML 2.0 [no namespace]*. De checkboxes onderaan hoeven niet aangevinkt te worden. Als u nu in MathType een formule bouwt, kunt u deze formule gewoon selecteren, kopiëren en vervolgens in de *Equation Editor* van Maple T.A. plakken, maar dan wel in het tabblad MathML!!! U kopieert en plakt in feite de MathML-code met als resultaat een formule in de formule-Editor. Met deze instellingen van MathType wordt er dus geen plaatje van gemaakt (gif-bestand) zoals u dat gewend bent met kopiëren naar een Word-bestand.

Bevestig dan direct deze formule dus direct vanuit het tabblad MathML in de *Equation Editor* van Maple T.A. dus zonder eerst weer terug te keren naar de *Equation Editor* zelf.

TIP: Gebruik liever geen dollartekens voor de randomvariabelen in de Editor want die zouden dan cursief worden weergegeven, maar voeg eventuele dollartekens later toe in de broncode van de vraag zoals *Figure 1.44* (page 36) laat zien. Doe dat dus niet in de *Equation Editor* maar in de source code van het item die te bereiken is met *Edit Source* vlak nadat u de vraag aangeklikt hebt in de Question Repository.

TIP. Ook als u bepaalde formules van de ene vraag naar de andere wilt kopiëren, doe dat dan met de MathML-code in de broncode van de vraag, dat is de snelste en mooiste manier.

1.2.3.8 MathML met Wiris

Er is een mogelijkheid om op de site van Wiris <<http://www.wiris.com/editor/demo/en/mathml-latex.html>> de gewenste formule te maken en dan meteen ook over de MathML-code te beschikken. Met Copy en Paste is een en ander direct in uw Equation Editor te importeren.

Soms is het nodig om het font aan te passen maar dat hoeft voor Maple T.A. niet.

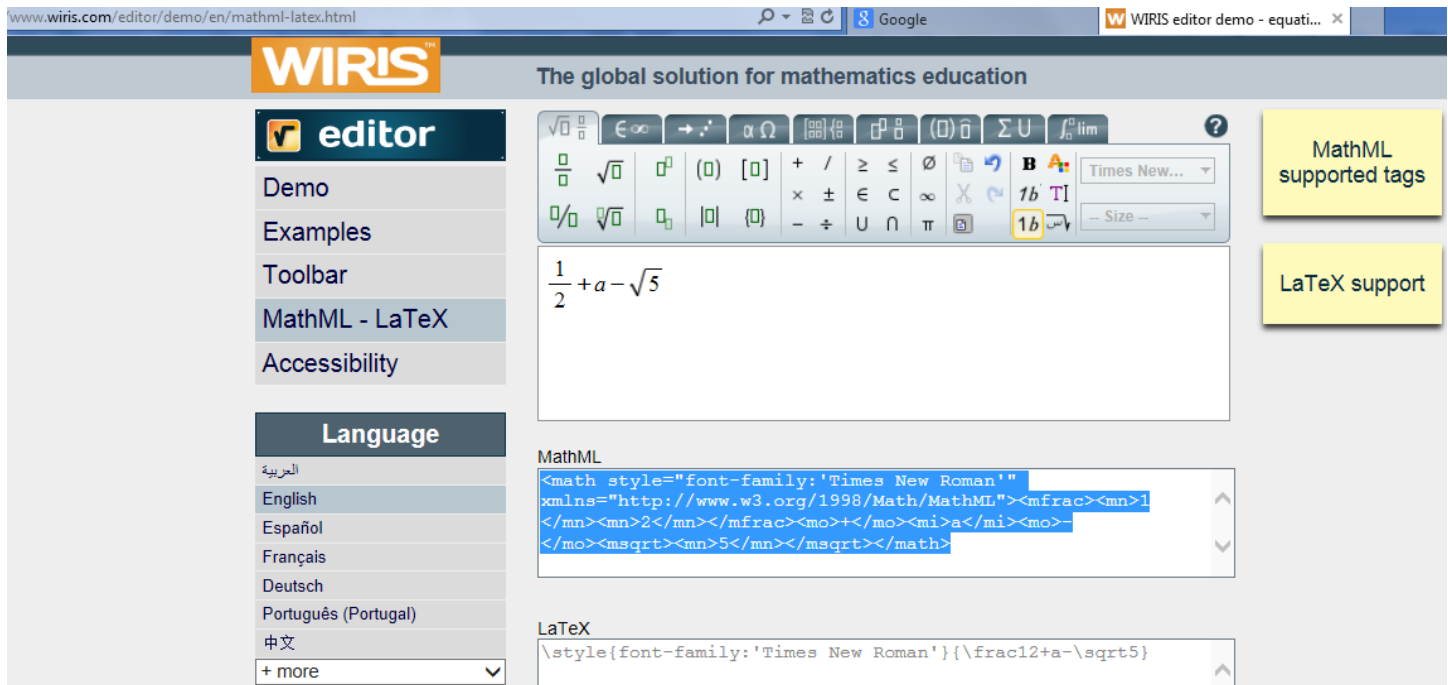


Figure 1.46: Formules met Wiris

Zoals in bovenstaande figuur te zien is, is het ook mogelijk om te beschikken over de LaTeX-code. Echter als u deze wilt gebruiken moet u nog wel open- en sluitingstekens toevoegen zoals te zien is in de volgende paragraaf. $\backslash(\dots)\backslash$.

In Maple T.A. wordt het font default Times New Roman gebruikt zonder dat dit geselecteerd wordt in de Wiris Editor.

1.2.3.9 LaTeX

Vanaf versie Maple T.A. 9.5 wordt LaTeX ondersteund en is er ook een mogelijkheid om in de *Equation Editor* LaTeX-code te gebruiken. De *Equation Editor* is fors uitgebreid en geeft ook een beter resultaat met het oog op in en uitzoomen.

U kunt ook de LaTeX-code gebruiken direct in de tekst maar ook in het *Algorithm* om formules voor te bereiden. Echter tikt u de code in de tekst van de vraag, en u wilt later daarin iets veranderen, dan moet u naar de broncode gaan om de veranderingen aan te brengen.

Begin een LaTeX-gecodeerde formule in de regel van de tekst altijd met een backslash \backslash gevolgd door een haakje openen (en vervolgens aan het eind van de formule afsluiten met een backslash \backslash gevolgd door een haakje sluiten).

$\backslash(\dots)\backslash$

Als de formule gecentreerd moet staan, tikt u in plaats van het openingshaakje het vierkante haakje en ook als sluitingshaakje het vierkante haakje.

$\backslash[\dots]\backslash$

Deze laatste manier is geschikt om grotere formules te maken en te centreren waarbij de subscript en superscript van integralen en sommaties ook echt onder en boven het teken staan (zie in *Figure 1.59 (page 44)* limieten en sommatie).

Maar natuurlijk kan centreren ook met een formule op een regel en daarna de tekst centreren.

De LaTeX-code kan dus gewoon in de tekst getikt worden, maar als u met de knop *Sigma* werkt, kunt u in de *Equation Editor* in het tabblad voor LaTeX met één het effect zien van de ingetikte formule en anders kunt u dat pas zien als de vraag opgeslagen is en u hem bekijkt hoe het geworden is.

Als u niet bekend bent met de LaTeX-code zie dan in de uitleg hieronder en om het helemaal gemakkelijk te maken kunt u de code ook genereren op de website van Wiris (zie vorige paragraaf) of bij de volgende website:

<https://www.codecogs.com/latex/eqneditor.php>

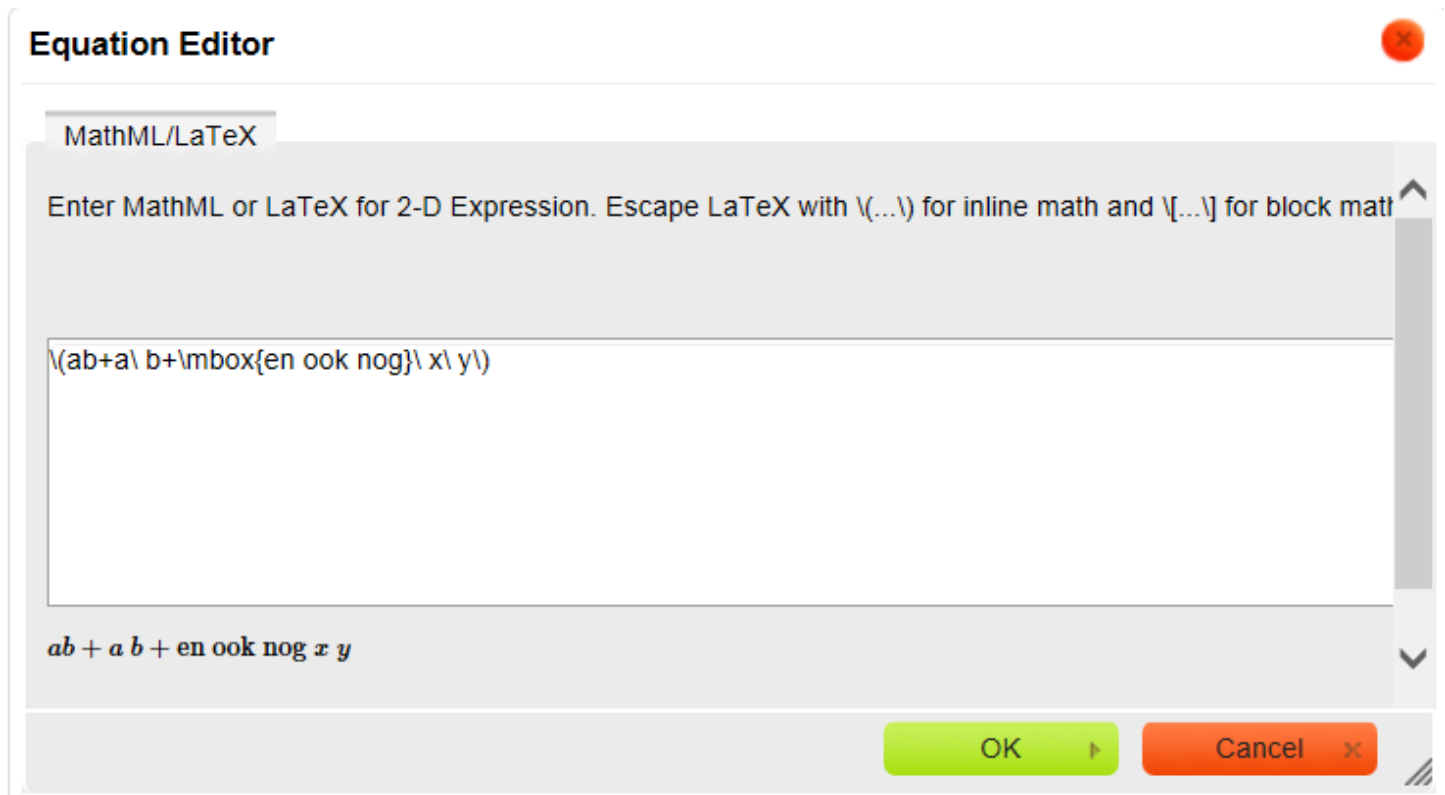


Figure 1.47: LaTeX met de Equation Editor

TIP: Eventueel aanpassen van de LaTeX-code kan later altijd nog in de broncode van de tekst van de vraag (met *Source*).

TIP: U kunt zelfs algorithmische variabelen gebruiken in de LaTeX-code voor zover dat getallen betreft of letters bijvoorbeeld in het *Algorithm* kunt u een variabele formule definiëren:

```
$test="\(\frac{a}{b}\)";
$formulea="\(\frac{A_w}{L_{pp}}\cdot B_{mld}\)";
```

Met als resultaat:

| <pre>\$test="\(\frac{a}{b}\)"; \$formulea="\(\frac{A_w}{L_{pp}}\cdot B_{mld}\)";</pre> | |
|--|------------------------------------|
| Variable | Value |
| test | $\frac{a}{b}$ |
| formulea | $\frac{A_w}{L_{pp} \cdot B_{mld}}$ |

Figure 1.48: voorbeeld met LaTeX

TIP: Stukken formules die bij elkaar moeten blijven zet u tussen accolades.

TIP: Heel veel LaTeX-symbolen vindt u op internet: http://nl.wikibooks.org/wiki/LaTeX/Wiskundige_formules

TIP: U kunt ook in Maple zelf de formule laten converteren naar LaTeX als volgt:

```
> latex(Int(1/(x^2+1), x) = int(1/(x^2+1), x));
\int \! \left( {x}^{{2}+1 \right) ^{-1}\!,{\rm d}x=\arctan \left( x \right) \! \right)
```

U kunt deze output in de tekst kopiëren als u deze tussen de haakjes plaatst: `\(..... \)`.

Echter `\left` en `\right` zijn niet beslist noodzakelijk maar de haken worden dan zo nodig wel groter en `\!` haalt een spatie weg of maakt die iets kleiner.

U kunt deze tussen dubbele quotes ook in the Algorithm plaatsen zoals hieronder in de figuur te zien is.

`$test="\(\int \! \left({x}^{{2}+1 \right) ^{-1}\!,{\rm d}x=\arctan \left(x \right) \! \right)";`

| Variable | Value |
|----------|---------------------------------------|
| test | $\int (x^2 + 1)^{-1} dx = \arctan(x)$ |

Figure 1.49: LaTeX in het Algorithm voorbereiden

Een schat aan informatie vindt u op <http://en.wikibooks.org/wiki/LaTeX/Mathematics>

Verder bestaat er wel een formulier op Internet met nog meer tekens en letters die te maken zijn met behulp van LaTeX.

<http://omega.albany.edu:8008/Symbols.html>

of

<http://w2.syronex.com/jmr/tex/textsym.old.html>

of

<http://garsia.math.yorku.ca/MPWP/LATEXmath/latexsym.html>

1.2.3.9.1 Variabelen, spaties en tekst

Variabelen zijn altijd cursief en spaties ertussen krijgt u met `\` en een spatie daarachter.

Wilt u echter een stukje tekst (rechttop) hebben, dan kan dat ook maar dat moet met `\mbox{tekst}`.

Deze tekst is dan wel altijd standaard in het font Times New Roman.

Binnen de accolades voor de tekst worden gewoon spaties meegenomen, maar daarbuiten niet.

```
\(ab+a\ b+\mbox{en ook nog}\ x\ y\)
```

ab + a b + en ook nog x y

Figure 1.50: LaTeX met variabelen spaties en tekst

TIP: De tekst komt echter wel in het font Times te staan terwijl u voor de tekst van de vraag meestal het default schreefloze font gebruikt op internet. Als u voor incidentele situaties voor een bepaalde letter in de tekst van de vraag het font Times wilt gebruiken, kunt u eventueel een kleine LaTeX-formule maken.

Voor niet cursieve letters kunt u gebruiken (rm betekent roman) `\rm niet cursief`

1.2.3.9.2 Superscript en subscript

Superscript gaat gewoon met `^` en subscript met `_` (underscore).

`\(a^b+c^{\text{bc}}+a_b+a_{\text{bc}}\)`

$$a^b + c^{bc} + a_b + a_{bc}$$

Figure 1.51: LaTeX met superscript en subscript

Deze superscript en subscript komt straks ook terug bij integralen en sommatie.

1.2.3.9.3 Vectoren en onderstrepingen

In de volgende regel enkele voorbeelden voor de versieringen van formules.

`\(\vec{v}+\vec{ab}+\overrightarrow{abc}+\overline{abc}+\underbrace{abc}+\underbrace{abc}_{\text{mbox {tekst}} }+\widehat{P}\)`

$$\vec{v} + \vec{ab} + \overrightarrow{abc} + \overline{abc} + \underbrace{abc} + \underbrace{abc}_{\text{tekst}} + \widehat{P}$$

Figure 1.52: LaTeX met vectoren en onderstreping

Eventueel met stip of dubbele stip: `\(\dot{a}+\ddot{b} \)`

$$\dot{a} + \ddot{b}$$

1.2.3.9.4 Vectoren en matrices

Een paar voorbeelden om kolomvectoren en zelfs matrices te maken met latex. Hoewel met Maple ook een goede manier is.

Twee dimensionale vectoren kan bijvoorbeeld met `binom`:

`"\(\binom{4}{3}+\binom{a_1}{a_2,a_3} + {32 \brack 5} + {a \brace b} \)`";

U kunt ook met `pmatrix` (gewone haken) of met `bmatrix` (vierkante haken) een matrix maken met één of meer kolommen:

`"\(\begin{pmatrix} a\\b\\c \end{pmatrix} \)`";

```
"\(\begin{pmatrix}
m_{11} & m_{12} & m_{13} \\
m_{21} & m_{22} & m_{23} \\
m_{31} & m_{32} & m_{33}
\end{pmatrix} \)
```

```

$test="\(\binom{4}{3}+\binom{a_1}{a_2,a_3} + {32 \brack 5} + {a \brace b} \)";
$test2="\(\begin{pmatrix} a\\b\\c \end{pmatrix} \)";
$test3="\(\begin{bmatrix} a\\b\\c \end{bmatrix} \)";
$test4="\(\begin{pmatrix}
    m_{11} & m_{12} & m_{13} \\
    m_{21} & m_{22} & m_{23} \\
    m_{31} & m_{32} & m_{33}
\end{pmatrix} \)";

```

| Variable | Value |
|----------|---|
| test | $\left(\binom{4}{3} + \binom{a_1}{a_2, a_3} + \left[\begin{smallmatrix} 32 \\ 5 \end{smallmatrix} \right] + \left\{ \begin{smallmatrix} a \\ b \end{smallmatrix} \right\} \right)$ |
| test2 | $\left(\begin{pmatrix} a \\ b \\ c \end{pmatrix} \right)$ |
| test3 | $\left(\begin{bmatrix} a \\ b \\ c \end{bmatrix} \right)$ |
| test4 | $\left(\begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \right)$ |

Figure 1.53: Kolomvectoren en matrices met LaTeX

1.2.3.9.5 Functies en commando's cursief en niet cursief

Als het bekende functies zijn zoals sin, cos, log en dergelijke, dan horen die natuurlijk niet cursief weergegeven te worden.

Voorafgegaan door een backslash wordt deze automatisch door LaTeX gezien als officiële functie.

Hou als gewoonte altijd haakjes om het argument van een functie.

Hieronder enkele voorbeelden van het effect ervan.

```
\(\sin(x)+\sin(3x^2)\)
```

```
\(\log_a(x)+\ln(3\ y)+\Re(z)\ +\Re(z)+\mbox{Re}\ +\mathrm{i}\ +\log(x_z)\)
```

$$\sin(x) + \sin(3x^2)$$

$$\log_a(x) + \ln(3\ y) + \Re(z) + Re(z) + Re + i + \log(x_z)$$

Figure 1.54: LaTeX met functies en commando's

Het is wel mooi om een spatie te maken tussen 3 en y bijvoorbeeld bij $\ln(3\ y)$. Dat gaat dus met een backslash voor de spatie. Het zou eigenlijk ook moeten bij $3x^2$.

Voor het getal van Euler, dat niet cursief moet, kunnen we natuurlijk gewoon tekst doen, maar beter is nog `\mathrm{e}` te nemen (mathematics roman), hoewel er eigenlijk geen verschil is. Echter met `\mbox{}` kunt u gewone teksten met spaties tikken en `\mathrm{}` is meer voor formules. Tussen accolades moet komen te staan wat er dus vervolgens rechttop afgebeeld moet worden.

```
\(\exp(x)+e^x+\mbox{e}^x+\mathrm{e}^x\)
```

```
\(\mathrm{e}^{\mathrm{x}}+\mathrm{Re}+\mathrm{I}+e^y+\mathrm{e}^{\mathrm{\mathrm{y}}}\)
```

$$\exp(x) + e^x + e^x + e^x$$

$$e^x + \text{Re} + \text{I} + e^y + e^y$$

Figure 1.55: LaTeX met cursief en niet-cursief

1.2.3.9.6 Breuken en wortels

Een breuk kan gemaakt worden met de code `\frac{}{}`.

Tussen het eerste paar accolades komt de teller en tussen het tweede paar accolades de noemer (geen komma ertussen!).

De n-de machtswortel gaat met `\sqrt[n]{}{}`. Tussen de accolades het gehele argument, zodat de streep van het wortelteken doorgetrokken wordt. Als het om een gewone vierkantswortel gaat, dan kan `[]` worden weggelaten.

`\(\frac{abc}{a+b+c}+\sqrt{124}+\sqrt[3]{29}\)`

$$\frac{abc}{a+b+c} + \sqrt{124} + \sqrt[3]{29}$$

Figure 1.56: LaTeX met breuken en wortels

1.2.3.9.7 Ongelijkheden

TIP: Bij het tikken van een ongelijkheid in de tekst even letten op spatiering rond de kleiner dan-tekens en groter dan-tekens, anders wordt het in de tekst niet goed opgevat. Ook bij LaTeX een spatie tikken rond deze tekens.

`>= \geq`

`<= \leq`

`< \lt`

`> \gt`

`\infty`

`\cup`

`\cap`

1.2.3.9.8 Limieten, sommatie en producten

Voor de limiet is er het commando: `\lim{x \to 0}` en daarna kan de formule gemaakt worden. (Eventueel met haakjes eromheen maar dat hoeft meestal niet.)

Voor de limiet met het pijltje naar onder of naar boven is het `\lim{x \uparrow 0}` resp. `\downarrow` enz.

`\(\lim_{x \to 0} \frac{\sin x}{x} + \lim_{x \uparrow 0} \frac{\sqrt{x^2}}{x} + \lim_{x \downarrow 0} \frac{\sqrt{x^2}}{x}\)`

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} + \lim_{x \uparrow 0} \frac{\sqrt{x^2}}{x} + \lim_{x \downarrow 0} \frac{\sqrt{x^2}}{x}$$

Figure 1.57: LaTeX met limieten

Ook de commando's `\sum` en `\prod` zijn bekende commando's en met subscript en superscript kunnen de grenzen meegegeven worden.

`\(\sum_{i=2}^5 2^i = 2^2 + 2^3 + 2^4 + 2^5 = 60\)`

`\(\prod_{i=2}^5 2^i = 2^2 \cdot 2^3 \cdot 2^4 \cdot 2^5 = 16384\)`

$$\sum_{i=2}^5 2^i = 2^2 + 2^3 + 2^4 + 2^5 = 60$$

$$\prod_{i=2}^5 2^i = 2^2 \cdot 2^3 \cdot 2^4 \cdot 2^5 = 16384$$

Figure 1.58: LaTeX met sommatie en product

Let ook eens op de `\cdot` voor de middenstip voor vermenigvuldiging. Voor drie stipjes kunt u `\cdots` gebruiken. Beslist geen gewone stip gebruiken in verband met verwarring met decimale punt.

Let ook eens op het verschil in afgebeelde formule als deze gecentreerd wordt. Dan begint u de LaTeX-code niet met `\(` maar met `\[` en eindigen met `\]` in plaats van `\)`. Het effect is dat bij sommatie en integratie de grenzen er niet achter maar onder en boven geplaatst worden met dezelfde code.

`\[\sum_{i=2}^5 2^i = 2^2 + 2^3 + 2^4 + 2^5 = 60\]`

$$\sum_{i=2}^5 2^i = 2^2 + 2^3 + 2^4 + 2^5 = 60$$

Figure 1.59: LaTeX met gecentreerde formule

U kunt ook bereiken dat de grenzen voor integralen en sommatie echt boven en onder worden geplaatst door de extra optie met `limits`:

U tikt dan: `\(\sum\limits_{i=2}^5 2^i = 2^2 + 2^3 + 2^4 + 2^5 = 60\)`

1.2.3.9.9 Differentiëren en integreren

Het integratiecommando is `\int`. Met subscript en superscript kunnen de grenzen van de integraal aangegeven worden (al of niet met accolades eromheen).

Om aan te geven waarnaar geïntegreerd wordt, gaat met een komma voorafgegaan door een backslash, maar mag ook met een spatie voorafgegaan door een backslash.

Omdat de d van dx niet cursief moet, maken we daar weer `\mathrm{d}` van met `\mathrm{d}`.

`\(\int_a^b f(x) \mathrm{d}x\)` maar hetzelfde effect geeft de volgende code:

`\(\int_a^b f(x) \mathrm{d}x\)`

$$\int_a^b f(x) dx$$

Figure 1.60: LaTeX met integralen

Een kringintegraal is ook aardig om te maken met `\oint_C`: `\(\oint_C f \mathrm{d}s\)`

$$\oint_C f ds$$

Figure 1.61: LaTeX met kringintegraal

Differentiëren gaat met behulp van een breuk `\frac{}{}` waarbij ervoor gezorgd moet worden dat de d niet cursief is:

`\(\frac{\mathrm{d}f(x)}{\mathrm{d}x}\)`

$$\frac{df(x)}{dx}$$

Figure 1.62: LaTeX en differentiëren

Oneindig symbool ∞ met `\infty`.

De stip (middot) met `\cdot`

Het officiële teken voor partiële differentiatie:

`$dzdx="(\frac{\partial z}{\partial x})$";`

2-D math, plot, image, or embedded component

$$\frac{dz}{dx}$$

Figure 1.63: partiële differentiatie

Grotere integraalteken krijg je met `$displayvraag="(\int \frac{1}{\sqrt{1-x^2}} \mathrm{d}x)$"`

1.2.3.9.10 Haakjes en pijltjes

Als u haakjes ergens omheen wilt, kan dat gewoon met (...) en vierkante haakjes met [...]. Ook accolades { ...} en absoluutstrepen kunt u met |...| maken.

Als u echter deze afsluittekens automatisch wilt laten meeschalen met de formule, dan moet u met een tag werken zoals `\left(.....\right)`.

`\left(1+\left(\frac{\sin(2x)}{\cos(3x)}-x\right)\right)`

$$1 + \left(\frac{\sin(2x)}{\cos(3x)} - x \right)$$

Figure 1.64: LaTeX en haakjes

Grote haken met `\bigl(f\bigr)`

pijltje met `\rightarrow`

1.2.3.9.11 Tekens en Griekse letters

De tekens en Griekse letters gaan ongeveer net zoals met html maar dan nu met een backslash ervoor.

`\left(\tan(\alpha)+\Delta+\delta\right)`

$$\tan(\alpha) + \Delta + \delta$$

Figure 1.65: LaTeX en tekens en Griekse letters

1.3 Vraagtype Maple-graded

Het vraagtype *Maple-graded* is een van de belangrijkste vraagtypen om formules te toetsen en biedt zeer veel mogelijkheden voor het stellen van open vragen met formules met randomisering die door middel van Maple beoordeeld worden op juistheid. Hierbij is er heel veel mogelijk met het programmeren van de grading met behulp van de zogenoemde *Grading Code*. Voor uitgebreide informatie over de mogelijkheden in de *Grading Code* zie betreffende paragraaf *Tips voor de Grading Code* (page 56).

Als u formules wilt overhoren gebruik dan hoofdzakelijk dit vraagtype en liever niet het vraagtype *Formula*.

Er is bij elk vraagtype (niet alleen bij *Maple-graded*) altijd wel de mogelijkheid voor het gebruik van randomvariabelen, zodat u met het maken van één toets item eigenlijk een hele familie van toets items maakt.

Het gaat erom de organisatie van de vraag zo simpel mogelijk te houden, zodat modificaties gemakkelijk gedaan kunnen worden (ook door anderen). De rubriek *Algorithm* is een belangrijk element binnen de organisatie van de vraag.

Soms kan met het aanpassen van één formule in de rubriek *Algorithm* een hele nieuwe familie van vragen verkregen worden als de rest van de organisatie van de vraag samenhangend en goed in elkaar zit.

Het vraagtype *Maple-graded* komt als zelfstandige vraag voor, maar is ook vertegenwoordigd binnen de *Question Designer* en zelfs ook binnen de *Adaptive Question Designer* met dezelfde functionaliteit. De zelfstandige vraag hoeft u niet meer te gebruiken en zal in volgende versies van Maple T.A. ook niet meer ondersteund worden.

In het algemeen is de grading van de door de student ingevoerde formule (respons) goed te regelen door middel van het checken van het verschil tussen het correcte antwoord en de respons van de student. Maple draait op de achtergrond om de onderliggende berekening van deze match te doen en daarom heet het vraagtype *Maple-graded*.

Als bijvoorbeeld het verschil tussen het correcte antwoord en de respons van de student gelijk is aan 0, is de ingevoerde formule in principe goed.

Er zijn echter heel veel meer mogelijkheden voor het matchen van het correcte antwoord met de respons van de student, afhankelijk van de vraagstelling. Zie voorbeelden in paragraaf *Tips voor de Grading Code* (page 56).

1.3.1 Algemene structuur van Maple-graded vraagtype binnen de Question Designer

We beginnen met een voorbeeld van een *Maple-graded* vraagveld.

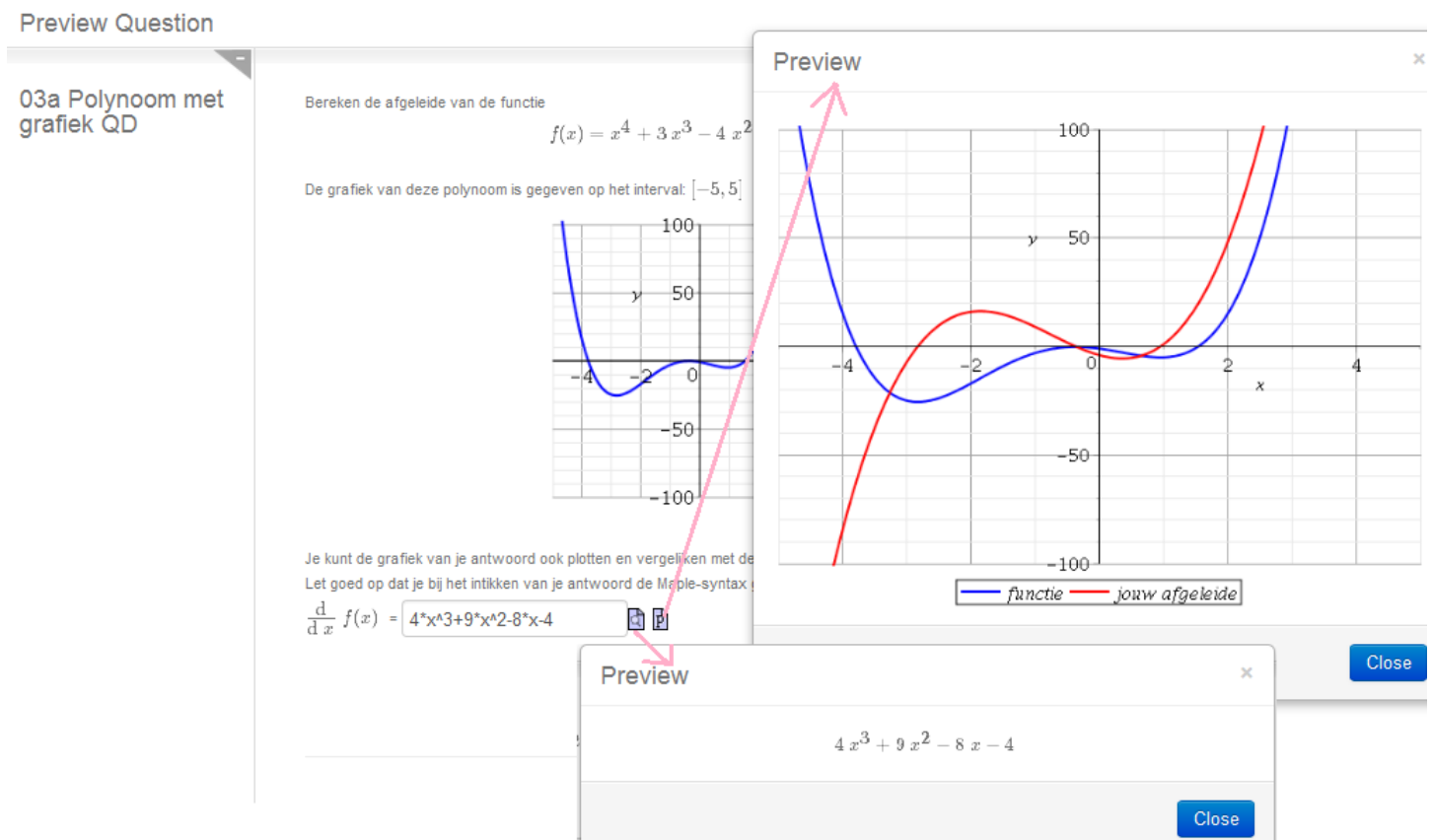


Figure 1.66: Voorbeeld van een Maple-graded vraag

In bovenstaande figuur is een vraag te zien met een formule en een grafiek. Op zich is dit niets bijzonders, want dit kan in elk vraagtype verwezendlijkt worden wat betreft de presentatie van formule en grafiek in de tekst van de vraag. Voor het maken van de formule, zie paragraaf *Formules* (page 18) en voor het maken van dynamische grafieken zie paragraaf *Dynamische figuren* (page 138). De formule en bijbehorende grafiek zijn beide gerandomiseerd, zodat elke keer als de vraag geopend wordt een andere functie tesamen met de bijbehorende grafiek tevoorschijn komt.

Wat er speciaal is aan dit vraagtype, is het invulveld waar de student een formule kan intikken.

Voor dit invulveld is er een drietal instellingen mogelijk. In Figure 1.66 (page 46) hierboven moet de student beslist *Maple syntax* gebruiken en er is voor de student nog een knop *P* beschikbaar waarmee hij eventueel een grafiek kan oproepen, zoals hier de grafiek van zijn eigen antwoord tesamen met de oorspronkelijke grafiek. In het volgende wordt uiteengezet hoe de instellingen daarvoor zijn.

Edit Response Area

Choose question type

- Clickable Image
- Essay
- Scanned Document
- Free Body Diagram
- HTML
- List
- Maple-graded**
- Matching
- Math App
- Mathematical Formula
- Multiple Choice
- Numeric
- Sketch
- Sorting
- True/False

Maple-graded:

Weighting: 1

Answer: \$f

(referenced when grading as \$ANSWER)

Grading Code: `evalb(simplify(($f)-($RESPONSE)) = 0) and evalb(0=StringTools[Search]("diff","$RESPONSE")) ;`

Expression Type: Maple Syntax - e.g. diff(2*f(x),x)

Text/Symbolic entry: Text entry only

Optional: Maple Repository:

Maple Repository

Plotting Code: `plot([$f,$RESPONSE],x=-5..5,y=-100..100,color=[blue]`

Custom Previewing Code:

OK Cancel

Figure 1.69: title of the figure

U kunt hier nog niet zoveel invullen als u nog niet precies weet wat u wilt. Daarom gaat u eerst voorbereidingen treffen in de rubriek *Algorithm* van de vraag.

1.3.2 Algorithme van het Maple-graded vraagveld

Ga nu eerst naar het *Algorithm* om de variabelen aan te maken die nodig zijn voor de vraag.

Algorithm

Edit the code for your algorithm in the text box to the right, or click "Show Designer" to use the algorithm designer. The algorithm designer tool allows you to define algorithms for your question by completing a form.

Show Designer

Refresh algorithm preview

```
$p = maple("randomize():RandomTools:-Generate( polynom(integer(range=-5..5),x,degree=3))");
$poly=maple("sort(($p)+x^4)");
$polydisplay = maple("printf(MathML:-ExportPresentation( f(x)=$poly))");
$antw=maple("diff(($poly),x)");
$displayantw = maple("printf(MathML:-ExportPresentation($antw))");
$displaydfdx=maple("printf(MathML:-ExportPresentation(diff(f(x),x)))");
$displayinterval=maple("printf(MathML:-ExportPresentation([-5,5]))");
$figuur=plotmaple("plot($poly,x=-5..5,y=-100..100,color=blue,thickness=2,gridlines=true),plotoptions='height=250, width=250' ");
```

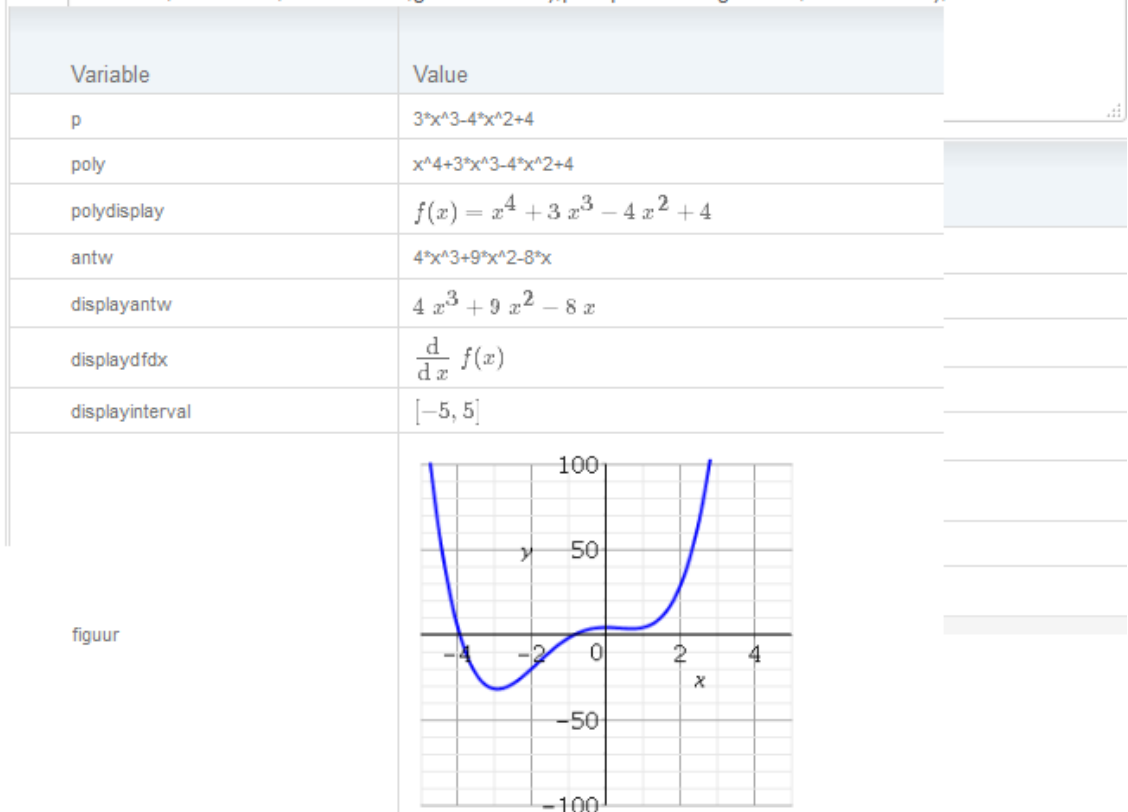


Figure 1.70: Het algoritme van een Maple graded vraagtype

In het *Algorithm* vult u het volgende in en als u op *Refresh algorithm preview* klikt, dan worden de variabelen, aangemerkt met een dollarteken, ververs en kunt u een en ander controleren.

```
$p=maple("randomize():RandomTools:-Generate( polynom(integer(range=-5..5),x,degree=3))");
$poly=maple("sort(($p)+x^4)");
$polydisplay=maple("printf(MathML:-ExportPresentation( f(x)=$poly))");
$antw=maple("diff(($poly),x)");
$displayantw = maple("printf(MathML:-ExportPresentation($antw))");
$displaydfdx=maple("printf(MathML:-ExportPresentation(diff(f(x),x)))");
$displayinterval=maple("printf(MathML:-ExportPresentation([-5,5]))");
$figuur=plotmaple("plot($poly,x=-5..5,y=-100..100,color=blue,thickness=2,gridlines=true),plotoptions='height=250, width=250' ");
```

Met behulp van Maple wordt een random derdegraads vorm gegenereerd. Vervolgens voegt u er x^4 aan toegevoegd en zet u het geheel op volgorde (sort), zodat de gegeven uitdrukking altijd begint met x^4 . De uiteindelijke vraag wordt weer MathML-gecodeerd en wordt gedefinieerd als de variabele \$polydisplay. Met behulp van een Maple-opdracht (diff) wordt het correcte antwoord \$antw gegenereerd

en vervolgens in de variabele $\$displayantw$ omgezet naar MathML-code ten behoeve van de *Feedback*. Zie voor het maken van dynamische grafieken in de *Handleiding Items Maken deel C*.

U kunt nu de algoritmische variabelen door het hele vraagstuk heen gebruiken door ze bij hun naam aan te roepen.

1.3.3 Question Text van het Maple-graded vraagveld

In de *Question Text* kan alsvast de formule en de figuur als variabele in de tekst gezet worden. Ook de voorbereide formule van $\$displayfdx$.

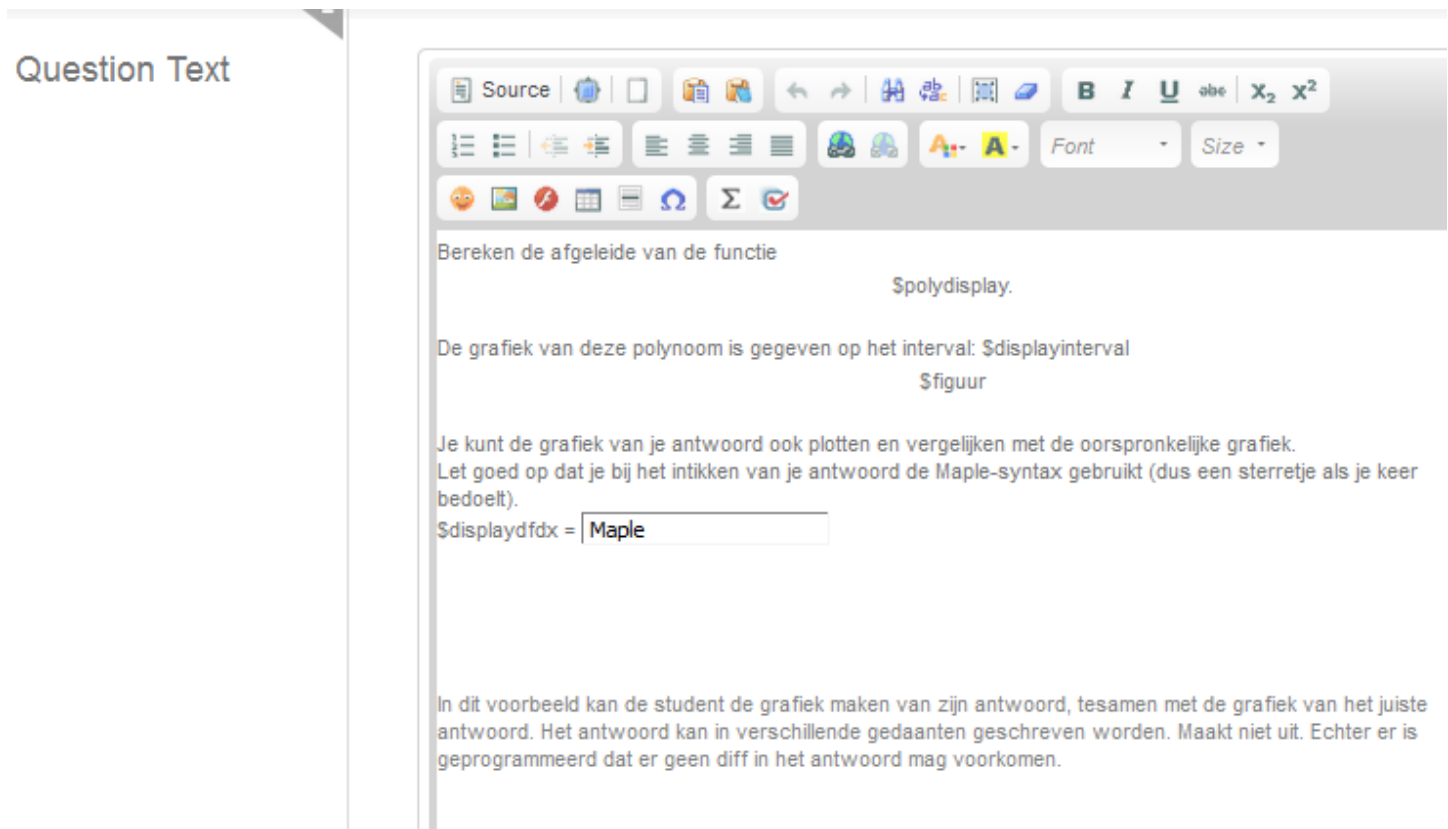


Figure 1.71: Tekst van de Maple Graded vraag

U bepaalt zelf waar het invulveld geplaatst wordt en eventueel stelt u nog meer vragen en maakt u meer invulvelden.

1.3.4 Answer van het Maple-graded vraagveld

In de rubriek *Answer* zorgt u ervoor dat u hier een formule plaatst die de student later zal zien als het correcte antwoord. Met de opdracht `printf(MathML:-ExportPresentation($antw));` zal dit antwoord in 2D in de feedback als correct antwoord gepresenteerd worden. Maar u kunt ook gewoon de formule $\$antw$ hier invullen en dan wordt in de feedback de formule getoond die de student ook ingetikt zou hebben.

Edit Response Area

Choose question type

- Essay
- Free Body Diagram
- List
- Maple-graded
- Math App
- Mathematical formula
- Multiple Choice
- Numeric

Maple-graded:

Weighting:

Answer (circled in red):

(referenced when grading as \$ANSWER)

Grading Code:

Figure 1.72: Het correcte antwoord in 2D

In deze rubriek moet u beslist een Maple-opdracht geven en u dient af te sluiten met een puntkomma. Dit resultaat wordt automatisch opgeslagen als de variabele `$ANSWER` en u kunt deze variabele eventueel gebruiken bij de rubriek *Grading Code* om het antwoord van de student mee te matchen. In dit voorbeeld is in deze rubriek echter de MathML-code geprogrammeerd `printf(MathML:-ExportPresentation($antw))`; zodat de student dan de 2D-versie van het correcte antwoord op zijn scherm te zien krijgt in de feedback als zijnde het correcte antwoord in geval het gegeven antwoord fout was. Daarmee kunt u niet het antwoord van de student matchen.

TIP: Het antwoord dat u in de rubriek *Correct answer:* invult kan alleen gebruikt worden in de *Grading Code* als het een echte formule is (dus niet als MathML-code). Het is ook verplicht deze rubriek in te vullen.

1.3.5 Grading Code van het Maple-graded vraagveld

In de *Grading Code* voert u in: `evalb(simplify(($RESPONSE)-($antw))=0)` and `evalb(0=StringTools[Search]("diff", "$RESPONSE"));`

Edit Response Area

Maple-graded:

- Essay
- Free Body Diagram
- List
- Maple-graded
- Math App
- Mathematical formula
- Multiple Choice
- Numeric
- Sketch

Weighting: 1

Answer: \$antw
(referenced when grading as \$ANSWER)

Grading Code:
`evalb(simplify(($RESPONSE)-($antw))=0) and
evalb(0=StringTools[Search]("diff", "$RESPONSE"));`

Expression Type: Maple Syntax - e.g. diff(2*f(x),x)

Text/Symbolic entry: Text entry only

Figure 1.73: Grading Code van de Maple-graded vraag

De **Grading Code** is de rubriek waar u Maple-code moet invoeren om tot de grading te komen van het antwoord van de student. Deze *Grading Code* is erg belangrijk bij het vraagtype *Maple-graded* en biedt tal van mogelijkheden voor de meest uiteenlopende situaties. In dit geval wordt de respons van de student (per definitie de automatisch gegenereerde variabele \$RESPONSE) vergeleken met het juiste antwoord \$antw dat al voorbereid was in de rubriek *Algorithm*. Van elkaar afgetrokken en het resultaat ervan vereenvoudigd, moet dat 0 opleveren. De opdracht *evalb* (evalueer boolean) geeft uitsluitsel (true of false).

`evalb(simplify(($RESPONSE)-($antw))=0);`

TIP: Zie voor meer mogelijkheden van *Grading Code* in paragraaf *Tips voor de Grading Code* (page 56).

TIP: In dit geval (Figure 1.72 (page 51)) kunt u in de *Grading Code* beslist niet voor het correcte antwoord de variabele \$ANSWER gebruiken, want het correcte antwoord is in dit geval een MathML-code en geen formule waarmee gerekend kan worden.

TIP: Echter deze programmering met `evalb(simplify(($RESPONSE)-($antw))=0);` is hier nog niet toereikend, want bij de instellingen voor *Expression type* (zie paragraaf *Expression type in de Maple-graded vraag* (page 53)) is gekozen voor *Maple syntax* én voor *Text entry only*, wat impliceert dat de student ook Maple-opdrachten (zoals `diff(...)`) zou kunnen intikken om tot het goede antwoord te komen. Vandaar dat er ook nog geprogrammeerd dient te worden dat het antwoord de karaktercombinatie "diff" niet bevat. Immers de student zou het antwoord met pen en papier zelf moeten kunnen berekenen. Dit kan geprogrammeerd worden met de extra toevoeging `....and evalb(0=StringTools[Search]("diff", "$RESPONSE"))`. Het betekent dat er in de string van het antwoord van de student "\$RESPONSE" gezocht wordt naar de string "diff" en dat het resultaat 0 moet zijn. (Een aantal karakters tussen quotes is een string).

Als u gekozen had voor *Symbol entry only*, werd er in de vraag een Editor aangeboden waarin de student de formule zelf kan maken. Daarin kunnen (tot op zekere hoogte) geen Maple-opdrachten opgenomen worden en dan is deze extra programmering

vaak niet nodig, maar bij differentiëren juist weer wél want de student kan in de *Symbol Mode* met de knop $\frac{\partial}{\partial x}$ aangeven dat er gedifferentieerd moet worden en dus toch nog de opdracht aan Maple uitbesteden. De extra restrictie is dus bij "diff" bij beide instellingen nodig: *Text Mode* én *Symbol Mode*.

Zie voor uitgebreide informatie over de *Grading Code* in paragraaf *Tips voor de Grading Code* (page 56).

1.3.6 Expression type van het Maple-graded vraagveld

Kies in de meeste gevallen voor de instelling *Maple Syntax* en in het algemeen beslist niet voor *Formula*! Deze laatste is namelijk zeer zwak en niet altijd correct en kan zelfs geen karaktercombinaties aan.

Ook geeft het systeem vervelende errors als *Formula* ingesteld is en de student vult iets in dat niet aan de syntax voldoet. Dus in het algemeen voor *Maple syntax* kiezen. Vervolgens kiest u *Text entry only* als de student Maple syntax in een tekstveld moet intikken met de mogelijkheid voor Preview. Of anders biedt u de Editor aan bij de instelling *Symbolic entry only*.

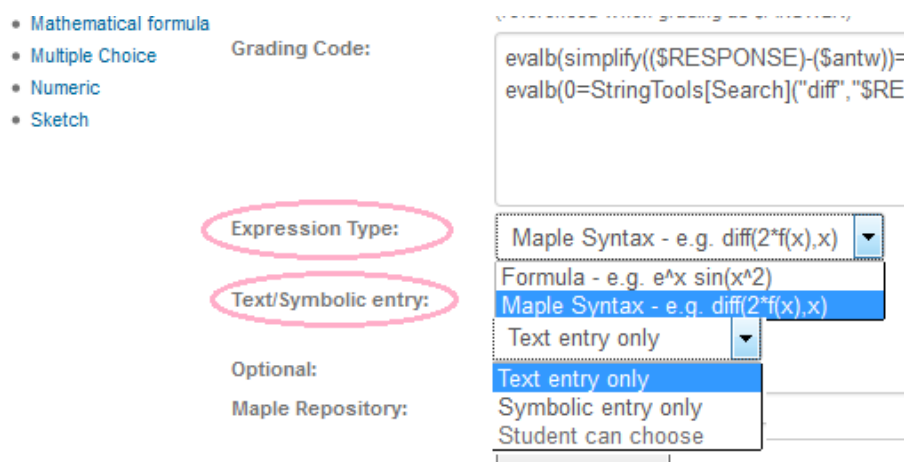


Figure 1.74: Instellingen voor Maple syntax bij Maple-graded vraagtype

Zie verder voor de instellingen in paragraaf *Instellingen van het formule-invulveld* (page 1).

1.3.7 Maple Repository van het Maple-graded vraagveld

Het is mogelijk om vooraf Maple-functies te definiëren en deze in de op de server in de File Manager te uploaden. In de vraag kunt u deze importeren en ernaar verwijzen. U doet dat als u ingewikkelde opdrachten aan Maple wilt overlaten. Hoe dat in zijn werk gaat, kunt u lezen in de *Handleiding Items Maken deel C*.

1.3.8 Plotting Code van het Maple-graded vraagveld

Onderaan is nog een rubriek **Plotting Code**. Hier kunt u een of meer Maple-opdrachten formuleren (sluit elke opdracht af met dubbele punt en de laatste opdracht met een puntkomma) waarmee een grafiek kan worden gegenereerd als de student in de vraag op *Plot* klikt. In deze opdracht kunt u variabelen uit de rubriek *Algorithm* aanroepen en de grafiek kunt u vrij uitgebreid programmeren.

Edit Response Area

- List
- Maple-graded
- Math App
- Mathematical formula
- Multiple Choice
- Numeric
- Sketch

Answer

Grading Code:

Expression Type:

Text/Symbolic entry:

Optional:

Maple Repository:

Plotting Code:

Custom Previewing Code:

```
printf(MathML:-ExportPres
```

(referenced when grading as \$ANSWER)

```
evalb(simplify(($RESPONSE)-($antw))=0) and
evalb(0=StringTools[Search]("diff",$RESPONSE));
```

Maple Syntax - e.g. diff(2*f(x),x)

Text entry only

Maple Repository

```
s=2,legend=['functie`,`jouw afgeleide`],gridlines=true);
```

Figure 1.75: Plotting Code in de Maple-graded vraag

In deze opdracht voor de plotting is zelfs ook de formule van de student aangeroepen (\$RESPONSE) en de grafiek van zowel de respons van de student als de oorspronkelijke grafiek van \$poly worden afgebeeld bij het klikken op *Plot*.

Deze *Plotting Code* is een maple opdracht afgesloten met puntkomma.

```
plot([$poly,$RESPONSE],x=-5..5,y=-100..100,color=[blue,red],thickness=2,legend=['functie`,`jouw afgeleide`],gridlines=true);
```

TIP: Voor een goed gebruik van de plotting faciliteiten is het meest handig als u de formule-instellingen op *Maple syntax* met *Text entry only* zet, maar met de andere instellingen werkt het meestal ook goed. Verstandig is het om altijd even uit te proberen. In de meeste gevallen gaat het ook goed als u gekozen had voor *Symbolic entry only* en zelfs bij de instelling *Formula*. De Editor vertaalt alles keurig naar *Maple syntax*.

TIP: Als u niets invult bij de *Plotting*, dan is het knopje *Plot* niet actief als de student de vraag voor zich krijgt.

TIP: Bij vectoren werkt de *Plotting Code* niet goed als u de Editor hanteert. Als u toch een grafiek wilt laten zien, neem dan de instellingen Maple-syntax text entry only.

1.3.9 Custom Previewing van de Maple-graded vraag

In het algemeen vult u in deze rubriek niets in, maar een enkele keer zult u het nodig hebben om hier iets in te programmeren.

expression type: Maple Syntax - e.g. $\text{diff}(2^x, x)$

Text/Symbolic entry: Text entry only

Optional:

Maple Repository:

Plotting Code: `plot([$poly,$RESPONSE],x=-5..5,y=-100..100,color=`

Custom Previewing Code: `1 $RESPONSE: end: printf(InertForm:-ToMathML(%));`

Figure 1.76: Custom Previewing Code

Als u niets invult, en de student klikt op Preview na het invullen van zijn formule, dan krijgt de student de formule te zien in 2D formaat. Echter u bent dan wel afhankelijk van de automatische vereenvoudiging die Maple doet. Wilt u de vereenvoudiging tegenhouden, dan kunt u het volgende programmeren:

```
use InertForm:-NoSimpl in $RESPONSE: end: printf(InertForm:-ToMathML(%));
```

Probeer het wel even uit maar in de meeste gevallen zal de *Preview* de niet-vereenvoudigde formule die de student heeft ingetikt weergeven in 2D.

Ook is het mogelijk om vooraf geprogrammeerde Maple-opdrachten te importeren en werken met Maple Repositories (zie Handleiding Items Maken deel C.)

U kunt bijvoorbeeld ook nog iets anders hierin programmeren. U vraagt bijvoorbeeld naar een vector die ingevoerd moet worden met $[a,b]$. Als u dan de volgende code invult bij Custom Previewing:

```
if evalb(StringTools[CountCharacterOccurrences]("$RESPONSE", "[") < 2) then
  printf(MathML[ExportPresentation](Vector($RESPONSE))) else printf("tik één vector") end
if;
```

Dan kunt u ook nog andere dingen communiceren hierin. Zie verder voor het overhoren van vectoren in paragraaf *Matrices en Vectoren* (page 124).

1.3.10 Feedback van de vraag

Feedback

Students will be shown this feedback when they view their graded assignments. You can use the Assignment Editor to customize the level of feedback shown.

Source
↩ ↪
↺ ↻
↻ ↺
↻ ↺
↻ ↺
↻ ↺
↻ ↺
↻ ↺
↻ ↺

≡ ≡ ≡
≡ ≡ ≡
≡ ≡ ≡
≡ ≡ ≡
≡ ≡ ≡
≡ ≡ ≡
Font
Size

😊 📷 🔍 📅 📄 🔗 Σ

Gebruik de machttregel.

$\$displaydfdx = \$displayantw$

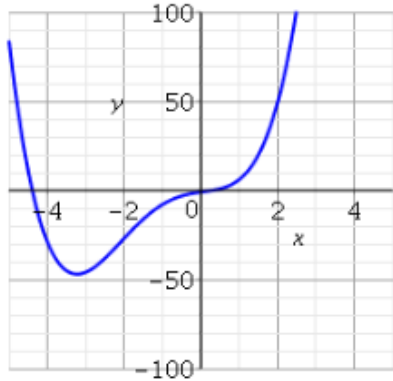
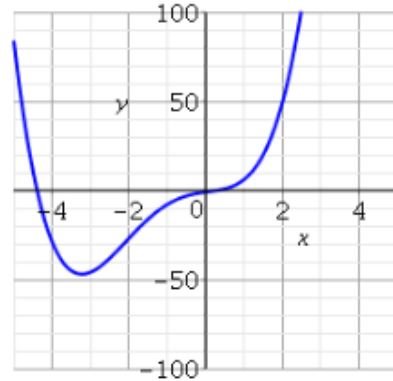
Figure 1.77: Feedback bij de Maple-graded vraag

In de figuur is te zien dat u bij de *Feedback* het goede antwoord ook kunt communiceren. Dit goede antwoord is in de rubriek *Algorithm* reeds voorbereid en MathML-gecodeerd ($\$displayantw$).

```
$displayantw=maple("printf(MathML[ExportPresentation]($antw))"); of
$displayantw=maple("printf(MathML:-ExportPresentation($antw))");
```

In de volgende figuur is de uitslag met de feedback te zien na klikken op *Grade*.

Grade Question

| Your response | Correct response |
|---|--|
| <p>Bereken de afgeleide van de functie</p> $f(x) = x^4 + 4x^3 - x^2 + 3x - 1.$ <p>De grafiek van deze polynoom is gegeven op het interval: $[-5, 5]$</p>  <p>Je kunt de grafiek van je antwoord ook plotten en vergelijken met de oorspronkelijke grafiek. Let goed op dat je bij het intikken van je antwoord de Maple-syntax gebruikt (dus een sterretje als je keer bedoelt).</p> $\frac{d}{dx} f(x) = 4x^3 + 12x^2 - 2x + 3 \quad (100\%)$ <p>Comment:</p> <p>Gebruik de machttregel.</p> $\frac{d}{dx} f(x) = 4x^3 + 12x^2 - 2x + 3$ | <p>Bereken de afgeleide van de functie</p> $f(x) = x^4 + 4x^3 - x^2 + 3x - 1.$ <p>De grafiek van deze polynoom is gegeven op het interval: $[-5, 5]$</p>  <p>Je kunt de grafiek van je antwoord ook plotten en vergelijken met de oorspronkelijke grafiek. Let goed op dat je bij het intikken van je antwoord de Maple-syntax gebruikt (dus een sterretje als je keer bedoelt).</p> $\frac{d}{dx} f(x) = 4x^3 + 12x^2 - 2x + 3$ |



Correct

Figure 1.78: Grading van een Maple-graded vraag met feedback

In bovenstaande figuur is de vraag weer te zien en daaronder precies wat de student heeft ingevoerd. De formule-instellingen stonden op *Text entry only* en dus is tekst te zien. Als u de instellingen op *Symbol entry only* had gezet, dan was dit antwoord in de tweedimensionale vorm weergegeven precies zoals de student het ook in de Editor had ingevoerd.

Wat er bij *Comment* staat, was in de rubriek *Feedback* van het eerste formulier van de vraag al voorbereid.

1.4 Tips voor de Grading Code

De *Maple-graded* vraag zoals besproken in paragraaf *Vraagtype Maple-graded* (page 45) is bij uitstek het vraagtype om formules te toetsen. Er zijn in dit vraagtype gigantisch veel mogelijkheden om de programmering voor de grading naar uw hand te zetten. Het meest eenvoudige om de respons van de student te checken is het volgende principe:

De formule die door de student is ingetikt, per definitie de variabele \$RESPONSE, wordt (eventueel na vereenvoudiging) afgetrokken van de correcte formule (eventueel in het *Algorithm* voorbereid) en het resultaat moet gelijk zijn aan 0.

Bij het *Maple-graded* vraagtype in de *Question Designer* is iets dergelijks standaard voorgeprogrammeerd en staat er in de *Grading Code* reeds ingevuld:

```
is ( ($ANSWER) - ($RESPONSE) = 0 ) ;
```

De variabele \$ANSWER is het resultaat van de formule die u als zijnde het correcte antwoord hebt ingevuld in de rubriek *Answer* (*Enter Maple code that evaluates to the correct answer:*)

Maar ook het volgende is mogelijk en komt vaak op hetzelfde neer:

```
evalb ( ($ANSWER) - ($RESPONSE) = 0 ) ;  
of  
evalb ( $RESPONSE = $ANSWER ) ;
```

De commando's *is* en *evalb* zijn boolean commando's met als resultaat *true* of *false*. Hiermee geeft het systeem uitsluitsel of het antwoord van de student als correct beoordeeld wordt of niet.

TIP: Let op dat *evalb* en *is* niet altijd hetzelfde werken zoals het onderstaande commando in maple laat zien:

```
> is ( 4*(2*x-5)*(2*x+5)=(4*x+10)*(4*x-10) ) ;  
  
true (1.1)
```

```
> evalb ( 4*(2*x-5)*(2*x+5)=(4*x+10)*(4*x-10) ) ;  
  
false (1.2)
```

TIP: U kunt ook iets programmeren dat als resultaat een getal oplevert tussen 0 en 1. Daarmee kan de grading nog wat genuanceerd worden. Het resultaat 0 is fout en het resultaat 1 is goed. Iets ertussenin geeft een gedeeltelijk goede beoordeling.

Zie ook paragraaf *Getallen met Maple-graded* (page 58) en ook in paragraaf *Ongeordende lijsten en geordende lijsten* (page 105) voor een voorbeeld daarvan.

Verder geldt dus dat standaard altijd \$RESPONSE het antwoord is dat de student heeft ingevuld en \$ANSWER het resultaat van de formule die u zelf hebt ingevuld in de rubriek *Answer* (*Enter Maple code that evaluates to the correct answer:*)

Deze rubriek *Answer* MOET beslist worden ingevuld als het gaat om een *Maple-graded*-vraagtype binnen de *Question Designer*.

Als het antwoord reeds in het *Algorithm* voorgeprogrammeerd is met \$antw of iets dergelijks, dan kunt u dat in die rubriek invullen. Voor de *Grading Code* is het eigenlijk verstandiger om ALTIJD gebruik te maken van het correcte antwoord dat u in het *Algorithm* geprogrammeerd hebt; dus als volgt:

```
evalb ( ($antw) - ($RESPONSE) = 0 ) ; of  
evalb ( ($antw) = ($RESPONSE) ) ;
```

TIP: Let ook op de haakjes voor het geval de respons met een minteken begint of gewoon altijd voor de veiligheid.

TIP: Soms is het nodig om het verschil tussen het antwoord van de student en het correcte antwoord eerst te laten vereenvoudigen en dan te checken of er nul uit komt. Dit doet men vaak als bijvoorbeeld de schrijfwijze van de student er niet toe doet. Vaak is dat een kwestie van uitproberen.

```
evalb ( simplify ( ($RESPONSE) - ($antw) ) = 0 ) ;
```

Maar er is meer mogelijk!! U kunt niet alleen de formule van de respons van de student (\$RESPONSE) beoordelen, maar zelfs de string van het ingetikte antwoord van de student bekijken ("RESPONSE"). Deze string bestaat uit een aantal karakters achter elkaar die door de student precies zo zijn ingetikt.

In de volgende paragrafen wordt een groot aantal mogelijkheden getoond en toegelicht voor het programmeren van de *Grading Code* in allerlei situaties, denk aan vergelijkingen, integralen, matrices en dergelijke waar het niet zo eenvoudig ligt met het matchen van het correcte antwoord met dat van de student door eenvoudig gelijkstellen of van elkaar aftrekken en eisen dat er nul uit komt.

De filosofie is eigenlijk algemeen dat u iets doet met het antwoord van de student, dat standaard \$RESPONSE is, en dan een match maakt met het correcte antwoord dat we in de volgende paragraaf steeds zullen aanduiden met \$antw of \$antwoord, omdat vaak in de rubriek *Algorithm* dit correcte antwoord al is aangemaakt en gecontroleerd.

TIP: Het antwoord dat u programmeert in de rubriek *Answer* is overigens niet altijd bruikbaar in de rubriek *Grading Code*, bijvoorbeeld als u dit antwoord bijvoorbeeld in 2D presentatie hebt voorbereid, waar geen formules, maar een Mathml-code gegenereerd wordt door Maple. Daarmee kan in de rubriek *Grading Code* natuurlijk niet gematcht worden. En zo zijn er nog wel wat meer voorbeelden waar de rubriek *Answer* even kritisch bekeken moet worden.

In geval van het matchen van verzamelingen hoeft de student de accolades niet in te tikken. Let op dat u bij het correcte antwoord de losse antwoorden formuleert met een komma ertussen en bij de grading code kunt u de verzameling die in het *Algorithm* voorbereid is gebruiken.

Bijvoorbeeld u wilt de verzameling \$verz={a,b,c}; overhoren. De student tikt in: a,b,c en dat is dus de \$RESPONSE. In de *Grading Code* matcht u dus de verzameling \$verz met {\$RESPONSE}. Let hierbij op dat u in de rubriek *Answer* niet de verzameling \$verz presenteert. De student zou bij het zien van het correcte antwoord kunnen denken dat hij accolades had moeten invoeren. Werk eventueel met operanden van de verzameling. (Zie paragraaf (page 69).)

1.4.1 Getallen overhoren met Maple-graded

Ook **getallen** kunt u overhoren met een Maple-graded vraagtype. Meestal gebruikt u voor getallen het numerieke vraagtype, maar soms wilt u bijvoorbeeld twee getallen goedrekenen. In dat geval is een Maple-graded vraag ook weer handig. In dit geval kunt u ook tolerantie aanbrengen in het antwoord. In de *Grading Code* programmeert u dan bijvoorbeeld het volgende:

```
evalb(abs(( $MVoptotA1)-($RESPONSE))<0.1)
or evalb(abs(( $MVoptotA2)-($RESPONSE))<0.1);
```

U kunt ook bijvoorbeeld het eerste antwoord helemaal goedrekenen en het tweede antwoord half goed rekenen met de volgende grading code:

```
if evalb(abs(( $MVoptotA1)-($RESPONSE))<0.1) then 1 elif evalb(abs(( $MVoptotA2)-($RESPONSE))<0.1) then 0.5 else 0
end if;
```

TIP: Let wel op de haakjes omdat hier \$RESPONSE een negatief getal kan zijn. Beter is het om altijd de haakjes eromheen te zetten.

U kunt ook een combinatie (lijstje met vaste volgorde) van bijvoorbeeld een letter met een getal vragen. (Het moment om een bepaald punt als het punt niet uitmaakt.)

```
evalb(A1=op(1,$RESPONSE) and
abs($MVoptotA1-(op(2,$RESPONSE)))<0.2 or
A2=op(1,$RESPONSE) and
abs($MVoptotA2-(op(2,$RESPONSE)))<0.2 );
```

1.4.2 Wanneer zijn formules gelijk aan elkaar?

Het steekt nogal nauw welke formules in de ogen van Maple als *aan elkaar gelijk* worden gezien.

In de volgende voorbeelden worden steeds twee formules als ongelijk beoordeeld terwijl deze twee formules afgetrokken en na vereenvoudiging wel nul opleveren.

In feite is $\text{evalb}((\$RESPONSE) = (\$antw))$; dus iets krachtiger dan $\text{evalb}((\$RESPONSE) - (\$antw) = 0)$; en dit is weer krachtiger dan

```
evalb(simplify(( $RESPONSE)-($antw))=0);
```

Deze laatste grading code is vaak bruikbaar als het helemaal niet uitmaakt in welke notatie de student het antwoord geeft.

Het is handig om een en ander uit te proberen met het computeralgebrasysteem Maple als u daar de beschikking over heeft of eventueel in het *Algorithm* het een en ander uit te proberen.

TIP: Hou er ook rekening mee dat in veel gevallen bij de *Grading Code* iets gedaan wordt met de \$RESPONSE (het antwoord van de student). Soms is het dan belangrijk dat deze \$RESPONSE in Maple syntax ingevoerd is!

Het herleiden van breuken is bijvoorbeeld moeilijk te checken bij een *Maple-graded*-vraag.

In paragraaf *Breuken* (page 73) wordt een verzameling tips geboden waarbij u uw voordeel kunt doen als het om de *Grading Code* gaat bij breuken.

TIP: Let altijd op of het mogelijk is dat de student ook gewoon de opgave kan overtuigen om een goede beoordeling te krijgen.

Of misschien klikt de student op *Preview* en krijgt dan een vereenvoudigde vorm te zien waarmee het antwoord in feite ook weggegeven wordt, zie *Figure 1.7 (page 6)*. Immers bij de instelling *Maple syntax Text entry only* wordt er vaak een vereenvoudiging aangeboden bij het klikken op *Preview*.

Er zijn echter trucs te over om het precies zo te krijgen als u het hebben wilt.

Hieronder een paar Maple-opdrachten om te zien wat het computeralgebrasysteem Maple eigenlijk doet met bijvoorbeeld breuken in allerlei vorm en hoe u zelf een en ander kunt uitproberen in het programma Maple.

```
> restart; breuk1:=a/x+b;
```

$$breuk1 := \frac{a}{x} + b \quad (1.3)$$

```
> breuk2:=simplify(breuk1);
```

$$breuk2 := \frac{bx + a}{x} \quad (1.4)$$

```
> evalb(simplify(breuk1-breuk2)=0);
```

$$true \quad (1.5)$$

Na vereenvoudiging is natuurlijk het verschil tussen de twee vormen gelijk aan 0. Echter, kijk eens naar de volgende twee statements:

```
> evalb(breuk1=breuk2);
```

$$false \quad (1.6)$$

```
> evalb(breuk1-breuk2=0);
```

$$false \quad (1.7)$$

De breuken worden als ongelijk beschouwd als er geen vereenvoudiging plaats vindt. Niet alleen bij getallen maar ook als b een veelvoud van a is, hebben we een probleem.

```
> breuk3:=a/x+3*a;
```

$$breuk3 := \frac{a}{x} + 3a \quad (1.8)$$

```
> breuk4:=simplify(breuk3);
```

$$breuk4 := \frac{a(3x + 1)}{x} \quad (1.9)$$

```
> breuk5:=(a+3*a*x)/x;
```

$$breuk5 := \frac{3ax + a}{x} \quad (1.10)$$

```
> evalb(breuk4=breuk5);
```

$$false \quad (1.11)$$

```
> evalb(breuk3=breuk4);
```

$$false \quad (1.12)$$

De twee breuken die wel hetzelfde zijn maar verschillen in schrijfwijze, worden niet altijd als hetzelfde gezien door Maple.

Dat geeft voordelen als we willen forceren dat de student de breuk vereenvoudigt of dat meer breuken samengenomen dienen te worden.

Let ook eens op de volgende uitdrukkingen die niet altijd als hetzelfde worden gezien.

```
> evalb(a*(x-5)=a*x-5*a);
```

false (1.13)

```
> evalb(a*(x-5)-(a*x-5*a)=0);
```

false (1.14)

```
> a*(x-5)-(a*x-5*a);
```

$a(x-5) - ax + 5a$ (1.15)

```
> simplify(a*(x-5)-(a*x-5*a));
```

0 (1.16)

Hier is te zien dat twee uitdrukkingen in verschillende gedaante niet altijd als hetzelfde worden gezien.

Ook het verschil van de twee uitdrukkingen is niet per definitie gelijk aan 0 als dat het na vereenvoudiging wel nul kan opleveren.

TIP: U kunt met `evalb(($antw) = ($RESPONSE));` twee uitdrukkingen, twee vergelijkingen, twee strings, twee lijsten of twee verzamelingen matchen waar dat niet zo gemakkelijk kan met het aftrekken van elkaar.

Kijk ook eens naar een Maple-opdracht hoe twee verzamelingen gematcht kunnen worden. Verzamelingen worden gedefinieerd met behulp van accolades.

```
> evalb({a,a,b,c,c,d}={c,b,a,d});
```

true (1.17)

Verzamelingen zijn gelijk als ze dezelfde elementen hebben. Dubbele elementen tellen in feite voor enkele en de volgorde doet er niet toe.

Ook kunnen **lijstjes**, met getallen of formules, waar de volgorde van belang is, met elkaar vergeleken worden. Lijstjes worden gedefinieerd met rechte haken.

```
> evalb([a,b,c*x-c*y]=[a,b,(c*(x-y))]);
```

false (1.18)

```
> is([a,b,c*x-c*y]=[a,b,(c*(x-y))]);
```

false (1.19)

```
> evalb([a,b,c*x-c*y]=[a,b,expand(c*(x-y))]);
```

true (1.20)

```
> is([a,b,c*x-c*y]=[a,b,expand(c*(x-y))]);
```

true (1.21)

Hierboven is te zien dat zelfs de formules in de lijstjes van dezelfde gedaante moeten zijn om ze als "hetzelfde" gewaardeerd te krijgen.

Met `is`, kunt u vaak hetzelfde als met `evalb`.

Strings kunt u heel eenvoudig matchen. Een string is een aantal karakters tussen dubbele quotes waarvan de volgorde vast ligt. Spaties tellen ook als karakter! (Bij formules worden de spaties rond een operator gewoon genegeerd en anders wordt er wel een syntax error in de *Preview* gemeld echter in de vertaling van de \$RESPONSE naar de string worden de spaties altijd wel meegenomen!)

```
> evalb("string1+s^2"="string1 +s^2");
```

$$\text{false} \quad (1.22)$$

```
> evalb("string1+s^2"="string1+s^2");
```

$$\text{true} \quad (1.23)$$

```
> StringTools[Remove](" ", "string1 +s^2");
```

$$\text{"string1+s^2"} \quad (1.24)$$

In het bovenstaande is duidelijk gemaakt dat het mogelijk is om bijvoorbeeld spaties uit een string te verwijderen.

Ook twee **vergelijkingen** kunnen gematcht worden als de vergelijkingen eerst voorbereid zijn (anders krijgt u meer dan één =-teken in de formule).

```
> verg1:=a*(-7+x+3*a-2*x^2)=8-p; verg2:=-a*(2*x^2-3*a+7-x)=-p+8;
  evalb(verg1 = verg2);
```

$$\text{verg1} := a(-2x^2 + 3a + x - 7) = 8 - p$$

$$\text{verg2} := -a(2x^2 - 3a - x + 7) = 8 - p$$

$$\text{false} \quad (1.25)$$

```
> verg1:=expand(a*(-7+x+3*a-2*x^2))=8-p; verg2:=expand(-a*(2*x^2-3*a+7-x))=-p+8;
  evalb(verg1 = verg2);
```

$$\text{verg1} := -2ax^2 + 3a^2 + ax - 7a = 8 - p$$

$$\text{verg2} := -2ax^2 + 3a^2 + ax - 7a = 8 - p$$

$$\text{true} \quad (1.26)$$

Wat er in feite gebeurt bij het matchen van twee vergelijkingen of deze hetzelfde zijn of niet, worden de linkerleden en de rechterleden met elkaar vergeleken. Ze moeten in dezelfde "vorm" staan (dus ontbonden of uitgewerkt of iets dergelijks), maar de volgorde is niet van belang. Het is dus zaak om een en ander wel uit te proberen.

TIP: Als u niet de beschikking hebt over Maple, kunt u eventueel in het *Algorithm* het een en ander uitproberen.

1.4.3 Werken met StringTools

De meeste formules in de vorm van uitdrukkingen kunnen gemakkelijk gematcht worden met de *Grading Code*.

Niet altijd ziet Maple de uitdrukkingen als gelijk, maar van elkaar aftrekken en vervolgens vereenvoudigen leidt altijd wel tot goede resultaten.

Dus met de *Grading Code* `evalb(simplify(($RESPONSE)-($antw))=0)`; is de beoordeling van een gewone formule dan heel flexibel wat de notatie van de student betreft. (Dit geldt niet voor matrices, verzamelingen en vergelijkingen of iets dergelijks.)

Soms is er extra programmering nodig als u een bepaalde vorm van een uitdrukking wilt afdwingen.

U kunt in feite op twee fronten naar de formule kijken die de student intikt.

1) U bekijkt de \$RESPONSE zoals in de twee vorige paragrafen is uitgelegd.

2) U bekijkt de string van de respons van de student en dat is "\$RESPONSE". Dus letterlijk wat de student heeft ingetikt.

U hebt daarvoor een aantal tools ter beschikking om bijvoorbeeld het aantal bepaalde karakters te tellen dat in de string "\$RESPONSE" voorkomt en daar iets mee te doen in de *Grading Code*. Maple bevat een pakket (StringTools) met commando's die

met strings kunnen werken. Een string is niets anders dan een aantal karakters achter elkaar en dat alles tussen dubbele quotes. Van de respons van de student kan dus heel gemakkelijk een string worden gemaakt door er gewoon quotes omheen te zetten. Dit gaat het mooist als u de instellingen voor het invoeren van de formules hebt gezet op *Maple Syntax en Text entry only*.

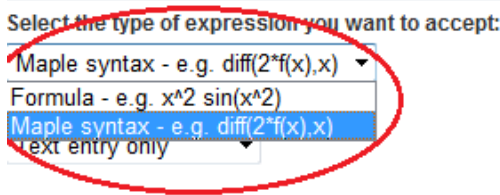


Figure 1.79: Instellingen met Maple Syntax

In paragraaf *De Analyse van de Response* (page 65) wordt een aantal tips gegeven om er achter te komen hoe de string van de respons van de student er precies uit ziet bij de verschillende instellingen zoals aangegeven in bovenstaande figuur. Er wordt dan gekeken naar hetgeen de student precies heeft ingetikt en hoe dat vertaald wordt naar een string. Een voorbeeld kan zijn dat het aantal keren dat de letter q in het ingetikte antwoord van de student geteld wordt. Dus u kunt checken met het boolean commando `evalb` of dat aantal bijvoorbeeld kleiner is dan 2 of niet (true of false).

```
evalb(StringTools[CountCharacterOccurrences]("$RESPONSE", "q") < 2);
```

Dit betekent: tel in de string van de respons van de student het aantal malen dat q voorkomt en controleer met `evalb` of dit kleiner is dan 2.

Ook kunt u "zoeken" naar een bepaald karakter in de string van het antwoord van de student met `Search` uit het `StringTools`-pakket.

```
evalb(StringTools[Search]("/", "$RESPONSE") = 0);
```

Dit betekent: zoek waar een slash in de string van de respons voorkomt. Het resultaat zal een getal zijn dat aangeeft op welke plaats de eerste de beste slash in de string verschijnt en als het resultaat nul is, dan komt er dus géén slash in de string voor.

TIP: Dit commando is dus alleen bruikbaar om te controleren of het karakter niet voorkomt of minstens één keer voorkomt door te eisen: `evalb(StringTools[Search]("/", "$RESPONSE") > 0);` Meer mogelijkheden zijn er niet.

Ook kan met dit commando gecontroleerd worden of bepaalde *opvolging van karakters* wellicht niet voorkomt in de string.

```
evalb(StringTools[Search]("factor", "$RESPONSE") = 0);
```

Dit betekent dus dat de opvolging van de karakters die het woord "factor" vormen wel of niet voorkomt in de string van de respons van de student (true of false). Het commando `Search` geeft als resultaat een getal aan als de karaktercombinatie wel voorkomt, namelijk op welke plaats het begin ervan is in de string is gevonden. Als het nul oplevert, komt de karaktercombinatie dus niet voor. (Eventueel kunt u testen dat het minstens één keer voorkomt door `>0` te eisen.)

TIP: Toch kunt u in zekere zin het aantal keren tellen dat een karaktercombinatie voorkomt binnen een string. Dat gaat als volgt: U zoekt in de string de bepaalde karaktercombinatie met `SearchAll`. Het resultaat is een rijtje getallen dat aangeeft op welke plaats de eerste letter van deze karaktercombinatie zich in de string bevindt.

Vervolgens **maakt u daar een verzameling van** door er accolades omheen te zetten en vraagt u naar het aantal operanden in deze verzameling met het commando `nops`. Zo weet u dus hoeveel keer deze karaktercombinatie in de string voorkomt.

Hieronder een paar regels in Maple om een en ander uit te proberen.

```
> StringTools[SearchAll]("pq", "a+p*b+pq+q^2-1/q+1/pq");
```

7, 20 (1.27)

```
> nops({StringTools[SearchAll]("pq", "a+p*b+pq+q^2-1/q+1/pq")});
```

2 (1.28)

Met het programmeren van de volgende regel in de *Grading Code* kunt u er dus voor zorgen dat bijvoorbeeld de combinatie x^2y^2 minder dan 2 maal voorkomt in de string van het antwoord van de student. Moet u natuurlijk nog wel checken of het antwoord in zijn totaliteit gelijk is aan het correcte antwoord.

```
evalb(nops({StringTools[SearchAll]("x^2*y^2", "$RESPONSE")})<2);
```

TIP: U kunt ook variabelen nemen in dit commando, ook als de variabele bijvoorbeeld bestaat uit meer karakters.

```
evalb(nops({StringTools[SearchAll]("$x", "$RESPONSE")})<2);
```

In het volgende voorbeeld zien we bij verschillende instellingen (*Text entry only* of *Symbol entry only*) dat er gevraagd wordt naar een vereenvoudiging van een formule.

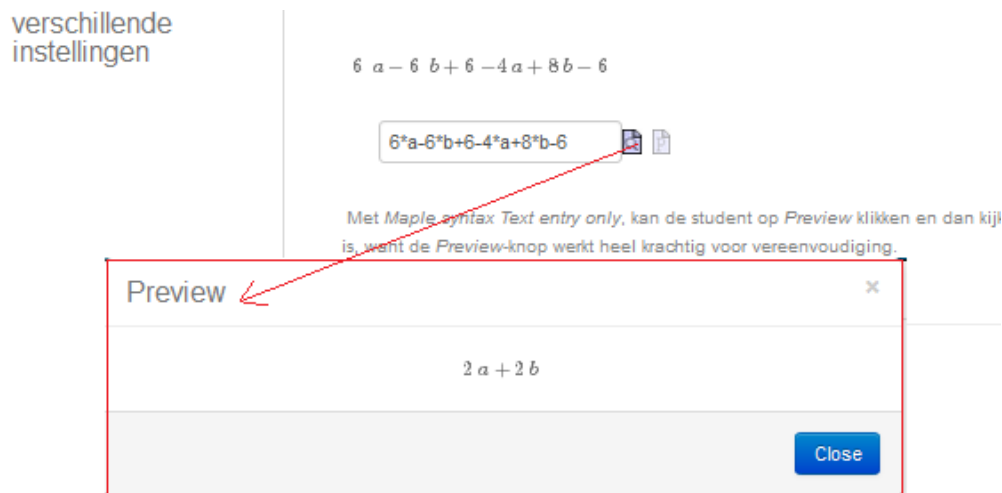


Figure 1.80: Vereenvoudigen

In bovenstaande opdracht wordt gevraagd om de gegeven formule te vereenvoudigen.

TIP: Gebruik in een dergelijk vereenvoudigingsgeval niet de instelling met *Text entry only* bij de instellingen van *Maple syntax*. Immers de *Preview*-functie bij *Maple syntax* is zó krachtig dat de vereenvoudiging wordt voorgezegd zoals in *Figure 1.80* (page 63) te zien is.

In een dergelijk geval zijn er twee mogelijkheden.

1) U gebruikt de *Custom Previewing Code* met daarin de volgende statement:

```
use InertForm:-NoSimpl in $RESPONSE: end: printf(InertForm:-ToMathML(%));
```

Zie paragraaf *Automatische vereenvoudiging bij Preview tegenhouden* (page 64).

2) U gebruikt de instelling *Symbol entry only*, zodat de student een Editor aangeboden krijgt waar géén *Preview*-knop beschikbaar is. De Editor is immers zelf de preview.

In beide gevallen kunt u controleren op het aantal specifieke karakters dat er in het antwoord van de student voorkomt. Zie ook paragraaf *De Analyse van de Response* (page 65).

In de programmering van de *Grading Code* van de vraag zoals te zien is in *Figure 1.80* (page 63) kunt u natuurlijk niet volstaan met het checken of het antwoord van de student afgetrokken van het correcte antwoord wel nul oplevert. Immers dan kan de student gewoon de vraag overtikken om een goede beoordeling te krijgen. Gelijksortige termen dienen hier samengenomen te worden. In de *Grading Code* kunt u verder geprogrammeren dat het aantal malen dat een letter(combinatie) voorkomt bijvoorbeeld kleiner moet zijn dan 2. Van het antwoord van de student wordt dan eerst een string gemaakt door er dubbele quotes omheen te zetten: "\$RESPONSE" en vervolgens kan het aantal karakters geteld worden met de opdracht `CountCharacterOccurrences`. Op deze manier kan de vorm waarin de student het antwoord moet geven, afgedwongen worden. Echter dit geldt alleen voor enkelvoudige karakters. De opdrachten *Search* of *SearchAll* komen hier beter van pas want de variabele \$x en \$z zijn afwisselend verschillende letters of lettercombinaties.

Hieronder ziet u de *Grading Code* voor de extra eis dat de lettercombinatie \$z minder dan 2 keer mag voorkomen en evenzo andere letters of lettercombinaties. (In het *Algorithm* was de variabele \$z bijvoorbeeld \$z=switch(rint(6),"a","b","x^2","yz","p","qr");)

```
evalb($RESPONSE=$ans2) and evalb(nops({StringTools[SearchAll]("$z", "$RESPONSE")})<2) and
evalb(nops({StringTools[SearchAll]("$x", "$RESPONSE")})<2);
```

TIP: Gebruik in deze gevallen liever niet de instelling *Formula*. Immers *Formula* kan geen lettercombinaties aan. Verder zal de *Preview* van de instelling met *Formula* er beslist niet goed uitzien.

Advies: gebruik liever niet de instelling *Formula* tenzij het niet anders kan. Zie een voorbeeld in paragraaf *Breuken* (page 73) in *Figure 1.93* (page 78).

TIP: Nog een uiterste middel is het controleren of de student precies het antwoord tikt zoals u dat verlangt. U gaat dan gewoon exact na of de student het voorgeschreven antwoord intikt. In het *Algorithm* bereid u dan het "stringantwoord" voor waarbij u controleert of dit de gewenste vorm heeft bijvoorbeeld met de algoritmische variabele:

```
$stringantwoord:=maple("convert($antw,string)");
```

In de *Grading Code* programmeert u dan het volgende:

```
evalb(convert($RESPONSE,string)=$stringantwoord);
```

Echter als u denkt dat de student misschien spaties zou kunnen tikken in zijn antwoord, programmeert u voor de veiligheid dan het volgende:

```
stringresponse:=StringTools[Remove](IsSpace,convert($RESPONSE,string));
evalb(stringresponse=$stringantwoord);
```

of

```
stringresponse := StringTools[Remove](IsSpace, "$RESPONSE");
evalb(StringTools[SubString](stringresponse, 1..-1)=$stringantwoord);
```

Hiermee veegt u dus eerst alle spaties uit het antwoord van de student en daarna controleert u of het resultaat ook precies zo is als u dat had voorbereid.

Het commando *Remove* uit het *StringTools* pakket kan ook andere karakters uit de string halen. En een spatie is ook een karakter dus het volgende had ook gekund:

```
StringTools[Remove](" ",convert($RESPONSE,string));
```

TIP: Kijk ook nog eens in Maple naar de Maple-commando's *searchtext* en *searchText* waarmee een string doorzocht kan worden respectievelijk case insensitief en case sensitief. Dit commando valt buiten het *StringTools*-pakket.

Echter aan boven behandelde commando's hebt u ruim voldoende.

Als u nog meer met strings wilt doen, kijk dan in de handleiding *Randomiseren* met daarin een speciale paragraaf over strings.

TIP: Stel dat u bij het vereenvoudigen van breuken het antwoord als één breuk wilt zien. Het juiste antwoord wordt door Maple gegenereerd, maar het kan zijn dat in de feedback de student als juiste antwoord ziet als $(6/5)/(x^2*y)$ terwijl u nu juist wilt hebben: $6/(5*x^2*y)$.

U kunt dan in de Maple Graded question bij het correcte antwoord de string invullen die u wilt dat de student intikt. Met *printf* krijgt u de quotes weer weg. Dus bijvoorbeeld in de rubriek *Answer* invullen: `printf("$a/($b*$L1^2*$L2)");`

```
TIP:***** evalb(searchtext("diff", "$RESPONSE")=0) and evalb(searchtext("sqrt",
"$RESPONSE")>0);
```

1.4.4 Automatische vereenvoudiging bij Preview tegenhouden

De automatische vereenvoudiging die de Preview laat zien bij het Maple-graded vraagtype kan voor een belangrijk deel opgelost worden door bij *Custom Previewing Code* daarin te voorzien. Maple heeft een nieuw pakket *InertForm* met daarin de mogelijkheid om de niet vereenvoudigde formule naar MathML weg te schrijven:

```
use InertForm:-NoSimpl in $RESPONSE: end: printf(InertForm:-ToMathML(%));
```

De response van de student wordt niet vereenvoudigd en het resultaat wordt door Maple omgezet in MathML.

Probeer u altijd wel even uit of het gewenste resultaat hiermee bereikt wordt en anders kunt u altijd nog overschakelen naar de *Symbol Entry* instelling waar geen *Preview* mogelijk is.

Edit Response Area

- List
- Maple-graded**
- Math App
- Mathematical formula
- Multiple Choice
- Numeric
- Sketch

Answer: 40/a
(referenced when grading as \$ANSWER)

Grading Code: `evalb(($ANSWER)-($RESPONSE)=0) and evalb(StringTools[CountCharacterOccurrences]('$RESPONSE','/'))<2);`

Expression Type: Maple Syntax - e.g. diff(2*f(x),x)

Text/Symbolic entry: Text entry only

Optional:

Maple Repository:
Maple Repository

Plotting Code:

Custom Previewing Code: `use InertForm:-NoSimpl in $RESPONSE: end: printf(Inf`

OK Cancel

Figure 1.81: Tegenhouden van automatische vereenvoudiging in de Preview door gebruikmaking van de Custom Previewing Code

Zie ook paragraaf *Automatische vereenvoudiging* (page 29) bij het voorbereiden van formules in het *Algorithm*.

1.4.5 Custom Previewing Code

Er zijn heel veel mogelijkheden om in de rubriek Custom Previewing Code nog iets te programmeren als de student bepaalde vormen niet mag intikken, bijvoorbeeld twee vectoren (in de vorm van twee lijsten). Als u wilt dat de student daarop gecontroleerd wordt kunt u bijvoorbeeld ingeven:

```
if evalb(StringTools[CountCharacterOccurrences]('$RESPONSE',"["<2) then printf(MathML[ExportPresentation]
(Vector($RESPONSE))) else printf("tik één vector") end if
```

Zie ook in de paragraaf hiervoor dat automatische vereenvoudiging ook tegengehouden kan worden met iets te programmeren in deze rubriek.

1.4.6 De Analyse van de Response

In verschillende situaties wilt u misschien weten hoe de string van de response van de student er uit ziet met het oog op het gebruik van StringTools om eventuele notaties bij de student af te dwingen.

In de paragraaf *Werken met StringTools* (page 61) wordt een aantal voorbeelden gegeven van situaties waarbij het aftasten van het antwoord van de student zeker nut heeft. In de *Grading Code* kunt u daar dan gebruik van maken.

Alleen het *Maple-graded* (ook binnen de *Question Designer*) heeft de mogelijkheid van een *Grading Code*. Andere vraagtypen beslist niet.

Als u iets programmeert in deze *Grading Code* is het van belang dat u niet alleen anticipeert op wat de student mogelijk kan invoeren, maar ook hoe u de instellingen voor het antwoord hebt staan.

Bij de drie verschillende instellingen kan de gedaante van de string van de response van de student ("{\$RESPONSE}") anders zijn.

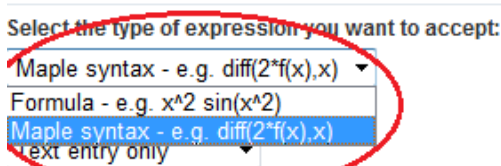


Figure 1.82: Instellingen voor het antwoord

De drie mogelijke instellingen zijn:

- 1) *Maple syntax Text entry only*
- 2) *Maple syntax Symbol entry only*
- 3) *Formula*

Deze laatste mogelijkheid met *Formula* gebruiken we liever niet voor formules, tenzij het niet anders kan.

De eerste instelling met *Maple syntax Text entry only* is het gemakkelijkste, want in de string "{\$RESPONSE}" komt gewoon precies dat te staan wat de student letterlijk heeft ingetikt. Aantal haakjes sterren en breukstrepen enzovoort wordt letterlijk in de string meegenomen. Echter soms is deze instelling onwenselijk bijvoorbeeld als het bezwaarlijk is dat de student gebruikmaakt van de automatische vereenvoudiging die bij *Preview* zichtbaar wordt. Maar daarin kan wat de meeste gevallen betreft in voorzien worden zoals behandeld in paragraaf *Automatische vereenvoudiging tegenhouden* (page 64). Maar soms moet u daardoor dus uitwijken naar een andere instelling. Ook kan met deze instelling de student gebruikmaken van Maple-commando's. Maar dat kunt u voorkomen door de string-response daarop te checken met behulp van *StringTools*. Zie paragraaf *Werken met StringTools* (page 61).

We gaan nu kijken hoe u er achter kunt komen wat de gedaante is van de "{\$RESPONSE}".

Dat is heel eenvoudig als u in een testvraagstukje van het type *Maple-graded* met verschillende formule-instellingen in het vak *Correct answer*, bedoeld voor het correcte antwoord, nu "{\$RESPONSE}" invult in plaats van het echte correcte antwoord (zie *Figure 1.83* (page 66)).

In de feedback na grading van de fout beantwoorde vraag ziet u precies hoe hetgeen er ingetikt is, vertaald wordt naar een string.

Edit Response Area

Figure 1.83: Genereren van de string van de RESPONSE

Als u dit testvraagstukje gemaakt hebt en u vult gewoon een fout antwoord in, dan komt er in de feedback letterlijk te staan wat u had ingetikt (*Your response*) en als *Correct response* komt er te staan wat voor **string** het systeem van uw ingetikte antwoord maakt. Dit is in het geval van de instelling met *Text entry only* natuurlijk geen probleem, want u zag toch wel wat er als antwoord was ingetikt, maar in het geval u de instellingen had gedaan met *Symbolic entry only*, is het een ander verhaal.

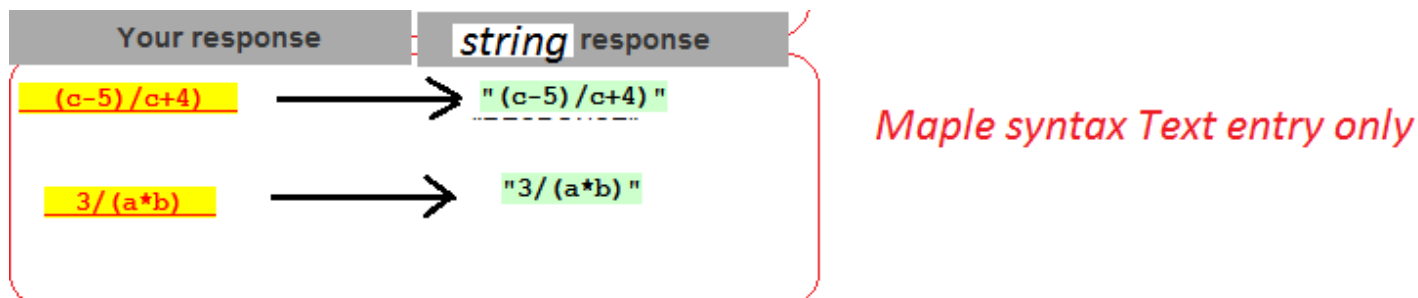


Figure 1.84: Het zichtbaar maken van de "\$RESPONSE" met Text entry only

In bovenstaande figuur zijn de instellingen met *Text entry only* gemakkelijk, want letterlijk komt in de string van de response altijd te staan wat de student heeft ingetikt.

Bij de formule-instelling van deze vraag met *Symbol entry only* zijn we overgeleverd aan de manier waarop de 2-dimensionale formules, door de student ingetikt in de Editor, omgezet wordt naar een string, zoals in de volgende figuur te zien is.

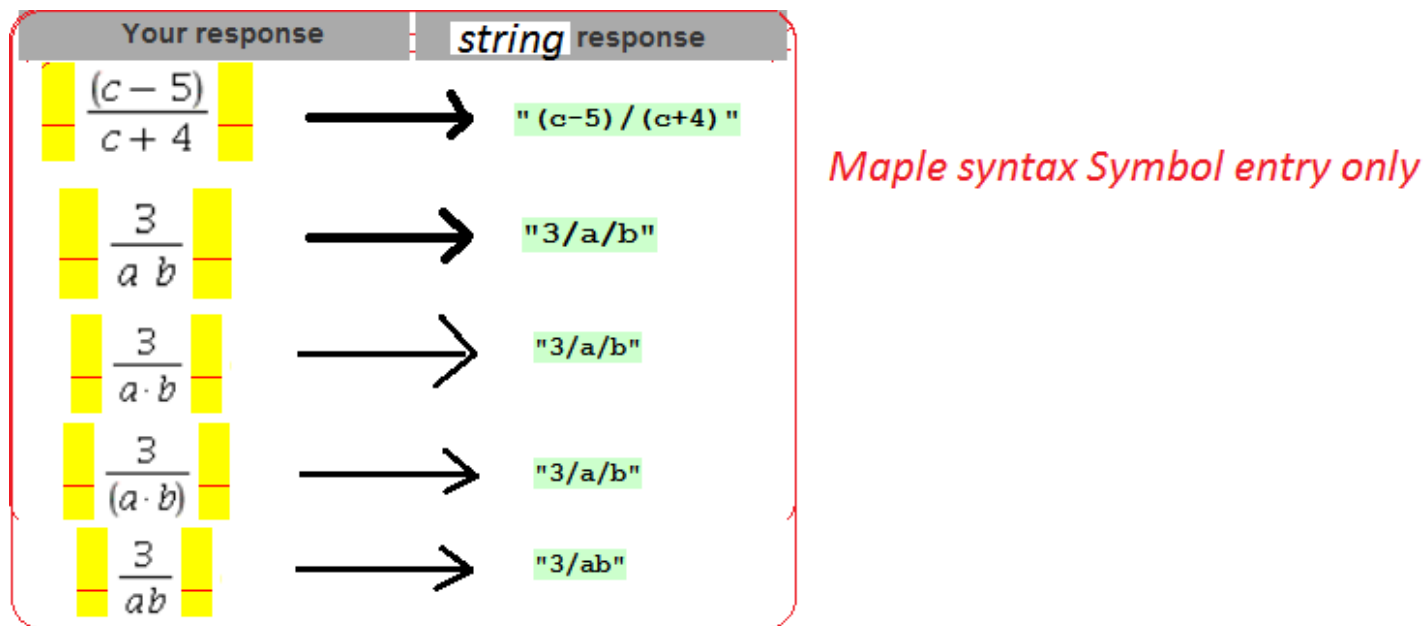


Figure 1.85: Het zichtbaar maken van de "\$RESPONSE" met Symbol entry only

In bovenstaande figuur is te zien dat het niet uitmaakt of u in de Editor haakjes om de teller of noemer zet of niet, de haakjes komen toch wel in de string tevoorschijn omdat dit zuivere, niet mis te verstane Maple syntax is. Ook is te zien dat als u een spatie tikt in de Editor, dit op de juiste manier vertaald wordt naar een vermenigvuldiging. Als de student bijvoorbeeld tussen twee karakters géén spatie tikt, dan wordt dit keurig als één geheel opgevat.

Nog meer voorbeelden met de formule-instelling *Symbol entry only*, op welke manier de ingetikte formule vertaald wordt naar een string is hier onder te zien:

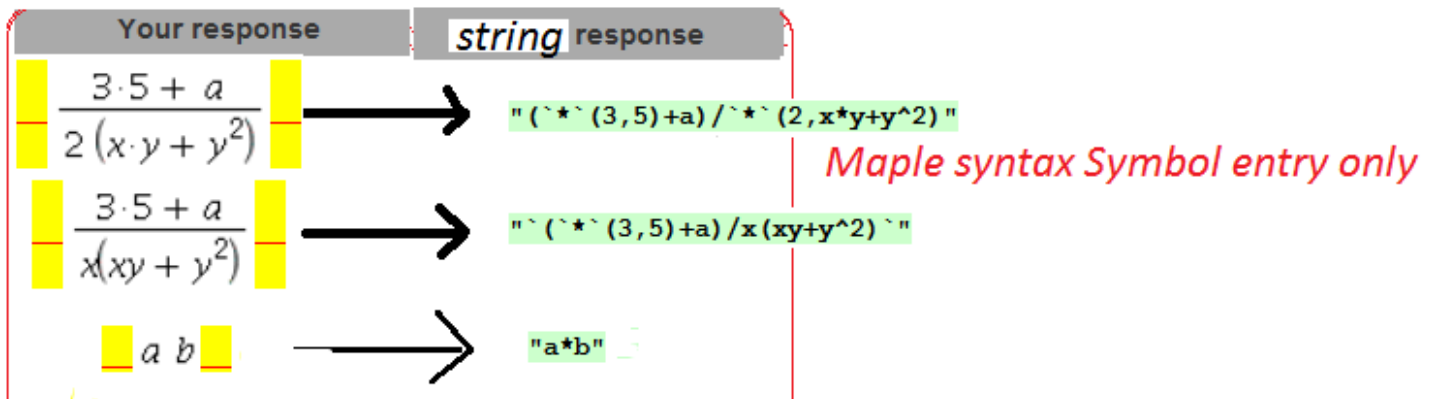


Figure 1.86: Het zichtbaar maken van de "\$RESPONSE" met Symbol entry only

In bovenstaande figuur is te zien hoe de vermenigvuldiging met behulp van een ster gedefinieerd wordt, maar vaak op verschillende manieren. Let ook eens op als u vlak voor het begin van de haakjes geen spatie tikt, dat er dan sprake is van een functievoorschrift vergelijkbaar met $f(x)$.

Het is heel goed mogelijk om in de *Grading Code* met StringTools te werken, maar dan wel met voorbedachte rade, want het aantal haakjes en slashes en dergelijke zal niet overeenkomen met wat u misschien in gedachten had. Echter de vertaling naar Maple syntax gebeurt op eenduidige wijze en er kan bijna geen misverstand ontstaan als de student er rekening mee houdt dat een spatie tussen twee karakters vertaald wordt naar een vermenigvuldiging.

Ten slotte laten we nog zien hoe de conversie is als de formule-instelling op *Formula* staat. Deze zogenaamde "gebruiksvriendelijke" instelling is in feite bijzonder gebruiksvriendelijk, want hij is voor meerdere uitleg vatbaar! Zo worden lettercombinaties altijd gezien als vermenigvuldigingen en ook als zodanig naar een string vertaald. Het is maar dat u het weet.

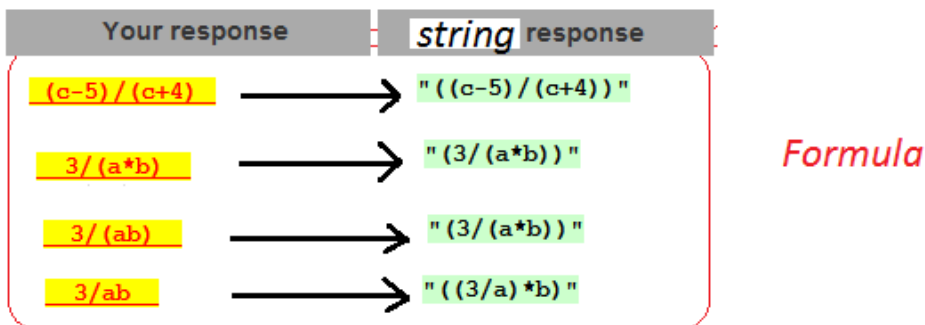


Figure 1.87: Het zichtbaar maken van de "\$RESPONSE" met Formula

Ook valt op dat er om de formule heen een paar extra haakjes gezet worden. Bij deze instelling is het toch mogelijk om in de *Grading Code* met StringTools te werken, maar ook hier geldt dat u dat met voorbedachte rade moet doen. Het tellen van het aantal haakjes zal dus lastig worden. Bij twijfel kunt u altijd een dergelijke test doen zoals boven besproken in figuur *Figure 1.83* (page 66) door in het *Answer*-veld "\$RESPONSE" te tikken en dan eventueel iets uit te proberen.

Nog meer voorbeelden van de analyse van de respons zijn te vinden in paragraaf *Wortelvormen* (page 85) in *Figure 1.100* (page 85) en *Figure 1.101* (page 86).

Voorbeelden van de analyse van de respons van Matrices en Vectoren zijn te vinden in paragraaf *Matrices en Vectoren* (page 124).

Voorbeelden van de analyse van de respons van differentialen en integralen zijn te vinden in paragraaf *Integralen* (page 117).

1.4.7 Coëfficiënten en operanden

Het is handig als u wat Maple-commando's kent om te werken met afzonderlijke operanden, coëfficiënten en eigenschappen van een formule.

Een test om te kijken of een polynoom van een bepaalde **graad** is, bijvoorbeeld van de vierde graad in x .

```
evalb(degree($RESPONSE,x)=4);
```

Op de volgende manier kunt u ook checken of de graad van de respons van de student dezelfde is als de graad van het correcte antwoord \$antw.

```
evalb(degree($RESPONSE,x)=degree($antw,x));
```

In onderstaande regels in Maple is te zien hoe een reeksontwikkeling van een functie wordt opgevraagd en geconverteerd wordt naar een polynoom. Van een vijfdegraads polynoom worden nu de **coëfficiënten** opgevraagd. De coëfficiënt van x^3 kan gemakkelijk opgevraagd worden met `coeff(f,x^3)`.

U kunt met het commando `coeffs` alle coëfficiënten van de polynoom opgevragen. Het is vervolgens mogelijk om van de gegenereerde rij een verzameling te maken (door er accolades omheen te zetten) en eventueel op alle elementen van die verzameling een functie los te laten met het commando `map`.

Het commando `map` kan namelijk in één keer iets doen met alle elementen van een matrix of verzameling of iets dergelijks. In dit geval zijn alle elementen van de verzameling met 100 vermenigvuldigd en vervolgens afgerond met `round`. Zo kunt u in feite alle coëfficiënten van een polynoom bekijken op 2 decimalen nauwkeurig.

```
> f:=evalf(convert(series(sin(3*x),x=0),polynom));
```

$$f := 3 \cdot x - 4.500000000x^3 + 2.025000000x^5 \quad (1.29)$$

```
> coeff(f,x^3);
```

$$-4.500000000 \quad (1.30)$$

```
> coeffs(f);
```

$$3., -4.500000000, 2.025000000 \quad (1.31)$$

```
> map(x->round(100*x),{coeffs(f)});
```

$$\{-450, 203, 300\} \quad (1.32)$$

```
> round~(100*~{coeffs(f)});
```

$$\{-450, 203, 300\} \quad (1.33)$$

```
> f1:=evalf(-6*sin(-8)*(x+3)+3*cos(-8));
```

$$f1 := 5.936149480x + 17.37194834 \quad (1.34)$$

```
> coeffs(-6*sin(-8)*(x+3)+3*cos(-8));
```

```
Error, invalid arguments to coeffs
```

```
> coeffs(evalf(-6*sin(-8)*(x+3)+3*cos(-8)));
```

$$17.37194834, 5.936149480 \quad (1.35)$$

Op bovenstaande manier kunt u ook met de operator gevolgd door de postfix in de vorm van een tilde (~) alle elementen van bijvoorbeeld een verzameling behandelen. (Elementwise operator.)

Let op dat u eventueel op numerieke getallen overgaat als de coëfficiënten niet gemakkelijk te bepalen zijn met het commando `coeffs`. Of eerst de polynoom schrijven zonder haakjes (met `expand`) anders gaat het opvragen van de coëfficiënten niet.

Met deze commando's kunt u gemakkelijk een polynoom waar de coëfficiënten op bijvoorbeeld 2 decimalen nauwkeurig zijn afgerond, matchen met het correcte antwoord waarbij u daarvan ook de coëfficiënten afrondt.

Zie het volgende voorbeeld waar de reeksontwikkeling van een functie gevraagd wordt met drie termen.

20 sinus reeks numeriek

Geef de reeksontwikkeling van $f(x) = \cos(4x)$ in de buurt van $x = 0$ en geef 3 termen.
(eventueel afronden op twee decimalen).

Klik eventueel op **Plot** om te zien hoe jouw reeksontwikkeling de functie benadert in de buurt van $x = 0$



Preview

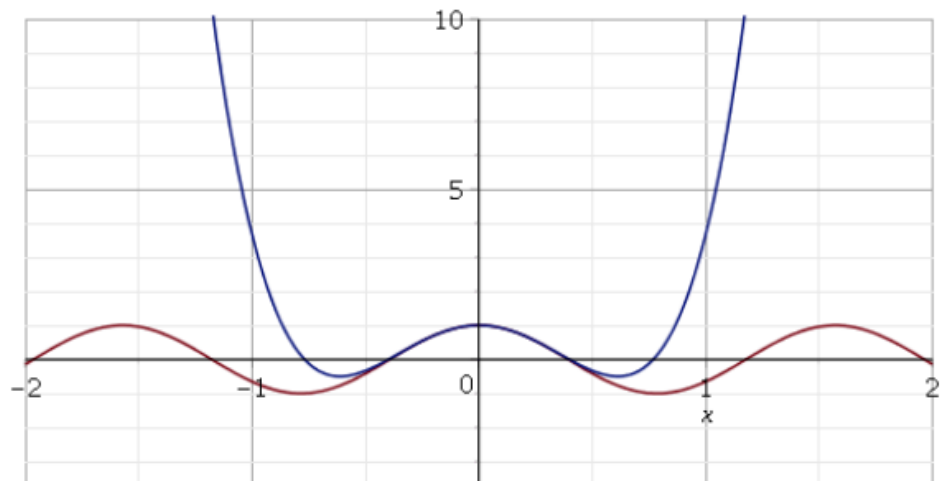


Figure 1.88: Een polynoom met decimalen matchen

In bovenstaande figuur is te zien dat er ook een mogelijkheid geboden wordt aan de student om zijn eigen antwoord in een grafiek te visualiseren! In een Maple-graded zelfstandige vraag hebt u dan ook iets meer mogelijkheden om de *Plotting Code* te programmeren, zodat de student op "Plot" kan klikken en de grafiek van niet alleen zijn eigen antwoord maar ook dat tesamen met de oorspronkelijke grafiek kan bekijken. De *Plotting Code* is: `plot([$functie, $RESPONSE], x=-2..2, -10..10);`

Zie *Figure 1.69* (page 48) waar u die *Plotting Code* moet invullen.

Het goede antwoord \$reeks is in het *Algorithm* voorbereid met behulp van het Maple-commando: `series`.

Besteed dit soort dingen zoveel mogelijk uit aan Maple voor het efficiënt en foutloos werken.

```
$a:=range(2,4);
$f:=maple("sin($a*x)");
$g:=maple("cos($a*x)");
$functie=switch(rint(2),"$f","$g");
$displayfunctie=maple("printf(MathML[ExportPresentation](f(x)=$functie))");
$reeks=maple("convert(series($functie,x=0),polynom)");
$displayreeks=maple("printf(MathML[ExportPresentation]($reeks))");
$p=plotmaple("plot([$functie,$reeks],x=-2..2,-10..10,color=[red,blue],thickness=2,legend=[functie,reeks]),plotoptions='height=350, width=250'");
$x0=maple("printf(MathML[ExportPresentation](x=0))");
```

In het *Algorithm* is verder te zien dat afwisselend de functie een sinus of een cosinus is.

Verder wordt ten behoeve van de feedback een grafiek $\$p$ voorbereid en ook de formule van het goede antwoord $\$displayreeks$.

... en in de *Grading Code* programmeren we het volgende:

```
evalb(degree($RESPONSE,x)=degree($reeks,x))
and evalb(StringTools[Search]("series",$RESPONSE)=0)
and evalb(map(x->round(100*x),{coeffs($reeks)})=map(x->round(100*x),{coeffs($RESPONSE)}))
and evalb(nops([coeffs($reeks)])=nops([coeffs($RESPONSE)])) ;
```

Hiermee bereiken we dat eerst gecheckt wordt of de polynoom die de student heeft ingetikt wel van de juiste graad is. Vervolgens wordt er ook nog gecheckt of de student niet het Maple-commando `series` gebruikt om het goede antwoord te geven. En ten slotte wordt de *verzameling* bestaande uit alle coëfficiënten van de polynoom van de student vergeleken met dezelfde verzameling, maar dan van het correcte antwoord $\$reeks$. Deze coëfficiënten worden wel eerst met 100 vermenigvuldigd en vervolgens afgerond, omdat het om een nauwkeurigheid van 2 decimalen gaat. Als extra wordt er nog gecheckt of het aantal coëfficiënten van de correcte reeks overeenkomt met het aantal coëfficiënten van de reeks die de student invoert. Immers bij het gelijkstellen van verzamelingen tellen dubbele elementen niet mee.

U kunt natuurlijk ook alle coëfficiënten stuk voor stuk controleren.

Zie ook paragraaf *Verzamelingen* (page 72) voor meer informatie over verzamelingen.

De **operanden** van een uitdrukking kunt u tevoorschijn roepen met het commando `op`. Er komt dan een rij uitdrukkingen te staan met achtereenvolgens alle operanden van de uitdrukking. Soms kan het handig zijn om eerst de uitdrukking met sort op volgorde te zetten, zodat het rijtje operanden eenduidig is. Zo kunt u ook een enkele operand uit de uitdrukking halen. Eventueel ook weer operanden van operanden (met behulp van een lijstje): bijvoorbeeld van de tweede operand de eerste nemen.

Het aantal operanden van een uitdrukking f kunt u ook tevoorschijn roepen met `nops(f)`.

```
> f:=evalf(convert(series(sin(3*x),x=0),polynom));
```

$$f := 3 \cdot x - 4.500000000x^3 + 2.025000000x^5 \quad (1.36)$$

```
> op(sort(f));
```

$$2.025000000x^5, -4.500000000x^3, 3 \cdot x \quad (1.37)$$

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

$$2.025000000x^5 \quad (1.38)$$

```
> op([2,1],f);
```

$$-4.500000000 \quad (1.39)$$

```
> nops(f);
```

$$3 \quad (1.40)$$

Een voorbeeld van het gebruik van operanden in uitdrukkingen is ook te vinden in figuur *Figure 1.95* (page 80).

Nog meer over operanden staat vermeld in paragraaf *Wortelvormen* (page 90).

TIP: Eventueel met coëfficiënten werken voor een kwadratische uitdrukking:

```
evalb(degree($RESPONSE,x)=degree($reeks,x)) and
evalb(abs(coeff($RESPONSE,x)-(coeff($reeks,x))<abs(coeff($reeks,x))*0.02) and evalb(abs(coeff($RESPONSE,x^2)-
(coeff($reeks,x^2))<abs(coeff($reeks,x^2))*0.02) and
evalb(abs(op(3,sort(expand($RESPONSE)))-
(op(3,sort($reeks))))
<abs(op(3,sort($reeks)))*0.02);
```

1.4.8 Verzamelingen

Als u een rijtje getallen of formules wilt overhoren waarbij de volgorde niet belangrijk is, is het handig om verzamelingen te matchen. U kunt hierbij denken aan de oplossingsverzameling van vergelijkingen of iets dergelijks. Zie paragraaf (page 99) voor een uitgebreid voorbeeld daarvan. De student tikt dan de verschillende antwoorden in, gescheiden door komma's (\$RESPONSE) en vervolgens maakt u daarvan een verzameling en matcht u dit {\$RESPONSE} met de oplossingsverzameling die u eerst in het *Algorithm* hebt voorbereid.

In geval van het matchen van verzamelingen hoeft de student de accolades niet in te tikken. Let op dat u bij het correcte antwoord de losse antwoorden formuleert met een komma ertussen en bij de grading code kunt u de verzameling die in het *Algorithm* voorbereid is gebruiken.

Bijvoorbeeld u wilt de verzameling \$verz={a,b,c}; overhoren. De student tikt in: a,b,c en dat is dus de \$RESPONSE. In de *Grading Code* matcht u dus de verzameling \$verz met {\$RESPONSE}. Let hierbij op dat u in de rubriek *Answer* niet de verzameling \$verz presenteert. De student zou bij het zien van het correcte antwoord kunnen denken dat hij accolades had moeten invoeren. Werk eventueel met operanden van de verzameling. (Zie paragraaf (page 69).)

TIP: Eventueel kan dat ook met andere vraagtypen. Kijk ook eens bij het vraagtype *Formula (Mathematical Formula)* (page 134) waar het ook mogelijk is om ongeordende lijsten te overhoren.

Bij verzamelingen worden dubbele elementen als enkele gezien. Echter bij het matchen van verzamelingen moet u er rekening mee houden dat deze altijd exacte elementen bevatten en geen decimale getallen gecombineerd met gehele getallen. De getallen 3.0 en 3. en 3 bijvoorbeeld, worden in de context van een verzameling niet als dezelfde gezien. De student geeft bijvoorbeeld een afgerond geheel getal en Maple ziet dit als niet hetzelfde als het een element van een verzameling is! zoals de onderstaande commandoregels in Maple laten zien.

```
> A:={3.0,5.23,1/3};B:={3,5.23,1/3};
```

$$A := \left\{ \frac{1}{3}, 3.0, 5.23 \right\}$$

$$B := \left\{ 3, \frac{1}{3}, 5.23 \right\} \quad (1.41)$$

```
> evalb(A=B);
```

$$false \quad (1.42)$$

```
> convert(A,rational);
```

$$\left\{ 3, \frac{1}{3}, \frac{523}{100} \right\} \quad (1.43)$$

```
> evalb(convert(A,rational)=convert(B,rational));
```

$$true \quad (1.44)$$

Ook is het eventueel handig om de getallen van de verzameling eerst met 100 te vermenigvuldigen en vervolgens af te ronden:

```
> map(x->round(100*x),A);
```

$$\{33, 300, 523\} \quad (1.45)$$

Eventueel met de elementwise operator:

```
> round~(100*~A);
```

$$\{33, 300, 523\} \quad (1.46)$$

```
> evalb(map(x->round(100*x),A)=map(x->round(100*x),B));
```

$$true \quad (1.47)$$

Of met de elementwise operator:

```
> evalb(round~(100*~A)=round~(100*~B));
```

true

(1.48)

Met de eerste truc, door de decimale getallen naar echte breuken om te zetten met `convert(...,rational)`, gaat het matchen van verzamelingen wel goed. Dit commando werkt op uitdrukkingen, verzamelingen, vergelijkingen, ongelijkheden etcetera en behandelt in één keer alle onderdelen van de uitdrukking, vergelijking of verzameling. Een voorbeeld hiervan vindt u in het voorbeeld van *Figure 1.129 (page 110)* in paragraaf *Ongelijkheden*.

Een andere mogelijkheid is om eerst alle getallen van de verzameling bijvoorbeeld 100 keer zo groot te maken en daarna af te ronden met `round`. Als u dus een nauwkeurigheid van 2 decimalen eist is dit ook een aardige manier van doen om decimalen kwijt te raken. Echter dan moet u met het commando `map` werken of met de elementwise operator (`~`) om alle getallen van de verzameling afzonderlijk te behandelen.

Ook handig is het werken met doorsnede (`intersect`) en vereniging (`union`) van verzamelingen.

Wilt u bijvoorbeeld een getal `$D1` checken op het aantal verschillende factoren dat dit getal bevat? Met het volgende commando kunt u checken of de doorsnede van de verzameling factoren van het correcte antwoord en de verzameling van de factoren van het gegeven antwoord dezelfde is als die van het gegeven antwoord.

```
evalb(numtheory[factorset]($D1) intersect numtheory[factorset]
($RESPONSE)=numtheory[factorset]($RESPONSE));
```

Het betekent dus dat de verzameling factoren van het gegeven antwoord geheel binnen de verzameling factoren van het correcte antwoord valt.

```
> numtheory[factorset](258);
```

{2, 3, 43}

(1.49)

```
> numtheory[divisors](258);
```

{1, 2, 3, 6, 43, 86, 129, 258}

(1.50)

Meer voorbeelden van het gebruik van **verzamelingen** in de *Grading Code* vindt u ongemerkt heel veel in bijvoorbeeld:

Figure 1.80 (page 63)

Figure 1.88 (page 70)

Figure 1.95 (page 80)

Figure 1.97 (page 82)

Figure 1.117 (page 98)

Figure 1.119 (page 100)

enzovoort.

$\text{abs}(2x-1) = x^2$

Je kunt aan elk van de drie oplossingen een eigen 'gewicht' toekennen met bijvoorbeeld de volgende grading-code:

```
`if'(member(-1-sqrt(2),{$RESPONSE}),0.4,0)+`if'(member(-1+sqrt(2),{$RESPONSE}),0.4,0)+`if'(member(1,
{$RESPONSE}),0.2,0)
```

1.4.9 Breuken

Bekijk eerst nog eens paragraaf *Wanneer zijn formules gelijk aan elkaar?* (page 58) waar uitgelegd wordt dat breuken verschillend genoteerd, niet altijd door Maple als hetzelfde gezien worden.

TIP: Let altijd op of het mogelijk is dat de student ook gewoon de opgave kan overtuigen om een goede beoordeling te krijgen.

Of misschien klikt de student op *Preview* en krijgt dan een vereenvoudigde vorm te zien waarmee het antwoord in feite weggegeven wordt, zie *Figure 1.80 (page 63)*. Immers bij de instelling *Maple syntax Text entry only* geeft de *Preview* vaak een vereenvoudiging te zien.

Er zijn echter trucs te over om het precies zo te krijgen als u het hebben wilt.

Bekijk ook nog even de paragraaf *Werken met StringTools (page 61)* om een bepaalde schrijfwijze in het antwoord van de student af te dwingen waarbij het belangrijk is te weten welke instellingen u hanteert voor het invullen van de formule.

Hieronder de verschillende mogelijkheden bij de verschillende instellingen (*Figure 1.82 (page 66)*).

1) Bij de instelling **Maple syntax met Text entry only** moet u zich er bewust van zijn dat de *Preview*-functie de ingevoerde formule vaak vereenvoudigt. Bovendien kan de student bij deze instelling ook Maple-commando's gebruiken. Het werken met StringTools heeft hier extra aandacht nodig. Voordeel van deze instelling is wel dat de string van de respons precies de invoer van de student overneemt. U kunt dan bijvoorbeeld controleren of de student één breuk invoert met het tellen van het aantal slashes in het antwoord als dat nodig is, of u kunt checken of de student geen Maple-commando's gebruikt of iets dergelijks.

In het voorbeeld hieronder zien we dat de *Preview* niet de bedoelde vereenvoudiging levert, dus we kunnen gerust met deze instelling werken.

Question Name: WB 2 herleiden 01

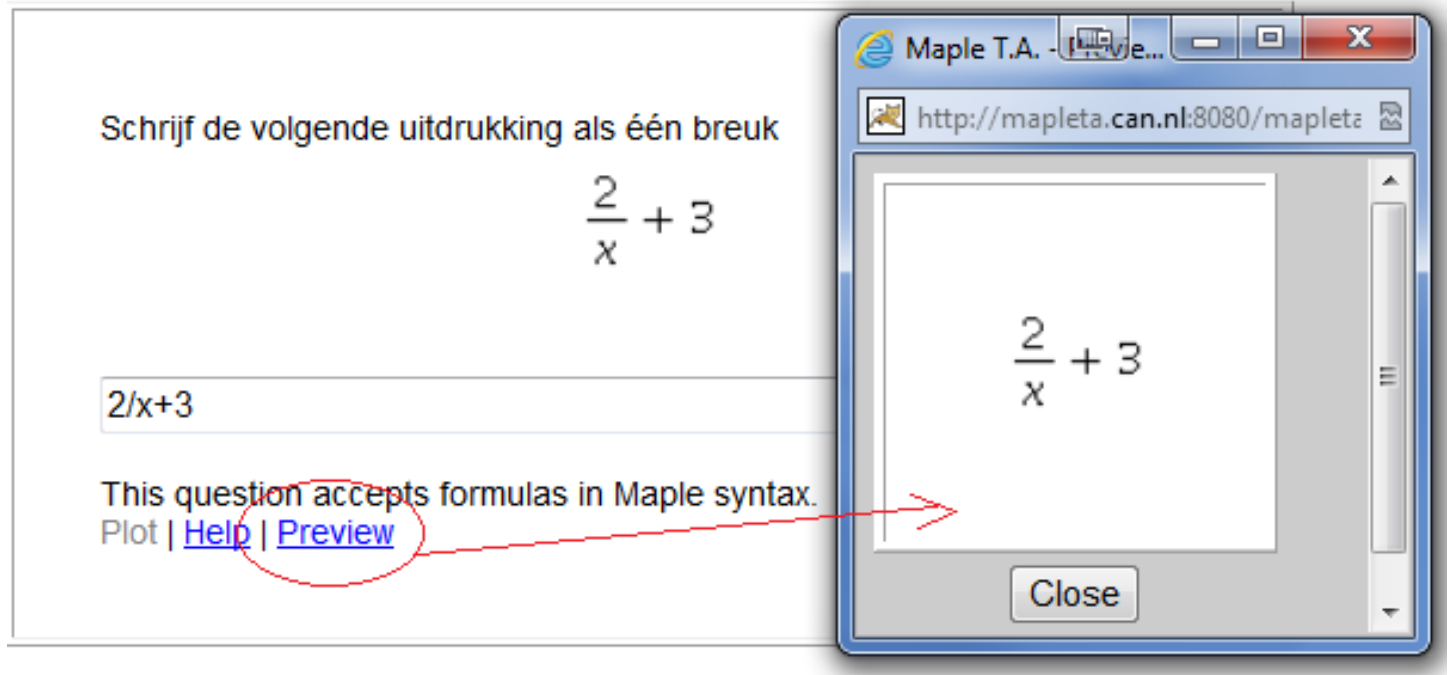


Figure 1.89: Breuken met instelling Text entry only

De *Grading Code* is:

```
evalb($RESPONSE=$antw) and (StringTools[Search]("simplify",$RESPONSE)=0) and
(StringTools[Search]("normal",$RESPONSE)=0);
```

In dit geval hebben we niet gekozen voor `evalb(simplify($RESPONSE-($antw))=0)`; want dat was té robuust geweest. Het maakt in dit geval juist wél uit in welke vorm het antwoord door de student ingevoerd wordt. Ook is in de *Grading Code* geprogrammeerd dat de Maple commando's `normal` en `simplify` niet in het antwoord van de student mogen voorkomen. Het correcte antwoord is reeds in het *Algorithm* geprogrammeerd als `$antw=maple("simplify($a/x+$b)");` en alleen in die vorm wordt het goedgekeurd.

Uiteindelijk zal het antwoord van de student zoals afgebeeld in *Figure 1.89 (page 74)* dus niet als correct beoordeeld worden, want de vorm ervan voldoet niet aan de vereenvoudigde vorm.

TIP: Deze grading is er niet in voorzien dat teller en noemer nog door een factor gedeeld kunnen worden. Maar daar is de vraag niet naar in dit geval.

2) Bij de instelling **Maple syntax met Symbol entry only** is het van belang dat u weet hoe de invoer van de student vertaald wordt naar een string voordat u weet welke programmering in de *Grading Code* met StringTools nodig is. Eventueel kunt u zelf iets uitproberen zoals beschreven in paragraaf *De Analyse van de Response* (page 65).

In het onderstaande voorbeeld is te zien waarom we hier voor een dergelijke situatie beslist NIET voor *Text entry only* kiezen vanwege de automatische vereenvoudiging met *Preview*! Als alternatief kunt u hier uw toevlucht nemen tot de *Equation Editor* die automatisch aangeboden wordt bij de instelling *Symbol entry only* en waarin de student echt de vereenvoudiging moet invoeren, want automatisch wordt dat hier niet gedaan.

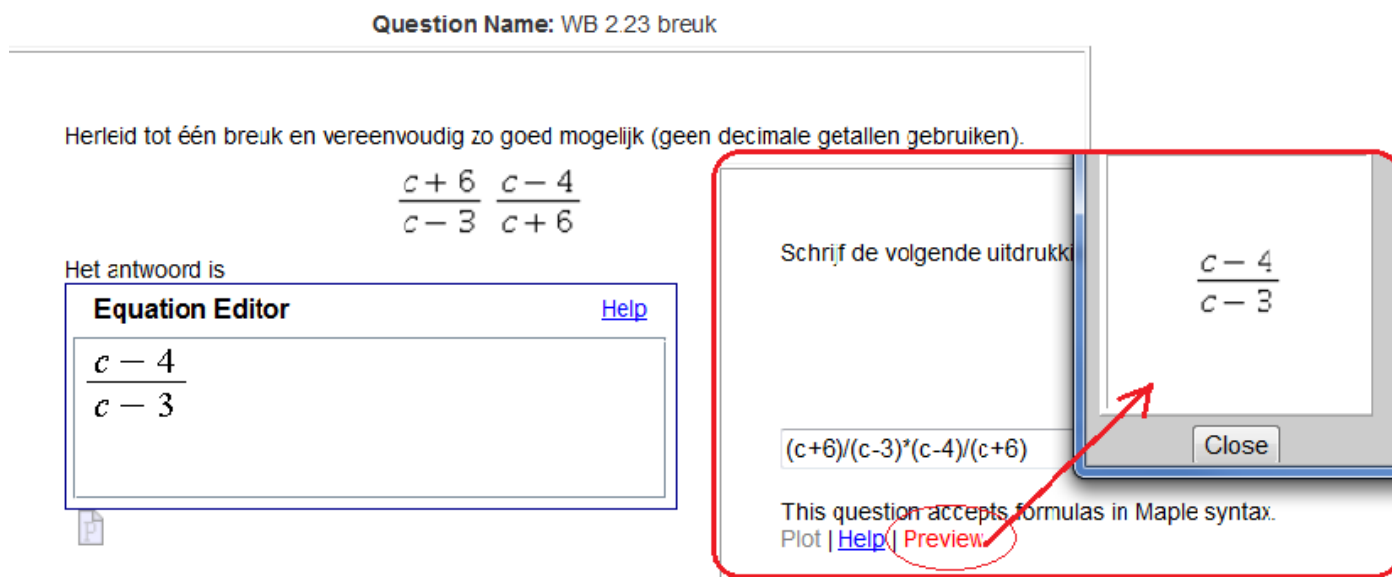


Figure 1.90: Breuken met instelling Symbol entry only

De programmering in de *Grading Code* is hier als volgt:

```
is(($antwoord)-($RESPONSE)=0) and evalb(StringTools[CountCharacterOccurrences]
("$RESPONSE", "(")=2) and evalb(StringTools[CountCharacterOccurrences]
("$RESPONSE", "/" )<=1) ;
```

Als u nog even kijkt in paragraaf *De Analyse van de Response* (page 65), zult u zien dat bij het vertalen naar een string van het antwoord, ingevoerd in de *Equation Editor*, altijd haakjes om teller en noemer zullen verschijnen. We dwingen in bovenstaande *Grading Code* bovendien af dat er slechts één breukstreep (of minder) wordt genoteerd. Echter hier even voorzichtig mee zijn, er zijn nog andere middelen om te checken of de student het antwoord in de juiste vorm geeft.

TIP: Als u toch liever de text entry wilt aanbieden in plaats van de editor, kunt u de automatische vereenvoudiging tegenhouden. Zie daarvoor paragraaf (page 64).

TIP: Geef in het *Answer*-veld altijd de MathML gecodeerde formule als correct antwoord! De student krijgt dan in de feedback ook altijd het twee-dimensionale antwoord te zien. Hij voert het antwoord immers ook tweedimensionaal in de Editor in.

3) Bij de formule-instelling **Formula** moet het gaan om formules die niet al te complex zijn en waar zeker geen gecombineerde karakters in voorkomen en ook geen exponentiële functies.

In het volgende voorbeeld (Figure 1.92 (page 76)) is gekozen voor deze instelling omdat het antwoord van de student anders met *Maple syntax* met *Text entry only* in de *Preview* vereenvoudigd zou worden. De instelling met *Maple syntax Symbol entry only* zou

wat problematisch zijn, omdat bij dit soort breuken het lastig is te eisen dat er slechts één slash wordt gebruikt. Immers de in de Editor ingevoerde formule wordt als string vertaald als volgt:

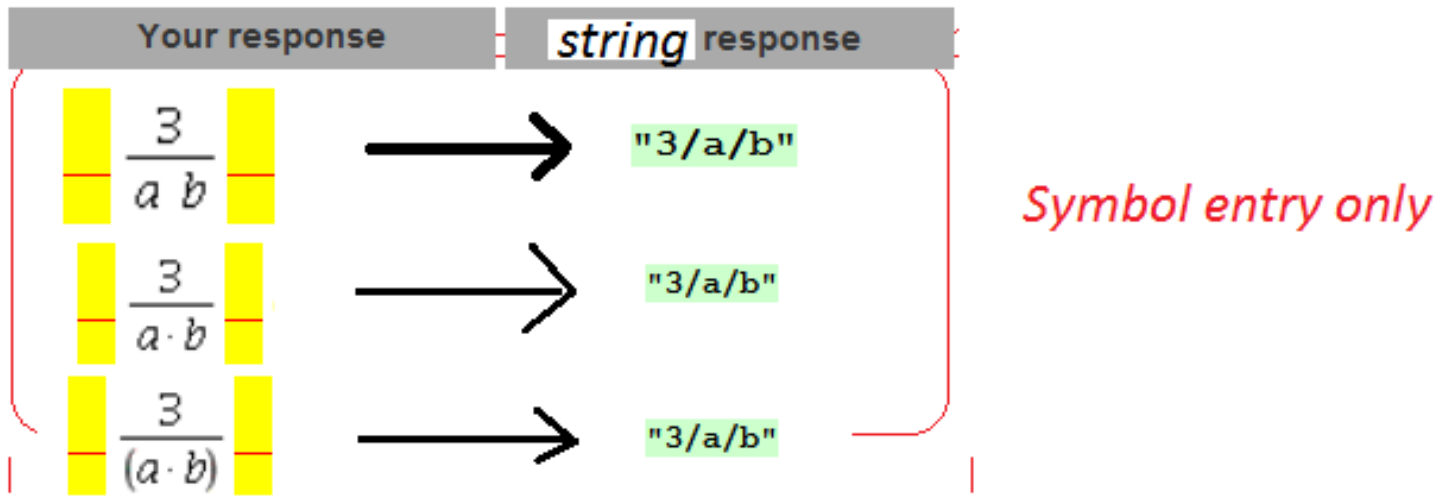


Figure 1.91: Breuken invoeren in de Editor

In de figuur hieronder is te zien dat er nu gekozen is voor de derde mogelijkheid om formules te overhoren met het *Maple-graded* vraagtype met de instelling *Formula*. Nogmaals: gebruikt u deze instelling alléén als het niet anders kan!

Question Name: WB 2 31 vereenvoudig 2

Vereenvoudig de volgende samengestelde breuk tot één breuk (met één deelstreep):

$$\frac{\frac{7}{a \ b}}{4 \ a}$$

This question accepts numbers or formulas
[Plot](#) | [Help](#) | [Change Math Entry Mode](#) | [Preview](#)

The screenshot shows a window titled 'Maple T.A. - Preview Math - Window' with the URL 'http://mapleta.can.nl:8080/mapleta/'. Inside the window, the fraction $\frac{7}{a \ b}$ is displayed over $4 \ a$. A red arrow points from the 'Preview' link in the question text to this window. A 'Close' button is visible at the bottom of the window.

Figure 1.92: Breuken invoeren als text bij de instelling Formula

In de figuur hierboven is te zien dat bij deze instelling (*Formula*) de *Preview* voor de student niets zal opleveren!

Het *Algorithm* van dit voorbeeld is als volgt geprogrammeerd.

```
$a=range(2,8);
$b=range(2,8);
$ggd=maple("igcd($a,$b)");
condition:eq($ggd,1);
$indexL=rint(3);
$L1=switch($indexL,"a","x","p");
$L2=switch($indexL,"b","y","q");
$vrage=maple("$a/($L1*$L2)/($b*$L1)");
$vrageDisplay=mathml("$a/($L1*$L2)/($b*$L1)");
$antwoord=maple("simplify($vrage)");
$antwoordDisplay=maple("printf(MathML[ExportPresentation]($antwoord))");
```

De twee getallen \$a en \$b zijn zó gerandomiseerd dat er geen gemeenschappelijke delers zijn. Dus de getallen kunnen niet vereenvoudigd worden in de breuk. Het gaat erom dat de student de vertaling goed legt naar de samengestelde breuk en er één breuk van dient te maken. De variabelen \$L1 (in dit geval dus de a in de samengestelde breuk) moeten bij elkaar genomen worden tot

$$\frac{7}{4a^2b}.$$

In de *Grading Code* is het volgende geprogrammeerd:

```
evalb(StringTools[CountCharacterOccurrences]("$RESPONSE", "/" ) <= 1) and
evalb(simplify(($antwoord)-($RESPONSE))=0) and
evalb(StringTools[CountCharacterOccurrences]("$RESPONSE", "$L1" ) = 1);
```

De *Grading Code* met gebruikmaking van StringTools kan dus evengoed voor de instelling *Formula* gebruikt worden, maar er is wel iets meer aandacht vereist met het oog op extra haakjes. Met de bovenstaande *Grading Code* wordt geëist dat de variabele \$L1 slechts één keer voorkomt in het stringantwoord van de student.

TIP: Let op dat deze variabele slechts uit één karakter bestaat, anders moet u SearchAll gebruiken.

```
evalb(nops({StringTools[SearchAll]("$x", "$RESPONSE")})=1);
```

(zie paragraaf *Werken met StringTools* (page 61))

Het zou lastig geweest zijn om slechts één slash te eisen als we er niet zeker van kunnen zijn hoe de formule naar de string vertaald wordt, zoals bij de instelling *Symbol entry only* en zoals te zien is in *Figure 1.91* (page 76). U kunt eventueel zelf uitproberen hoe de invoer vertaald wordt naar een string in bovenstaande situatie bij de instelling *Formula* van *Figure 1.92* (page 76).

Bij de instelling *Formula* wordt de invoer $4/(7*a^2*b)$ vertaald naar $"4/((7*(a^2))*b)"$.

U bent er dus zeker van dat er dan slechts één slash getikt is.

TIP: Bij deze instelling *Formula* is het dus heel lastig op haakjes te controleren met behulp van StringTools!!

TIP: De vraag is van het vraagtype *Maple-graded* waar het ook mogelijk is om helemaal niets in te voeren bij *Answer*. Dat is expres zo gedaan, want het is beter hier het correcte antwoord niet in te voeren als tekst, want dan zou er $4/7/a^2/b$ gecommuniceerd worden in de feed back waar u nou juist wilde dat er maar één breukstreep genoteerd werd door de student.

Eventueel kunt u in zo'n situatie het correcte antwoord in 2D notatie aanbieden in de *Feedback* als *Comment* óf bij *Answer* ook het correcte antwoord in 2D te noteren met de opdracht `printf(MathML[ExportPresentation]($antwoord))`;

Of met

```
use InertForm:-NoSimpl in ($steller)/($noemer): end: printf(InertForm:-ToMathML(%))
```

4) Hou altijd nog achter de hand dat u het antwoord kunt verdelen in **twee invulvelden** (of meer) zoals het volgende voorbeeld laat zien.

Question Name: WB 2 breuk vereenvoudigd en afgerond

Vereenvoudig de volgende vorm tot één breuk
en vul de teller en de noemer van de breuk apart in:

$$\frac{1}{5} + \frac{1}{9}$$

Afgerond op twee decimalen is dat:

Figure 1.93: Breuken met twee invulvelden voor teller en noemer

In dit voorbeeld moeten teller en noemer van het resultaat onafhankelijk van elkaar worden ingevuld. Er is dan per antwoordveld maar één antwoord mogelijk, namelijk een getal en dat kunt u gewoon met een numeriek veld overhoren. De student moet dan het resultaat vereenvoudigd hebben. De breukstreep kan gemaakt worden met het invoeren van een klein tabelletje met drie rijen en één kolom en border = 0. In de middelste rij plaatst u dan een horizontale lijn (met het knopje naast de tabel in de toolbar van de Editor van de vraag). In de andere vakken van de tabel komen de numerieke antwoordvelden.

De twee aparte antwoorden kunnen gemakkelijk in het *Algorithm* voorbereid worden:

```
$a1=range(2,15);
$b=range(2,20);
$a=if(ne(($a1),($b)),($a1),($a1)+1);
$aanwijzing1="bovenstaande betekent dat als a1 niet gelijk is aan b, dan a1, anders a1+1.";
$vraag=mathml("1/($a)+1/($b)");
$santwoord=maple("simplify(1/($a)+1/($b))");
$santwnumeriek=maple("evalf($santwoord)");
$AntwNum=decimal(2,$santwnumeriek);
$steller=maple("numer($santwoord)");
$noemer=maple("denom($santwoord)");
$santwoorddisplay=maple("printf(MathML[ExportPresentation]($santwoord))");
$ggd=maple("igcd($a,$b)");
$aanwijzing2="maak gebruik van de greatest common divisor = grootst gemene deler om te kijken of de noemers gemeenschappelijke factoren hebben.";
$commentaar1="Maak de breuken gelijknamig. In dit geval kun je het beste de nieuwe noemer kiezen door de twee noemers met elkaar te vermenigvuldigen.";
$commentaar2="Maak de breukengelijknamig. Let op dat de noemers een gemeenschappelijke factor hebben namelijk $ggd. Dus maak de noemer niet al te groot. Eventueel dus nog vereenvoudigen!!";
$commentaar=if(eq($ggd,1),$commentaar1,$commentaar2);
```

In bovenstaand *Algorithm* is te zien dat de vraag zelf met de opdracht `mathml("...")` gemaakt is omdat Maple de vorm sowieso direct zou vereenvoudigen. Met de Maple-commando's `numer` en `denom` voor respectievelijk teller en noemer, zijn gemakkelijk de afzonderlijke antwoorden voor de invulvelden te genereren.

TIP: In het script van het *Algorithm* kunnen ook aanwijzingen opgenomen worden in de vorm van een variabele. Hier is dat bijvoorbeeld `$aanwijzing1`.

TIP: Aardig om te zien is hier het voorbereide commentaar dat in de vorm van een variabele aan te roepen is om te gebruiken in de rubriek *Feedback*. Afhankelijk van de situatie is het commentaar verschillend.

1.4.10 Uitdrukkingen

Meestal is het controleren of het verschil tussen de correcte uitdrukking en de respons gelijk is aan 0, niet altijd voldoende. In de *Grading Code* kan er op meer fronten gecontroleerd worden of de respons aan de eisen voldoet.

Voorals u met decimale getallen wilt werken, valt het te proberen om de volgende *Grading Code* te proberen bij uitdrukkingen:

```
evalb( evalf( $p, 2 ) - ( evalf( $RESPONSE, 2 ) ) = 0 );
```

Het betekent dat het antwoord van de student en het correcte antwoord beide op 2 significante cijfers worden afgerond en dat ze daarna afgetrokken het getal nul moeten opleveren.

In het volgende voorbeeld moeten **haakjes weggewerkt** worden.

Question Name: 09 haakjeswegwerken

Werk van de volgende vorm alle haakjes weg en vereenvoudig het antwoord:

$$4(x - 1)^2$$

In dit voorbeeld wordt in de *Grading Code* geprogrammeerd dat er geen haakjes in het antwoord mogen komen. Verder wordt er een editor aangeboden want de Preview zou eventueel de vereenvoudiging kunnen laten zien.

Equation Editor
[Help](#)

$4x^2 - 8x + 4$

Figure 1.94: Haakjes weggewerken

In bovenstaande figuur zien we een uitdrukking waarvan de haakjes weggewerkt moeten worden. Ook zien we dat de instellingen op *Maple syntax Symbol entry only* staan. De student krijgt dan immers de *Equation Editor* aangeboden.

De *Grading Code* programmeren we dan zó dat niet alleen de respons gelijk moet zijn aan het correcte antwoord (in het *Algorithm* voorbereid als \$antwoord) maar ook mogen er geen haakjes in de respons voorkomen. Het is dan belangrijk dat gekozen wordt voor óf *Maple syntax* met *Text entry only* óf *Symbol entry only*, waar in beide gevallen het aantal haakjes prima gecheckt kan worden in de stringrespons.

Zie voor informatie daarover in paragraaf *De Analyse van de Response* (page 65) .

Dit programmeren we in de *Grading Code* als volgt:

```
evalb($RESPONSE=($antwoord)) and evalb(StringTools[CountCharacterOccurrences]("$RESPONSE",
"(")=0);
```

Er hoeft hier niet geprogrammeerd te worden dat het Maple-commando `expand` niet in de string van de respons mag voorkomen, want de instellingen staan hier immers op *Symbol entry only* wat betekent dat de student het antwoord moet intikken in de *Equation Editor* waar Maple-commando's niet gehonoreerd worden. Als u echter voor *Text entry only* kiest had de student met deze *Grading Code* ook het Maple-commando niet kunnen gebruiken. Immers er mogen geen haakjes in de string van het antwoord van de student voorkomen!

Nog een voorbeeld van het overhoren van **een uitdrukking waarvan de coëfficiënten decimale getallen** mogen zijn. Het is dan wat lastig om de uitdrukkingen van elkaar af te trekken en gelijk aan 0 te stellen als u ook nog wat marges in de coëfficiënten wilt toelaten.

Q 06 First order non homogeneous

Partial Grading Explained

Given the following difference equation

$$Y_{n+1} - 0.5 Y_n = 18$$

where Y_n represents income (€ m) in period n .

Note: round numbers correct to two decimal places (unless instructed otherwise)

1. Determine the general solution,

The general, $Y_n =$   (use the symbol A for the constant)

2. Determine the particular solution given $Y_1 = 15$.

The particular solution =  

Figure 1.95: Uitdrukking met decimale coëfficiënten

In het **tweede** antwoordveld van deze vraag moet als antwoord ingevuld worden het volgende:

$-101.25 \cdot 0.8^n + 100$.

(Het juiste antwoord is $\text{\$Ypart} = \text{maple}("\$Aa * (\$r)^n + \$YPi");$)

Hierin mag het getal $\$Aa = -101.25$ afgerond zijn op 2 decimalen en het getal $\$YPi = 100$ ook afgerond op twee decimalen (komt hier toevallig zo uit dat het een geheel getal is).

Voor dit antwoordveld van het vraagtype *Maple-graded* met de instelling *Text entry only* kunt u het volgende programmeren in de *Grading Code*:

```
evalb(abs(coeff($RESPONSE,$r^n)-($Aa))<0.02) and evalb(abs(op(2,sort($RESPONSE))-($YPi))<0.02);
```

Het betekent dat de coëfficiënt van $\$r^n$ (hier is dat dus 0.8^n) minder dan 0.02 mag afwijken van de correcte coëfficiënt $\$Aa$ (hier is dat -101.25). Het spreekt vanzelf dat de macht het grondtal 0.8 ($\$r$) moet hebben; daar zit geen marge in.

Verder moet de tweede operator van de uitdrukking, nadat deze op volgorde is gesorteerd, (hier is dat het getal 100) binnen een marge van 0.02 vallen met het correcte getal $\$YPi$.

Omdat het in beide gevallen gaat om de coëfficiënten af te ronden tot in de tweede decimaal en verder gaat het om een tweeterm, had u net zo goed ook kunnen programmeren met de volgende *Grading Code* om te matchen met het correcte antwoord $\text{\$Ypart}$:

```
evalb(nops($RESPONSE)=2) and evalb(map(x->round(100*x),{coeffs($Ypart)})=map(x->round(100*x),{coeffs($RESPONSE)}));
```


Hier wordt dus eerst het aantal operanden gecheckt (wat te maken heeft met de gedaante waarin het antwoord gegeven zal worden). Dit kan eventueel iets robuuster met `evalb(nops($RESPONSE)=nops($Ypart))`. Vervolgens worden alle coëfficiënten met 100 vermenigvuldigd en afgerond en met elkaar gematcht met behulp van een verzameling. Hieronder de figuur van dit antwoordveld met het *Algorithm* erbij. Soms is het handig om in Maple zelf een en ander uit te proberen en zo kunt u grotere uitdrukkingen splitsen in onderdelen.

```
> coeffs(-101.5*0.8^n+100);
```

$$100, -101.5 \quad (1.51)$$

```
> nops(-101.5*0.8^n+100);
```

$$2 \quad (1.52)$$

```
> coeff(-101.5*0.8^n+100,0.8^n);
```

$$-101.5 \quad (1.53)$$

```
> map(x->round(100*x),{coeffs(-101.5*0.8^n+100)});
```

$$\{-10150, 10000\} \quad (1.54)$$

Of met de elementwise operator (`~`):

```
> round~(100*~{coeffs(-101.5*0.8^n+100)});
```

$$\{-10150, 10000\} \quad (1.55)$$

Edit Response Area

Choose Question Type

- Essay
- Free Body Diagram
- List
- Maple-graded
- Math App
- Mathematical formula
- Multiple Choice
- Numeric
- Sketch

Maple-graded:

Weighting

Answer

(referenced when grading as \$ANSWER)

Grading Code:

evalb(abs(coeff(\$RESPONSE,\$r^n)-(\$Aa))<0.02) and
evalb(abs(op(2,sort(\$RESPONSE))-(\$YPi))<0.02);

Expression Type:

Text/Symbolic entry:

Optional:

Maple Repository:

Maple Repository

Figure 1.96: Grading code voor de coëfficiënten van een uitdrukking

Een soortgelijk voorbeeld staat nog genoemd in paragraaf *Coëfficiënten en Operanden* (page 69) in *Figure 1.88* (page 70).

Nog een voorbeeld om een **gedeelte van een uitdrukking goed te kunnen rekenen**:

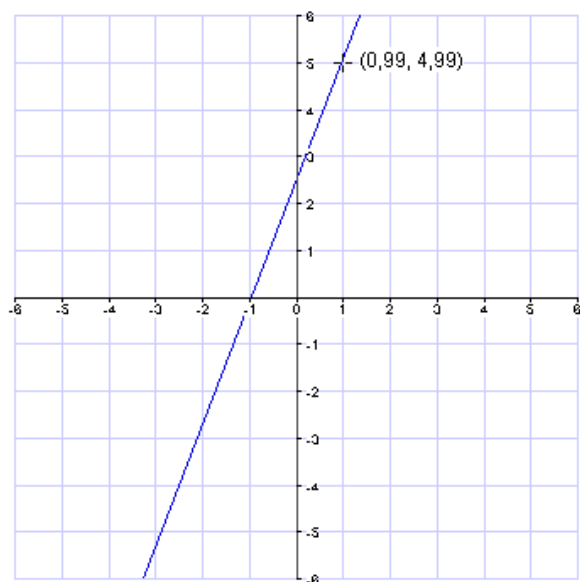
We werken dan met coëfficiënten van een veelterm. In dit geval van een eerstegraads polynoom.

Question Name: 21b grafiek rechte lijn 3 (aflezen en half goed)

Gegeven de grafiek van de rechte lijn.

In onderstaande figuur kun je met de muis de coördinaten in twee decimalen aflezen (linker muisknop ingedrukt houden en langzaam bewegen).

Je kunt ook met de muis slepen en de [Ctrl]-knop ingedrukt houden om in te zoomen en met de muis slepen en [Shift] ingedrukt houden om het venster te verplaatsen.



Bedenk de vergelijking van de rechte lijn ($y = a x + b$) waarbij je de coördinaten in de figuur moet aflezen.

Rond de uiteindelijke formule af op één decimaal nauwkeurig.

$y =$  

Klik eventueel even op "P" om te kijken of je de goede lijn hebt.

Figure 1.97: Lineaire uitdrukking met decimalen

Het *Algorithm* hiervan is vrij eenvoudig:

```
$a=switch(rint(2),decimal(1,rand(-3,-0.1)),decimal(1,rand(0.1,3)));
$b=switch(rint(2),decimal(1,rand(-3,-0.1)),decimal(1,rand(0.1,3)));
condition:not(eq($a,$b));
$p=maple("($a)*x+($b)");
$pdisplay=maple("printf(MathML:-ExportPresentation($p))");
$ljijn=maple("printf(MathML:-ExportPresentation(y=a*x+b))");
```

In de *Grading Code* kan het volgende geprogrammeerd worden:

```
if evalb(degree($RESPONSE,x)=1) then evalf(nops(map(x->round(10*x),{coeffs($RESPONSE)}))
  intersect map(x->round(10*x),{coeffs($p)}))/2) else 0 end if;
```

Er wordt hier eerst gecheckt of de graad van de ingevoerde uitdrukking in x gelijk is aan 1 en als dat het geval is, dan worden de verzamelingen van de coëfficiënten van de uitdrukking bekeken. Het aardige hier is dat het aantal elementen van de doorsnede van de verzamelingen wordt geteld met het Maple-commando `nops` en uiteindelijk wordt dat aantal gedeeld door 2 (het maximale aantal coëfficiënten) en met `evalf` omgezet naar een decimaal getal. De uitkomst zal dan 0 of 0.5 of 1 zijn.

TIP: Als u de *Grading Code* programmeert zodanig dat er een decimaal getal uit komt tussen 0 en 1, dan resulteert dat in het percentage van de grading. Als de uitkomst van deze *Grading Code* dus gelijk is aan 0.5 dan werd de vraag voor 50% goedgekeurd wat inhoudt dat er dus één van de coëfficiënten correct was.

TIP: Let wel goed op dat géén van de coëfficiënten gelijk is aan 0 en de coëfficiënten ook niet gelijk aan elkaar, anders klopt het niet met het aantal in de doorsnede, immers het maximale aantal geeft de correcte beoordeling en dubbele elementen in een verzameling worden als enkele gezien.

TIP: Op deze manier kunt u hogeregraads polynomen checken zoals dat op gelijksoortige manier gedaan is in het voorbeeld van *Figure 1.88 (page 70)* (waar niet per se ongelijke coëfficiënten vereist zijn).

Echter als u een polynoom van hogere graad wilt checken op deze manier (met gedeeltelijk goed) is het bijvoorbeeld voor een derdegraads polynoom de volgende *Grading Code*:

```
if evalb(degree($RESPONSE,x)=3) then evalf(nops(map(x->round(10*x),{coeffs($RESPONSE)}))
  intersect map(x->round(10*x),{coeffs($p)}))/4) else 0 end if;
```

Hier gaat het dus om een derdegraads polynoom met 4 coëfficiënten. Let op dat de coëfficiënten niet 0 zijn en ook niet gelijk aan elkaar! Het alternatief is dat u alle coëfficiënten afzonderlijk et elkaar vergelijkt.

1.4.11 Uitdrukkingen met de letter I

Als u een uitdrukking heeft met de letter I er in waar het niet de bedoeling is dat deze opgevat wordt als de imaginaire eenheid, dan moet op alle fronten dit kenbaar gemaakt worden in Maple opdrachten. Telkens weer opnieuw gaat u de letter I inactief maken of de imaginaire eenheid de letter j geven (of een andere letter).

```
$antw1 = maple("local I:=`I `:3*P*L/(E*I)");
```

```
$antw1display=maple("local I:=`I `:printf(MathML[ExportPresentation](3*P*L/(E*I)))");
```

Let op de spatie tussen de quotes.

Gebruik voor het antwoordveld een Maple-graded vraagtype en zet de instellingen op "Maple Syntax" (en vooral niet op "Formula"). Bij het correcte antwoord *Answer* geeft u de 2D weergave omdat de student bij de feedback de quotes niet moet zien.

Dus bij *Answer* voert u in:

```
printf(MathML[ExportPresentation]($antw1));
```

In de *Grading Code* moet u ook kenbaar maken dat I als symbool opgevat wordt want de student voert een I in:

```
local I:=`I `:evalb(($antw1)-($RESPONSE)=0);
```

Bij *Custom Previewing Code* kunt u ten behoeve van de *Preview* die de student gaat doen ook bewerkstelligen dat de hoofdletter I niet opgevat wordt als imaginaire eenheid door daar te programmeren:

```
local I:=`I `:printf(MathML[ExportPresentation]($RESPONSE));
```

De hoofdletter I wordt bij Maple opgevat
Tik de volgende formule na

$3 \frac{PL}{EI}$

$3 \cdot P \cdot L / (E \cdot I)$

Unit Response Area

- Essay
- Free Body Diagram
- List
- Maple-graded
- Math App
- Mathematical formula
- Multiple Choice
- Numeric
- Sketch

Weighting: 1

Answer: `printf(MathML[ExportPreser`
(referenced when grading as \$ANSWER)

Grading Code: `local I:='I':evalb(($antw1)-($RESPONSE)=0);`

Expression Type: Maple Syntax - e.g. diff(2*f(x),x)

Text/Symbolic entry: Text entry only

Optional:

Maple Repository:

Plotting Code:

Custom Previewing Code: `local I:='I':printf(MathML[ExportPresentation])($RESPONSE);`

Figure 1.98: De letter I niet als imaginaire eenheid

Als u werkelijk met complexe getallen wilt werken en bijvoorbeeld de i of de j wilt gebruiken als complex getal, bereid dan dingen voor in het Algorithm:

```
$test3= "1/(3+4*I)";
$test5=maple("local I:=j:printf(MathML[ExportPresentation])($test3)");
$test6= maple("1/(3+4*I)");
$test8=maple("local I:=j:printf(MathML[ExportPresentation])($test6)");
```

| | |
|-------|---------------------------------|
| test3 | $1/(3+4 \cdot I)$ |
| test5 | $\frac{1}{3+4j}$ |
| test6 | $3/25-(4/25) \cdot I$ |
| test8 | $\frac{3}{25} - \frac{4}{25} j$ |

Figure 1.99: Berekeningen met complexe getallen

Als U gemakkelijk wilt wisselen tussen de i en de j als imaginaire eenheid, kijk dan eens in het volgende :

```
$index=1;
$ij=switch($index,"i","j");
```

```

$a1=switch(rint(2),range(-10,-2),range(2,10));
$b1=switch(rint(2),range(-10,-2),range(2,10));
$a2=switch(rint(2),range(-10,-2),range(2,10));
$b2=switch(rint(2),range(-10,-2),range(2,10));
$vraag="$a1+($b1)*I+($a2)+($b2)*I";
$vraagdispij="$a1+($b1)*$ij+($a2)+($b2)*$ij";
$vraagdisplay=maple("use InertForm:-NoSimpl in $vraagdispij: end: printf(InertForm:-ToMathML(%))");
$santw=maple("evalc($vraag)");
$santwoord=maple("local I:=$ij: $santw");
$santwoorddisplay=maple("printf(MathML[ExportPresentation]($santwoord))");
$plot=plotmaple("plots[display]([plots[complexplot]([$santw], style = point, symbol = solidcircle, symbolsize =
15,color=blue),plots[complexplot]([0], style = point, symbol = solidcircle, symbolsize = 10,color=white)],title=`complex
vlak`,caption=`blauw is het juiste punt`,gridlines=true)");

```

Het antwoord in het invulveld is dus \$santwoord en daarmee moet ook gematcht worden in de grading code.

Eventueel plotting code:

```

resp:=subs($ij=I,$RESPONSE):antw:=$santw:plots[display]([plots[complexplot]([resp], style = point, symbol =
solidcircle, symbolsize = 15,color=red),plots[complexplot]([antw], style = point, symbol = solidcircle, symbolsize =
10,color=blue),plots[complexplot]([0], style = point, symbol = solidcircle, symbolsize = 10,color=white)],title=`complex
vlak`,caption=`rood is jouw punt en blauw het juiste punt`,gridlines=true);

```

1.4.12 Wortelvormen

Heel lastig is het om wortelvormen te overhoren waarbij de vereenvoudiging gevraagd wordt.

U kunt met StringTools een heel eind komen door de string van de respons van de student te checken op het al of niet voorkomen van bepaalde karakters of combinaties daarvan. Zie ook paragraaf *Werken met Stringtools* (page 61).

Maar eerst nemen we de invoer van de student nog eens even onder de loop bij de instelling **Maple Syntax Text entry only**.

Maple Syntax Text entry only:

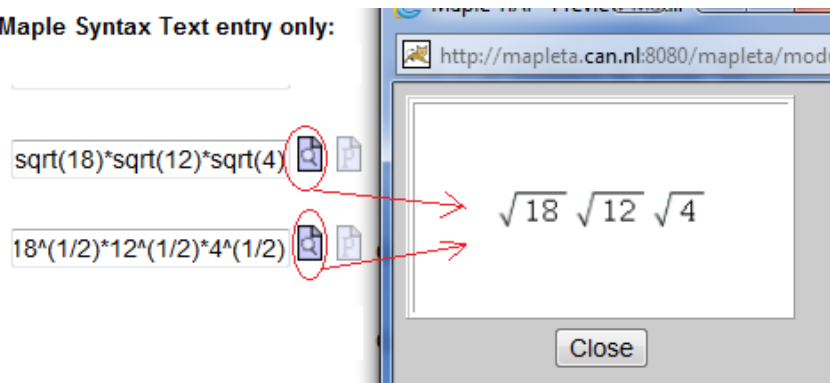


Figure 1.100: Hoe worden wortels vereenvoudigd

In bovenstaande figuur zien we dat de *Preview*-functie in beide gevallen (ingevoerd met `sqrt` óf met `^(1/2)`) hetzelfde oplevert: er wordt NIET vereenvoudigd wat betreft het kleiner maken van de getallen onder het wortelteken! (Let echter op met breuken.) Wortels worden ook niet samengenomen en dus ook niet vereenvoudigd in de *Preview* tenzij het twee dezelfde wortels zijn. Dus u kunt wat dat betreft rustig de *Maple Syntax* instelling met *Text entry only* hanteren voor dit soort vragen.

Met de truc van paragraaf *De Analyse van de Response* (page 65) gaan we nu bekijken wat het oplevert als we de string van het antwoord en ook het antwoord zelf onder de loop nemen.

Maple Syntax Text entry only:

| | | |
|--|--|---------------------------|
| <code>sqrt(18)*sqrt(12)*sqrt(4)</code> | <code>"sqrt(18)*sqrt(12)*sqrt(4)"</code> | <code>"\$RESPONSE"</code> |
| <code>sqrt(18)*sqrt(12)*sqrt(4)</code> | <code>12*2^(1/2)*3^(1/2)</code> | <code>\$RESPONSE</code> |
| <code>18^(1/2)*12^(1/2)*4^(1/2)</code> | <code>"18^(1/2)*12^(1/2)*4^(1/2)"</code> | <code>"\$RESPONSE"</code> |
| <code>18^(1/2)*12^(1/2)*4^(1/2)</code> | <code>18^(1/2)*12^(1/2)*4^(1/2)</code> | <code>\$RESPONSE</code> |

Figure 1.101: De respons en de stringrespons bij verschillende manieren van intikken van de wortels

In bovenstaande figuur zien we in de eerste twee regels dat de notatie van de wortels, door de student ingevoerd met "sqrt", exact overgenomen wordt in de string. Echter in de respons van de student worden de wortels omgezet naar de macht (1/2) en tegelijk ook vereenvoudigd, maar niet samengenomen.

Voert de student echter de wortels in met behulp van $^{(1/2)}$ zoals in de derde en vierde regel van bovenstaande figuur, dan wordt de string van de invoer weer precies overgenomen zoals verwacht bij deze instelling van *Text entry only*, maar de respons wordt vreemd genoeg niet vereenvoudigd door de getallen onder het wortelteken kleiner te maken.

U moet er rekening mee houden dat de student beide notaties zou kunnen gebruiken.

In de volgende voorbeelden gaan we kijken op welke manieren de *Grading Code* geregeld kan worden, steeds met de instellingen van *Text entry only*.

Uiterste middel is controleren of de student precies het antwoord tikt zoals u dat verlangt. Daarbij zijn altijd twee mogelijkheden bij het intikken de wortel, namelijk met sqrt en met de macht (1/2).

Question Name: 08 wortels vereenvoudigen

Vereenvoudig de volgende wortelvorm $\sqrt{20}$.

Deze vraag heeft als grading het vergelijken van stringantwoorden. Dus de student dient letterlijk het voorgeschrevene in te tikken. De stringantwoorden zijn in het *Algorithm* voorbereid om optimale controle te houden.

Als correct antwoord wordt de 2D formule gepresenteerd.

De instellingen zijn *Maple syntax* met *Text entry only* of met *Symbol entry only* allebei mogelijk. Echter NIET met *Formula*!

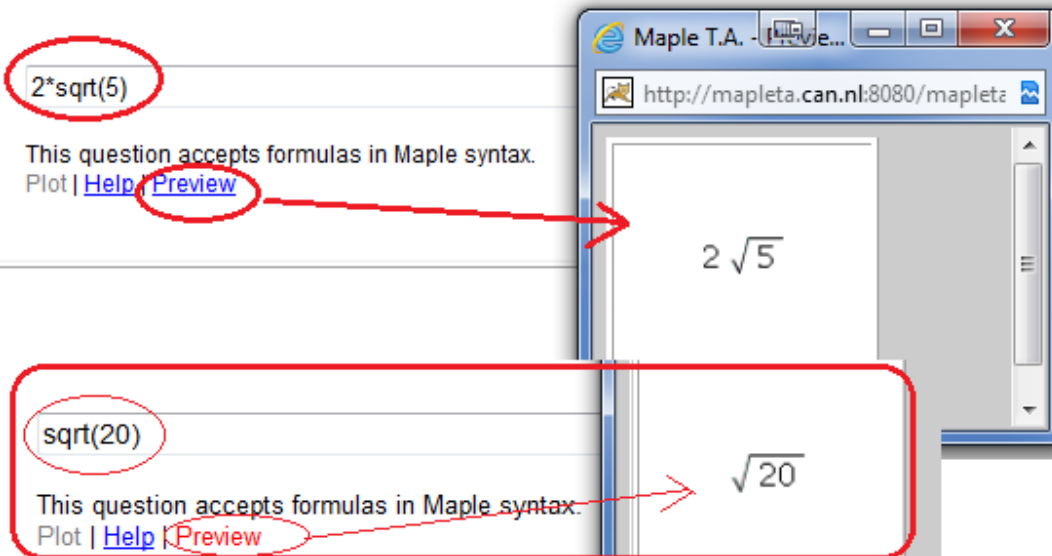


Figure 1.102: Wortels vereenvoudigen

In bovenstaande figuur zien we een vraag waar de vereenvoudiging van een wortel getoetst wordt. De *Preview* van de instelling *Maple syntax Text entry only* geeft twee dimensionaal weer wat de student heeft ingetikt. Als de student de vraag overtoetst dus in dit geval $\sqrt{20}$ zou hebben ingevuld, wordt de vereenvoudiging bij dit soort wortelformules in de *Preview* niet gedaan, dus u kunt rustig *Text entry only* instellen.

In de toelichting bij de vraag in Figure 1.102 (page 87) staat dat het correcte antwoord in 2D gepresenteerd wordt. Dat is iets om over na te denken. Immers als u het correcte antwoord in text-formaat aanbiedt, zal de student de notatie met $^{1/2}$ zien en zal denken dat hij volgende keer dus altijd genoemde notatie moet hanteren!

In de volgende figuur is in het bijbehorende *Algorithm* te zien hoe de mogelijke antwoorden zijn voorbereid als string.

Deze manier van doen vereist wel dat de student strict een bepaalde manier van invoeren hanteert. Veel flexibiliteit is hierbij niet.

```

$a=range(2,4,1);
$b=range(3,7,2);
$p=$a^2*$b;
$vraag=mathml("sqrt($p)");
$antwdisplay=maple("printf(MathML[ExportPresentation]($a*sqrt($b))");
$stringantw1=maple("convert($a*sqrt($b),string)");
$stringantw2=maple("convert($a*'sqrt($b)',string)");

```

| Variable | Value |
|-------------|-------------|
| a | 3 |
| b | 5 |
| p | 45 |
| vraag | $\sqrt{45}$ |
| antwdisplay | $3\sqrt{5}$ |
| stringantw1 | "3*5^(1/2)" |
| stringantw2 | "3*sqrt(5)" |

Figure 1.103: Wortels vereenvoudigen in Algorithm

In bovenstaande figuur zien we het *Algorithm* waar de stringantwoorden worden voorbereid en gecontroleerd.

Merk op dat de vraag \$vraag wordt voorbereid met behulp van de opdracht *mathml*, omdat met de MathML-conversie met Maple de automatische vereenvoudiging hier niet tegen te houden zou zijn.

Let ook op de quotes bij \$stringantw2 om evaluatie naar $5^{(1/2)}$ te voorkomen en ook echt $\sqrt{5}$ te krijgen.

Er worden dus twee antwoorden voorbereid waarmee straks in de *Grading Code* het antwoord van de student vergeleken kan worden op de volgende manier.

```
evalb("$RESPONSE"=$stringantw1) or evalb("$RESPONSE"=$stringantw2);
```

TIP: Als u vermoedt dat de student spaties zou kunnen tikken in zijn antwoord, dan kunt u iets zwaarder programmeren door eerst alle spaties eruit te halen met:

```
with(StringTools); stringrespons:=Remove(IsSpace,"$RESPONSE");
evalb(stringrespons=$stringantw1) or evalb(stringrespons=$stringantw2);
```

Het maakt voor *Maple syntax* immers niet uit of er hier en daar spaties getikt worden (rond operatoren). Spaties in de *Maple syntax* worden toch genegeerd. Echter in een string telt een spatie wel als karakter mee. Met bovenstaande code wordt het antwoord van de student als string bekeken en worden eerst alle spaties eruit gehaald. Dat resultaat wordt met het juiste antwoord vergeleken.

TIP: Hier wordt dus geëist dat hetgeen de student intikt precies *letterlijk* overeenkomt met mogelijke antwoorden in de vorm van een string: "exact text match". Een dergelijke vraag kan inderdaad ook gesteld worden met het vraagtype *List* in de *Question Designer*, waar het mogelijk is met *exact text match* te controleren of het goede antwoord is ingetikt. Echter bij dat vraagtype wordt er natuurlijk niet een *Preview*-knop aangeboden of een Editor om de formule in te vullen. Ook hoeft het antwoord bij vraagtype *List* niet als "string" te worden voorbereid, maar kunt u bij het vraagtype *List* gewoon de mogelijke antwoorden als formule intikken of als formule voorbereiden in het *Algorithm* en er vervolgens een gewicht (*Weight*) aan toekennen.

Het vraagtype *List* binnen de *Question Designer* wordt besproken in de *Handleiding Toets items maken Deel A*.

In het volgende voorbeeld is de manier van invoeren voor de student iets flexibeler. Hier ook weer de instelling op *Text entry only* en de opdracht om de vorm te vereenvoudigen. Het gaat er hierom dat in ieder geval het getal onder het wortelteken kleiner gemaakt moet worden voor de rest is het niet belangrijk.

Question Name: 08a Wortelherleiden 01a

Vereenvoudig de volgende wortelvorm.

$$\frac{-6 + \sqrt{180}}{8}$$

Het antwoord is






Figure 1.104: De wortelvorm vereenvoudigen

In het bijbehorende *Algorithm* wordt ervoor gezorgd dat het getal onder het wortelteken altijd kleiner gemaakt kan worden (door \$b een even getal te laten zijn lukt dat altijd).

```

$a=range(2,10);
$b=switch(rint(2),range(2,10,2),range(-10,-2,2));
$c=switch(rint(2),range(2,10),range(-10,-2));
$wortel=mathml("sqrt(3)");
$D=($b)^2-4*($a)*($c);
$noemer=2*$a;
$antwoord=maple("simplify((-($b)+sqrt($D))/(2*$a))");
$displayantwoord=maple("printf(MathML[ExportPresentation]($antwoord))");
$displayvraag=mathml("(-$b+sqrt($D))/$noemer");
condition:gt($D,0);

```

| Variable | Value |
|-----------------|------------------------------|
| a | 3 |
| b | 6 |
| c | -10 |
| wortel | $\sqrt{3}$ |
| D | 156 |
| noemer | 6 |
| antwoord | $-1+1/3 \cdot 39^{1/2}$ |
| displayantwoord | $-1 + \frac{1}{3} \sqrt{39}$ |
| displayvraag | $\frac{-6 + \sqrt{156}}{6}$ |

Figure 1.105: title of the figure

In bovenstaand *Algorithm* is \$D de vorm die onder het wortelteken komt te staan. (Er is voor gezorgd dat deze \$D groter is dan 0.) Let ook eens op hoe de formule van de vraag gemaakt is met de opdracht mathml(...) omdat met Maple de vereenvoudiging niet tegen te houden is.

Als u nu de wortelvorm altijd vereenvoudigd wilt zien, maar voor de rest geen eisen stelt, kunt u bijvoorbeeld programmeren dat in bovenstaand voorbeeld het getal \$D niet mag voorkomen.

U programmeert dan in de *Grading Code* het volgende:

```
is(simplify(($ANSWER)-($RESPONSE))=0) and evalb(StringTools[Search]("$D", "$RESPONSE")=0);
```

U kunt ook het antwoord van de student in **meer delen splitsen** zoals het volgende voorbeeld laat zien.

08b
rekenenmetwortels
03

Bereken en vereenvoudig de volgende vorm, waarbij eventuele in het antwoord voorkomende breuken als breuk (dus teller/noemer) geschreven moeten worden maar geen wortels in de noemer.

$$\frac{14}{\sqrt{12}}$$

Vul eerst het getal of breuk in en vervolgens de wortel (met sqrt).

7/3

×

sqrt(3)

Figure 1.106: Splitsen van wortelvormen

Kijk dan eens naar de operanden van het antwoord. U kunt dan van elk invulveld apart de *Grading Code* als volgt formuleren.

```
evalb(op(1,$antwoord)=$RESPONSE);
```

```
evalb(op(2,$antwoord)=$RESPONSE);
```

In het *Algorithm* kunt u uitproberen wat de operanden zijn van een uitdrukking.

```
$a=range(2,15);
$b=switch(rint(8),2,3,5,6,7,8,10,12);
condition:ne($a,$b);
$displayvraag=mathml("($a)/sqrt($b)","nosimplify");
$antwoord=maple("simplify(($a)/sqrt($b))");
$displayantwoord=maple("printf(MathML[ExportPresentation]($antwoord))");
$operanden=maple("op($antwoord)");
```

| Variable | Value |
|-----------------|-------------------------|
| a | 2 |
| b | 10 |
| displayvraag | $\frac{2}{\sqrt{10}}$ |
| antwoord | $1/5 \cdot 10^{(1/2)}$ |
| displayantwoord | $\frac{1}{5} \sqrt{10}$ |
| operanden | $1/5, 10^{(1/2)}$ |

Figure 1.107: De operanden van een wortelvorm

In dit geval bent u wel wat beperkt en moet u er natuurlijk voor zorgen dat de vorm inderdaad wel bestaat uit twee operanden: in die zin dat er altijd een getal of breuk voor de wortel komt gevolgd door de wortel. Dat is in ieder geval zo als \$a niet gelijk is aan \$b.

Als \$a\$ wel gelijk is aan \$b\$ bestaat het antwoord alleen uit een wortel en dat is hier niet de bedoeling, dus staat er een extra conditie in het *Algorithm*.

Overigens bestaat een enkele wortel wel weer uit twee operanden ($10^{(1/2)}$).

In de volgende regels ziet u een aantal Maple-opdrachten om een en ander uit te proberen in Maple zelf.

```
> op(sqrt(10));
```

$$10, \frac{1}{2} \quad (1.56)$$

```
> op(2*sqrt(10));
```

$$2, \sqrt{10} \quad (1.57)$$

```
> op(2*sqrt(6)*7*sqrt(50));
```

$$70, \sqrt{6}, \sqrt{2} \quad (1.58)$$

```
> op(3/sqrt(2));
```

$$\frac{3}{2}, \sqrt{2} \quad (1.59)$$

```
> op([2,2],3/sqrt(2));
```

$$\frac{1}{2} \quad (1.60)$$

In bovenstaande Maple-opdrachten wordt ook duidelijk dat de getallen steeds automatisch bijelkaar genomen worden, maar de wortels niet (tenzij er twee gelijke wortels voorkomen, die wel bijelkaar genomen worden).

In de laatste regel wordt er met een lijstje gevraagd naar de tweede operand en daarvan wordt de tweede operand genomen.

Het had ook anders gekund met dit soort vragen door te eisen dat er geen breukstreep in voorkomt of misschien het aantal breukstrepen te tellen. Echter hier is even een aantekening bij te geven: de student kan een wortel ook met behulp van de macht ($1/2$) geven en daar komt een slash in voor. Bied in dat geval dan de Editor aan met *Symbol entry only*, want het wortelteken van de Editor wordt altijd naar de string van de respons vertaald met behulp van sqrt dus daarin zit geen slash.

Echter wees toch even voorzichtig met het aantal slashes in de string van het antwoord.

In de volgende figuur wordt de invoer van de student omgezet en bekijken we hoe dat gaat.

Maple Syntax Symbol entry only:

"2/sqrt(2)/sqrt(3)"

"\$RESPONSE"

1/3*2^(1/2)*3^(1/2)

\$RESPONSE

Figure 1.108: De analyse van de wortelvorm in de Editor ingevoerd

In bovenstaande zien we dat de *string* van de wortelvorm die de student invoert in de Editor ALTIJD vertaald wordt met `sqrt`. (Echter in de respons heeft automatische vereenvoudiging plaatsgevonden, maar de wortels worden niet samengenomen.)

Nu volgt een voorbeeld van ongeveer dezelfde vraag als *Figure 1.106 (page 90)* maar nu dus met de instelling *Symbol entry only* om het aantal breukstrepen te kunnen checken in de string van het antwoord van de student. Immers de student voert de wortel in met behulp van de knoppen in deze *Editor* en de wortels worden zodoende altijd vertaald naar een string met `sqrt` (en dus niet met `^(1/2)`).

08c
rekenenmetwortels
02

Bereken en vereenvoudig de volgende vorm.

Dus géén wortels in de noemer van de breuk.

$$\frac{5}{\sqrt{5}}$$

Equation Editor

a^b

$\sin(a)$

$\frac{\partial}{\partial x} f$

$\begin{matrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{matrix}$

∞

α

Ω

↩

Help

Figure 1.109: Wortelvorm met Editor

In het *Algorithm* is er nu voor gezorgd dat er in het antwoord helemaal geen breuken komen te staan, dus u kunt dan rustig de string controleren op slashes.

```
$a=switch(rint(5),2,3,5,6,7);
$b=range(1,10);
$ab=$a*$b;
$displayvraag=mathml("$ab/sqrt($a)");
$antwoord=maple("simplify($b*($a)/sqrt($a))");
$displayantwoord=maple("printf(MathML[ExportPresentation]($antwoord))");
```

| Variable | Value |
|-----------------|-----------------------|
| a | 3 |
| b | 7 |
| ab | 21 |
| displayvraag | $\frac{21}{\sqrt{3}}$ |
| antwoord | $7 \cdot 3^{(1/2)}$ |
| displayantwoord | $7 \sqrt{3}$ |

Figure 1.110: Algorithm bij Wortelvorm met Editor

Dan programmeert u in de *Grading Code* het volgende:

```
is(simplify(($antw)-($RESPONSE))=0) and evalb(StringTools[Search]("/", "$RESPONSE")=0);
```

TIP: Als u dus de Editor aanbiedt met de instelling *Symbol entry only*, geef dan ook in het antwoordveld de 2D presentatie van het antwoord met `printf(MathML[ExportPresentation]($antw));` Let dan wel op dat u niet `$ANSWER` hanteert want deze variabele zou dan verwijzen naar de mathml-gecodeerde formule waar u niet mee kunt rekenen.

In het volgende voorbeeld hanteren we ook met voorbedachte rade de instelling *Symbol entry only*.

In dit voorbeeld moet de student de getallen én de wortels samennemen en bovendien het uiteindelijke getal onder het wortelteken zo klein mogelijk maken.

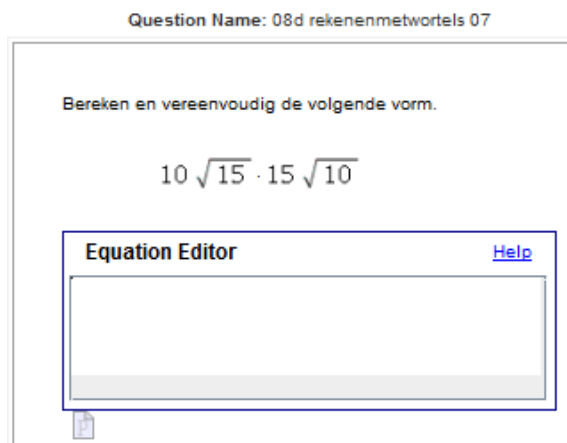


Figure 1.111: Wortelvormen checken met operanden van het antwoord

Het *Algorithm*:

```
$a=switch(rint(9),3,5,6,7,10,11,13,14,15);
$b=switch(rint(9),3,5,6,7,10,11,13,14,15);
condition:ne($a,$b);
$vraag1=maple("printf(MathML[ExportPresentation]($a*sqrt($b)))");
$vraag2=maple("printf(MathML[ExportPresentation]($b*sqrt($a)))");
$antwoord=maple("(($a)*($b)*sqrt($a*$b))");
$displayantwoord=maple("printf(MathML[ExportPresentation]($antwoord))");
$op=maple("op([2,1],$antwoord)");
```

| Variable | Value |
|-----------------|-----------------------|
| a | 11 |
| b | 6 |
| vraag1 | $11\sqrt{6}$ |
| vraag2 | $6\sqrt{11}$ |
| antwoord | $66 \cdot 66^{(1/2)}$ |
| displayantwoord | $66\sqrt{66}$ |
| op | 66 |

Figure 1.112: Algorithm van Wortelvormen checken met operanden

In het *Algorithm* wordt met de keuze van de variabelen ervoor gezorgd dat het antwoord altijd uit twee operanden bestaat. Dat wil zeggen uit een getal met een wortel. Ook wordt ervoor gezorgd dat het uiteindelijke getal onder het wortelteken (\$op) even

tevoorschijn gehaald wordt ten behoeve van de *Grading Code*. (Het is de tweede operand van het antwoord en daarvan de eerste operand.)

Verder zien we in het *Algorithm* de vraag die uit twee gedeelten bestaat \$vraag1 en \$vraag2. In de Editor van het tekstveld kunnen we er dan een stip tussen zetten zoals te zien is in *Figure 1.111* (page 93).

De *Grading Code* is dan:

```
is(op(1,$antwoord)=op(1,$RESPONSE)) and is(op(2,$antwoord)=op(2,$RESPONSE)) and
evalb(StringTools[Search]("sqrt($op)","$RESPONSE")>0);
```

Hierin wordt er dus voor gezorgd dat er in ieder geval ook "sqrt(\$op)" in het antwoord voorkomt. Daarmee wordt er voor gezorgd dat de wortel uit bijvoorbeeld 20 (dus "sqrt(20)") niet mag maar de wortel uit 5 wel. Immers de wortel uit \$a*\$b moet vereenvoudigd worden als dat kan.

TIP: Let er dan wel op dat u *Symbol entry only* instelt, want anders kan de student de wortel met de macht (1/2) invoeren en dan doet de opdracht evalb(StringTools[Search]("sqrt(\$op)","\$RESPONSE")>0); in de *Grading Code* niets. Immers bij de instelling *Symbol entry only* worden ALTIJD de wortels die ingevoerd zijn in de string vertaald naar sqrt(...).

Ook bij de instelling *Symbol entry only* is het handig om het correcte antwoord bij *Answer* in 2D te programmeren met printf(MathML[ExportPresentation](\$antwoord));

TIP: Als de respons van de student meer wortels bevat, dan worden deze allemaal als operanden gezien. De wortels die de student afzonderlijk zou intikken worden nooit automatisch samengenomen tot één wortel in de \$RESPONSE. Dus in het voorbeeld hierboven maken we er dan juist gebruik van om de operanden van de student stuk voor stuk te checken.

Nog een voorbeeld met het testen van het **aantal operanden** waaruit een antwoord moet bestaan:

Question Name: 08e rekenenmetwortels 08

Bereken en vereenvoudig de volgende vorm.

$$2\sqrt{3} \cdot \sqrt{18} \cdot \sqrt{5}$$

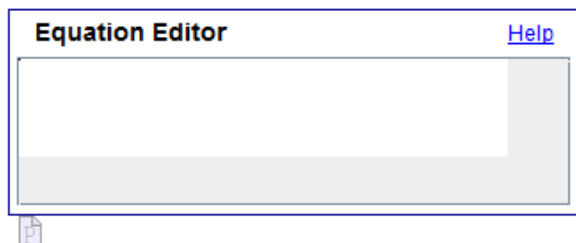


Figure 1.113: Wortels vereenvoudigen

Bij deze vraag is het iets lastiger om ervoor te zorgen dat de wortels elkaar niet opheffen en dat er in ieder geval een wortel in het antwoord komt. In het *Algorithm* programmeren we dat als volgt.

```
$a=switch(rint(6),2,3,5,6,7,10);
$b=switch(rint(3),8,12,18);
$c=switch(rint(5),2,3,5,6,7,10,11);
$d=range(2,6);
$vraag1=mathml("$d*sqrt($a)");
$vraag2=mathml("sqrt($b)");
$vraag3=mathml("sqrt($c)");
$antwoord=maple("$d*(sqrt(($a)*($b)*($c)))");
$n=maple("nops($antwoord)");
condition:ne($n,1);
$displayantwoord=maple("printf(MathML[ExportPresentation]($antwoord))");
```

```
$op=maple("op([2,1],$antwoord)");
```

In het *Algorithm* is te zien dat de onderdelen van de vraag nu voorbereid zijn met de opdracht `mathml("...")` om te voorkomen dat bijvoorbeeld de wortel uit 18 niet automatisch vereenvoudigd wordt.

Verder wordt er geprogrammeerd dat het aantal operanden van het antwoord in ieder geval niet gelijk is aan 1 zodat we er zeker van zijn dat de wortels elkaar niet uitdoven.

Met de variabele `$op` kunnen we in de grading code weer controleren of de wortel uit `$op` in het stringantwoord voorkomt in de vorm van `"sqrt($op)"` en dat is dan ook de reden om voor de instelling met *Symbol entry only* te kiezen. Immers bij gebruik van *Symbol entry only* wordt ALTIJD de wortel vertaald naar `sqrt`.

Dezelfde *Grading Code* als boven kan dan weer gebruikt worden:

```
is(op(1,$antwoord)=op(1,$RESPONSE)) and is(op(2,$antwoord)=op(2,$RESPONSE)) and
evalb(StringTools[Search]("sqrt($op)","$RESPONSE")>0);
```

Nog een iets ingewikkelder te programmeren vereenvoudigingsvraag met wortels is de volgende.

Question Name: 08f rekenenmetwortels 11

Bereken en vereenvoudig de volgende vorm, waarbij eventuele in het antwoord voorkomende breuken als breuk (dus teller/noemer) geschreven moeten worden. Dus ook geen wortels in de noemer van de breuk.

$$\frac{7\sqrt{6}}{6\sqrt{7}}$$

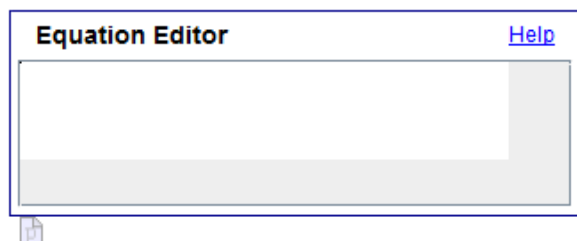


Figure 1.114: Wortels met breuken vereenvoudigen

Met het volgende *Algorithm*:

```
$a=range(2,10);
$b1=range(2,9);
$b=if(ne(($a),($b1)),($b1),($b1)+1);
$displayvraag=mathml("($b*sqrt($a))/($a*sqrt($b))");
$antwoord=maple("($b/$a)*sqrt($a/$b)");
$displayantwoord=maple("printf(MathML[ExportPresentation]($antwoord))");
$n=maple("if type(op(2,$antwoord),`^`) then 0 else 1 end if");
condition:ne($n,1);
$op=maple("op([2,1],$antwoord)");
```

Om te controleren of de tweede operand van het antwoord een wortel is (van het type `^`), testen we dat met `$n` en zorgen ervoor dat `$n` niet gelijk is aan 1 want als `$n = 0` dan is de tweede operand in ieder geval een wortel. In het antwoord zoeken we dan naar `"sqrt($op)"` met `.....and evalb(StringTools[Search]("sqrt($op)","$RESPONSE")>0);` Net als in het vorige voorbeeld en daarom gebruiken we dan ook de instelling *Symbol entry only*.

Als we hadden volstaan met alleen het eisen van het aantal operanden gelijk aan 2, dan had het ook gekund dat het antwoord een breuk was en die bestaat immers ook uit twee operanden: teller en noemer.

TIP: Ten slotte nog een andere mogelijkheid die nog gebruikt kan worden in het geval er met wortels in vereenvoudigde vorm gewerkt wordt, is uit te wijken naar een totaal ander vraagtype namelijk het vraagtype (*Mathematical*) *Formula* met speciale instellingen om vereenvoudigde vormen te overhoren. Zie daarvoor in paragraaf *Formula without simplification* (page 132).

1.4.13 Vergelijkingen

Bij het matchen van vergelijkingen komt het er niet simpel op neer dat u de vergelijkingen van elkaar aftrekt en vervolgens te eisen dat er nul uit komt.

Als de student een vergelijking moet invullen, kan het zijn dat hij de vergelijking wellicht vereenvoudigt of anders opschrijft. Het gaat er om dat hij een vergelijking invult die *gelijkwaardig* is met de correcte vergelijking.

TIP: Gebruik steeds Maple-graded vraagtype met de instelling *Maple syntax*.

TIP: Het ligt bij vergelijkingen gevoeliger als u decimale getallen wilt hanteren:

De twee getallen 4 en 4.0 worden wel als gelijk beschouwd, maar de vergelijkingen $x=4$ en $x=4.0$ worden niet als gelijk gezien!!! Gebruikt u dan `convert(vergelijking,rational)` om de decimalen kwijt te raken.

In het volgende voorbeeld moet de student een vergelijking invullen in de variabelen x , y en z waarbij het ook wel eens kan zijn dat één of meer van deze variabelen ontbreekt.

Question Name: 02a Lineair Systeem

Schrijf de eerste vergelijking op die correspondeert met het lineaire systeem waarvan de aangevulde matrix hieronder is gegeven door:

$$\left(\begin{array}{ccc|c} 7 & -2 & 13 & -7 \\ -6 & -5 & 15 & 6 \\ 0 & 4 & -2 & 9 \end{array} \right)$$

Gebruik x , y en z als variabelen voor dit lineaire systeem
b.v.: $x+2*y+z=1$

Figure 1.115: Lineair systeem, vergelijking invullen

Het correcte antwoord is $7*x-2*y+13*z=-7$.

Als u ervoor zorgt dat de vergelijkingen niet vereenvoudigd kunnen worden, dan kunt u in de *Grading Code* de correcte vergelijking controleren of deze dezelfde is als de respons van de student. Als twee vergelijkingen gelijk aan elkaar zijn, wil dat zeggen dat de linkerleden met elkaar vergeleken worden en de rechterleden ook. Deze uitdrukkingen moeten dan gelijk zijn (volgorde maakt niet uit). Hooguit kan de student zijn vergelijking links en rechts met -1 vermenigvuldigd hebben en daarin voorziet de onderstaande *Grading Code* ook. In de *Grading Code* is hier geprogrammeerd:

```
evalb(($RESPONSE)=$antw)); or evalb(($RESPONSE)=-1*($antw));
```

(Let op de haakjes aan weerszijden van de vergelijking!)

Het wil zeggen dat de linkerleden en de rechterleden van de vergelijking van de student met die van de correcte vergelijking (\$antw) gematcht worden. Het maakt niet uit hoe de student het linkerlid en het rechterlid noteert. Maar als de student iets van het rechterlid naar het linker verhuist of andersom, wordt dat helaas niet als dezelfde vergelijking gezien. Immers de linker en rechter leden komen

dan niet meer overeen. Ook moet de student linker en rechterlid niet omwisselen, want dan ziet het systeem niet dat het dezelfde vergelijkingen zijn.

TIP: Hoe zorgt u ervoor dat de vergelijking niet vereenvoudigbaar is?

De coëfficiënten van de vergelijking mogen dan niet een gemeenschappelijke factor hebben die weggedeeld kan worden. Als deze coëfficiënten vooraf als variabelen gedefinieerd zijn kunt u het volgende doen in het *Algorithm*:

```
$gcd1:=maple("igcd($a,$b,$c,$j)");
condition:=eq($gcd1,1);
```

Het betekent dat de grootste gemene deler van de getallen \$a, \$b, \$c en \$d gelijk moet zijn aan 1.

TIP: Nog beter kunt u de *oplossing* van de vergelijking van de student matchen met de *oplossing* van de correcte vergelijking. U hoeft dan niet zo krampachtig ervoor te zorgen dat de vergelijkingen niet vereenvoudigbaar zijn.

Het is gemakkelijk als de oplossing van de vergelijking uit slechts één element bestaat.

Zijn de oplossingen gelijk, dan zijn immers de vergelijkingen *gelijkwaardig*, en daar gaat het tenslotte om.

Generieker is dan de programmering voor de *Grading Code* van bovenstaand voorbeeld:

```
evalb(solve($RESPONSE,x)=solve($antw,x)) or evalb(solve($RESPONSE,y)=solve($antw,y)) or
evalb(solve($RESPONSE,z)=solve($antw,z)) and type($RESPONSE,equation);
```

In dit geval worden uitdrukkingen gematcht. Immers als x vrijgemaakt wordt uit de vergelijking met x en y en z , dan krijgt u een uitdrukking in de variabelen y en z .

TIP: In bovenstaande is erin voorzien dat wellicht de vergelijking geen x of y of z bevat. Daarom zijn er drie condities meegegeven met `or` ertussen (dus niet `and`).

TIP: Misschien hebt u gezien dat er in de Grading Code ook geprogrammeerd is: `and type($RESPONSE,equation);`

Dat is speciaal gedaan omdat het anders mogelijk is dat bijvoorbeeld de student de vergelijking op 0 herleidt en alleen het linker lid invult. Het commando `solve` werkt bij Maple niet alleen op vergelijkingen maar ook op uitdrukkingen in de veronderstelling dat deze dan gelijk aan 0 gesteld moeten worden.

TIP: Ook is aardig om in de Custom Previewing Code te programmeren:

```
if evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","=")=0) then printf("voer een vergelijking in x en y in.") else
printf(MathML[ExportPresentation]($RESPONSE)) end if;
```

TIP: Let op dat er geen decimale getallen mee gemoeid zijn, want dan zijn de oplossingen niet meer exact gelijk aan elkaar.

Als u met decimael werkt, rond dan de vergelijking van het antwoord én de vergelijking die de student geeft af met een aantal significante cijfers (met `evalf[5]` bijvoorbeeld). Vervolgens kunt u de oplossingen van beide vergelijkingen matchen.

02 Lineair systeem

Gevraagd wordt de vergelijking $34.56x + 45.67y = 1.567$

Voer deze in bij afronden op 3 significante cijfers.

Hierbij maakt het niet uit hoe de vergelijking van het antwoord ingevoerd wordt.
Er wordt gematcht op de oplossing van de vergelijking.

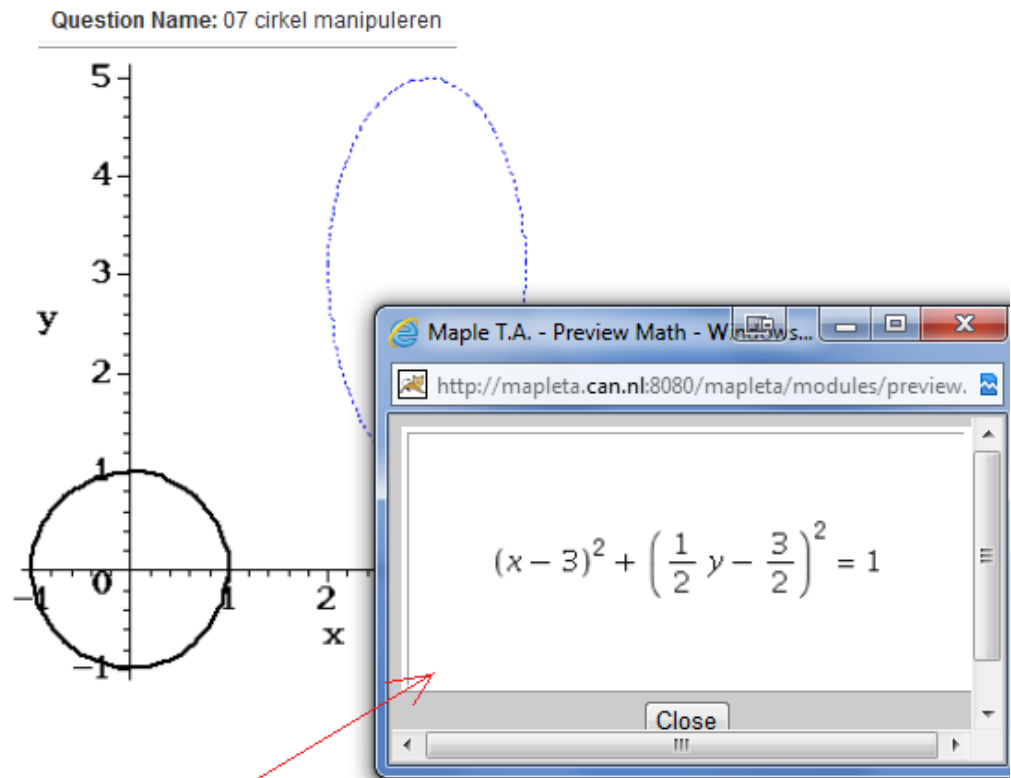
Figure 1.116: Vergelijkingen met decimalen

In de Grading Code kunt u programmeren:

```
evalb(solve(evalf[3](100*$verg),x)=solve(evalf[3](100*$RESPONSE),x)) or
evalb(solve(evalf[3](
100*$verg),y)=solve(evalf[3](100*$RESPONSE
y)))
and type($RESPONSE,equation);
```

Zie ook in paragraaf *Verzamelingen* (page 72) voor nog meer tips bij het matchen van verzamelingen, zoals het geval waarin er decimalen mee gemoeid zijn.

In het volgende voorbeeld zien we dat er een vergelijking gevraagd wordt van een vervormde cirkel.



Bedenk de vergelijking voor de blauwe figuur (met x en y als variabelen) als de grafiek van deze figuur ontstaat uit de eenheidscirkel door deze eerst op te rekken t.o.v. de x -as met factor 2 en daarna 3 naar rechts te verplaatsen en 3 naar boven. Vul de *vergelijking* in met een zelfgekozen vorm zoals je deze ook in Maple zou invoeren zonder decimale getallen en klik op "Plot" om jezelf te controleren voordat je naar de volgende opgave gaat.

$(x-3)^2 + ((y-3)/2)^2 = 1$

This question accepts formulas in Maple syntax.

[Plot](#) | [Help](#) | [Preview](#)

Figure 1.117: Vergelijking van een vervormde cirkel

In dit geval moet de student een kwadratische vergelijking in x en y invullen.

Als nu de oplossing van de vergelijking die de student invoert, gematcht moet worden met de oplossing van de correcte vergelijking, dan zijn er meer uitdrukkingen die als oplossing gelden. Deze exacte oplossingen komen in een rij te staan waarvan de volgorde niet van belang is. Het beste is dan van de oplossing een verzameling te maken en dus in feite twee verzamelingen te matchen:

```
evalb({solve($g,y)}={solve($RESPONSE,y)}) and type($RESPONSE,equation);
```

Het betekent dat van de correcte vergelijking \$g\$ de oplossingsverzameling gemaakt wordt met bijvoorbeeld \$y\$ als onbekende. Dat kan gemakkelijk door er accoladen omheen te zetten. Evenzo wordt ook de oplossingsverzameling van de vergelijking van de student gemaakt met ook weer \$y\$ als onbekende.

Dit kan alleen bij exact oplossen. Vergelijk ook paragraaf *Verzamelingen* (page 72) voor nog meer tips bij het matchen van verzamelingen, zoals het geval waarin er decimalen mee gemoeid zijn.

De oplossing van de vergelijking $(x-3)^2 + \left(\frac{y}{2} - \frac{3}{2}\right)^2 = 1$ is een rijtje van twee uitdrukkingen in \$x\$. Maken we daar een verzameling van, dan kunnen er dus twee verzamelingen gematcht worden. De volgorde van de elementen in een verzameling doet er namelijk niet toe.

$$\left\{3 + 2\sqrt{-8 - x^2 + 6x}, 3 - 2\sqrt{-8 - x^2 + 6x}\right\}$$

TIP: Zie ook in paragraaf *Vergelijkingen* (page 136) om met het vraagtype *Formula* vergelijkingen te overhoren.

TIP: Zie ook in paragraaf *Verzamelingen* (page 72) voor nog meer tips bij het matchen van verzamelingen, zoals het geval waarin er decimalen mee gemoeid zijn.

TIP: Voor het overhoren van de **oplossingen van vergelijkingen** staat er in paragraaf *Ongeordende en geordende lijsten* (page 105) een aantal interessante mogelijkheden.

TIP: Nog een aardigheid om van een vergelijking linker en rechterlid om te wisselen is de volgende:

```
$verg:=maple("p=x+y");
```

```
$verg1:=maple("(rhs = lhs)($verg)");
```

1.4.14 Stelsels vergelijkingen

Bij het oplossen van vergelijkingen kunt u weer Maple inzetten die moeiteloos vrijwel elk stelsel kan oplossen. Er zijn twee manieren om de oplossing te genereren: met verzamelingen en met lijsten. Hieronder een paar voorbeelden om te bekijken wat Maple voor u kan doen.

lees ook nog in het boek *Handleiding Maple17* paragraaf 2.6.7 en 2.6.8 over stelsels vergelijkingen.

```
> restart;verg1:=7*a-11*b=18;verg2:=3*a+4*b=2;
```

$$verg1 := 7a - 11b = 18$$

$$verg2 := 3a + 4b = 2 \quad (1.61)$$

```
> solve({verg1,verg2},{a,b});
```

$$\left\{a = \frac{94}{61}, b = -\frac{40}{61}\right\} \quad (1.62)$$

```
> solve([verg1,verg2],[a,b]);
```

$$\left[\left[a = \frac{94}{61}, b = -\frac{40}{61}\right]\right] \quad (1.63)$$

```
> verg3:=y=x^2+x-2;verg4:=y=4*x+2;
```

$$verg3 := y = x^2 + x - 2$$

$$verg4 := y = 4x + 2 \quad (1.64)$$

```
> solve({verg3,verg4},{x,y});
```

$$\{x = -1, y = -2\}, \{x = 4, y = 18\} \quad (1.65)$$

> solve([verg3,verg4],[x,y]);

[[x = -1,y = -2], [x = 4,y = 18]]

(1.66)

In het bovenstaande ziet u het verschil tussen het gebruik van verzamelingen en lijsten. Let op dat lijsten een vaste volgorde impliceert! In het geval van meer dan één oplossing, krijgt u bij verzamelingen een *rij* van *verzamelingen* en bij lijsten als antwoord een *lijst* bestaande uit *lijsten*.

In het volgende voorbeeld wordt de oplossing gevraagd van een stelsel lineaire vergelijkingen.

We benaderen dit probleem op twee manieren: met verzamelingen en met lijsten.

Question Name: 02b Stelsel lineair verzameling

Los het volgende stelsel vergelijkingen op:

$7a - 8b = 18$

$3a + 4b = 2$

Geef het antwoord in de volgende vorm bijvoorbeeld:
a=3/4, b=5/8
Dus ZONDER af te ronden met de rekenmachine.
Probeer de uitwerking zo goed mogelijk gestructureerd op te schrijven.

Question Name: 02b Stelsel lineair als lijst

Los het volgende stelsel vergelijkingen op:

$7a - 3b = 18$

$3a + 4b = 2$

Geef het antwoord in de volgende vorm bijvoorbeeld:
[a=3/4, b=5/8]
Dus ZONDER af te ronden met de rekenmachine.
Probeer de uitwerking zo goed mogelijk gestructureerd op te schrijven.

Figure 1.118: Stelsel vergelijkingen

De bijbehorende *Algorithms* zijn hieronder te zien:

```
$a=range(2,13);
$v1=maple("7*a-$a*b=18");
$v2=maple("3*a+4*b=2");
$displayv1=maple("MathML[ExportPresentation]($v1)");
$displayv2=maple("MathML[ExportPresentation]($v2)");
$opl=maple("solve({$v1,$v2},{a,b})");
$displayopl1=maple("MathML[ExportPresentation]($opl[1])");
$displayopl2=maple("MathML[ExportPresentation]($opl[2])");
```

| Variable | Value |
|-------------|-------------------------|
| a | 11 |
| v1 | $7a - 11b = 18$ |
| v2 | $3a + 4b = 2$ |
| displayv1 | $7a - 11b = 18$ |
| displayv2 | $3a + 4b = 2$ |
| opl | {a = 94/61, b = -40/61} |
| displayopl1 | $a = \frac{94}{61}$ |
| displayopl2 | $b = -\frac{40}{61}$ |

```
$a=range(2,13);
$v1=maple("7*a-$a*b=18");
$v2=maple("3*a+4*b=2");
$displayv1=maple("MathML[ExportPresentation]($v1)");
$displayv2=maple("MathML[ExportPresentation]($v2)");
$opl=maple("solve([$v1,$v2],[a,b])");
$displayopl=maple("MathML[ExportPresentation]($opl[1])");
```

| Variable | Value |
|------------|--|
| a | 12 |
| v1 | $7a - 12b = 18$ |
| v2 | $3a + 4b = 2$ |
| displayv1 | $7a - 12b = 18$ |
| displayv2 | $3a + 4b = 2$ |
| opl | [[a = 3/2, b = -5/8]] |
| displayopl | $\left[a = \frac{3}{2}, b = -\frac{5}{8} \right]$ |

Figure 1.119: Stelsel vergelijkingen met verzameling en met lijst

De bijbehorende *Grading Code*:

Voor de verzameling: `evalb(($opl) = { $RESPONSE });`

We zien dat de oplossing \$opl een verzameling is en deze matchen we met die van de student door accolades eromheen te plaatsen. Het voordeel is dat de student geen haakjes of accolades hoeft te plaatsen en de volgorde van intikken is ook voor de student niet belangrijk. Net zo goed had de student de variabelen a en b kunnen omwisselen.

TIP: Let op dat u in de rubriek *Answer* niet de verzameling presenteert met de accolades. De student zou bij het zien van het correcte antwoord kunnen denken dat hij accolades had moeten invoeren. Werk eventueel met operanden van de verzameling. (Zie paragraaf (page 69).)

Voor de lijst is de *Grading Code*: `evalb(($opl[1]) = $RESPONSE);`

We vergelijken hier de correcte lijst (de eerste lijst van de lijst) met de lijst van de student. Het nadeel van een lijst is, dat de volgorde van deze lijst vast ligt. De student dient dus eerst a en dan b af te handelen en haakjes te plaatsen.

TIP: Zorg ervoor dat u binnen een toets wat eenheid hebt in de manier van invoeren van het antwoord. Anders moet de student steeds verschillende manieren van invoer hanteren en dat kan verwarrend zijn.

Eventueel kunt u ook de afzonderlijke variabelen bevragen, zodat er meer mogelijkheden zijn voor deelscores:

Question Name: 02b Stelsel lineair variabelen apart

Los het volgende stelsel vergelijkingen op en rond eventueel af op 3 decimalen:

$$7a - 10b = 18$$

$$3a + 4b = 2$$

$a =$

$b =$

```

$a=range(2,13);
$v1=maple("7*a-$a*b=18");
$v2=maple("3*a+4*b=2");
$displayv1=maple("MathML[ExportPresentation]($v1)");
$displayv2=maple("MathML[ExportPresentation]($v2)");
$opl=maple("solve([$v1,$v2],[a,b])");
$opla=maple("rhs($opl[1,1])");
$oplb=maple("rhs($opl[1,2])");

```

| Variable | Value |
|-----------|-------------------------|
| a | 12 |
| v1 | $7a - 12b = 18$ |
| v2 | $3a + 4b = 2$ |
| displayv1 | $7a - 12b = 18$ |
| displayv2 | $3a + 4b = 2$ |
| opl | $[[a = 3/2, b = -5/8]]$ |
| opla | $3/2$ |
| oplb | $-5/8$ |

Figure 1.120: Stelsel vergelijkingen met de variabelen apart

Maak dan weer gebruik van lijsten bij het commando `solve`, omdat u er dan zeker van bent dat de volgorde van de variabelen vast ligt. Met het aanroepen van bijvoorbeeld `rhs($opl[1,2])` krijgt u dan het rechterlid van het tweede element van de eerste lijst van de lijst. Immers de oplossing wordt gegenereerd als een lijst met daarin lijsten (hier slechts eentje).

TIP: In dit laatste voorbeeld kunt u dus numerieke invulvelden maken en eventueel afrondingen toelaten als u dat wilt. U kunt ook kiezen voor een *Formula*-veld en het exacte antwoord afdwingen.

In het volgende voorbeeld komen er meer oplossingen van een stelsel vergelijkingen. Er wordt gevraagd naar de coördinaten van de snijpunten van twee grafieken.

Question Name: 02c stelsel niet lineair

Bepaal de snijpunten van de grafieken van de volgende twee functies:

$$f(x) = x^2 + x - 2$$

$$g(x) = -2x + 2$$

Geef het antwoord in de vorm bijvoorbeeld:

[x=20 , y=30] , [x=3,y=5]

zonder af te ronden met de rekenmachine.

Figure 1.121: Stelsel met meer oplossingen

Het bijbehorende *Algorithm* is het volgende:

```
$a=range(-2,4,3);
$f=maple("y=x^2+x-2");
$g=maple("y=$a*x+2");
$displayf=maple("MathML[ExportPresentation](f(x)=rhs($f))");
$displayg=maple("MathML[ExportPresentation](g(x)=rhs($g))");
$opl=maple("solve([$f,$g],[x,y])");
```

| Variable | Value |
|----------|-------------------------------------|
| a | 1 |
| f | $y = x^2 + x - 2$ |
| g | $y = x + 2$ |
| displayf | $f(x) = x^2 + x - 2$ |
| displayg | $g(x) = x + 2$ |
| opl | $[[x = 2, y = 4], [x = -2, y = 0]]$ |

Figure 1.122: Stelsel vergelijkingen met meer oplossingen Algorithm

Met de *Grading Code* van het invulveld is het een kwestie van verzamelingen bestaande uit lijstjes matchen, zodat de volgorde van de lijstjes er niet toe doet:

```
evalb( { $opl[1] , $opl[2] } = { $RESPONSE } );
```

Er is voor gezorgd dat het mooi uitkomt, zodat de student niet in de verleiding komt om numerieke benaderingen toe te passen.

Bent u van plan om toch numerieke benadering toe te laten, dan is de volgende manier van doen een optie.

Question Name: 19e ongelijkheid dubbel kwadratisch

Bepaal de coördinaten van de snijpunten van de grafieken van de volgende twee functies:

$$f(x) = x^2 + 5x + 3$$

$$g(x) = -2x^2 + 2x + 8$$

Geef het antwoord in de vorm bijvoorbeeld:

[2.56,14.8] , [16.34,30.45]

Rond zo nodig af op minstens 2 decimalen.

[x,y]=  

Figure 1.123: Stelsel vergelijkingen met meer oplossingen numeriek

In dit vraagstuk zijn de coördinaten van de snijpunten van de twee grafieken niet meer getallen die mooi uitkomen. Het ligt voor de hand dat er nu numerieke benadering moet worden toegepast. U kunt natuurlijk vragen naar de x -waarden en daarvoor twee afzonderlijke numerieke invulvelden maken, maar het kan ook in één keer zoals hier gedaan is. Omdat dit vraagstuk onderdeel is van een groter geheel, is het in dit geval niet erg als de student tools zoals grafische rekemachine of computeralgebra inzet voor deze berekening. (In een vervolgvraag wordt er namelijk gevraagd naar de intervallen waarin de ene functie groter is dan de ander en daarbij is het antwoord van deze eerste vraag noodzakelijk.)

Het *Algorithm* is als volgt:

```
$a=range(2,5);
$b=range(1,2);
$f=maple("y=x^2+$a*x+3");
$f1=maple("y=x^2+$a*x+3.0");
$g=maple("y=-$b*x^2+2*x+8");
$displayf=maple("MathML[ExportPresentation](f(x)=rhs($f))");
$displayg=maple("MathML[ExportPresentation](g(x)=rhs($g))");
$opl=maple("solve([$f1,$g],[x,y])");
$opl1=maple("map(x->rhs(x),$opl[1])");
$opl2=maple("map(x->rhs(x),$opl[2])");
```

| Variable | Value |
|----------|---|
| a | 3 |
| b | 1 |
| f | $y = x^2 + 3x + 3$ |
| f1 | $y = x^2 + 3x + 3.0$ |
| g | $y = -x^2 + 2x + 8$ |
| displayf | $f(x) = x^2 + 3x + 3$ |
| displayg | $g(x) = -x^2 + 2x + 8$ |
| opl | $[[x = 1.350781059, y = 8.876952648], [x = -1.850781059, y = .8730473516]]$ |
| opl1 | $[1.350781059, 8.876952648]$ |
| opl2 | $[-1.850781059, .8730473516]$ |

Figure 1.124: Algorithm van een vraag over het berekenen van snijpunten van grafieken

In bovenstaand *Algorithm* zien we dat er behalve de variabele \$f ook een variabele \$f1 gedefinieerd is. Dat is met opzet zo gedaan, want als bij het oplossen van het stelsel vergelijkingen ergens een decimaal getal voorkomt, dan gaat Maple ogenblikkelijk over op numerieke rekenwijze en krijgen we de oplossing ook in de vorm van decimale getallen, wat we hier ook willen. Omdat er in de opgave naar de coördinaten van de snijpunten gevraagd wordt bereiden we die voor met de variabelen \$opl1 en \$opl2. Van de lijstjes wordt dan steeds het rechterlid van de vergelijkingen genomen en de volgorde blijft gehandhaafd.

In de *Grading Code* zal de verzameling van de twee lijstjes gematcht worden met de verzameling van de twee lijstjes die de student invult. Echter we willen dan geen lijstjes met decimale getallen met elkaar vergelijken en daarom vermenigvuldigen we de elementen van die lijstjes met 100 en ronden daarna af.

```
lijst:=[ $RESPONSE]: evalb( {map(round,100*$opl1),map(round,100*
$opl2)}={map(round,100*lijst[1]),map(round,100*lijst[2])} );
```

Doordat we *verzamelingen* met elkaar vergelijken is de volgorde van de lijstjes die de student invult niet belangrijk.

TIP: Het maakt niet uit als de student meer dan 2 decimalen zou invullen, want het antwoord van de student wordt toch vermenigvuldigd met 100 en dan afgerond. Minder dan 2 decimalen zal niet goedgerekend worden, tenzij de tweede decimaal een 0 zou zijn.

TIP: in de rubriek *Answer* kan dan bijvoorbeeld ingevuld worden:

```
evalf[3]($opl1),evalf[3]($opl2);
```


De student heeft dan in ieder geval het goede antwoord bij de hand in geval hij deze vraag fout beantwoord had.

1.4.15 Ongeordende en geordende lijsten

Als er vergelijkingen opgelost moeten worden met meer dan één oplossing, dan kan een geordende of ongeordende lijst overhoord worden met het *Maple-graded* vraagtype en dan kunt u de oplossing van de vergelijking matchen met de correcte oplossing eventueel met het matchen van verzamelingen of desnoods met aparte invulvelden.

Maar let op! Als u met verzamelingen werkt, kan het alleen maar met exacte antwoorden en als er decimale getallen in de vergelijking voorkomen, dan werkt het niet met verzamelingen. Neem dan de truc met het vermenigvuldigen en afronden ter harte die in paragraaf *Verzamelingen* (page 72) genoemd is.

In het volgende voorbeeld gaat het over het exact oplossen van een kwadratische vergelijking.

Question Name: 13 kwadratische vergelijking half goed

Bepaal de oplossingen van de kwadratische vergelijking:

$$(5z - 3)^2 = 64$$

Geef de antwoorden gescheiden door een komma en vul alleen getallen in zonder af te ronden met de rekenmachine.

Een alternatief om ook half goed te kunnen rekenen voor twee antwoorden is apart te bevragen, maar dan moet de volgorde van te voren wel vastgelegd zijn.

Geef de antwoorden op volgorde van klein naar groot.





Kleinste antwoord =   en het grootste antwoord is  

Figure 1.125: title of the figure

In bovenstaande figuur zien we dat het eerste invulveld van het type *Maple-graded* is.

De volgende invulvelden zijn van het type *Formula*, waar alleen maar een getal ingevuld hoeft te worden. (Had ook met een numeriek veld gekund.)

In de *Grading Code* van het eerste invulveld kan zelfs geregeld worden dat bijvoorbeeld een gedeelte goed gerekend wordt.

Het *Algorithm* van deze vraag is als volgt.

```
$a=range(1,9);
$b=$a^2;
$c=switch(rint(2),range(-9,-1),range(1,9));
$d=range(2,6);
$vraag=maple("($d*z-($c))^2=$b");
$displayvraag=maple("printf(MathML[ExportPresentation]($vraag))");
$sopl=maple("solve($vraag,z)");
$Antw1=maple("[solve($vraag,z)][1]");
```

```
$Antw2:=maple("[solve($vraag,z)][2]");
$antw2:=maple("max([$Antw1,$Antw2])");
$antw1:=maple("min([$Antw1,$Antw2])");
```

In dit geval zijn er twee antwoorden voor de oplossing van de vergelijking. Deze antwoorden kunnen dus ook afzonderlijk overhoord worden in twee verschillende invulvelden van bijvoorbeeld het type *Numeric*. Op die manier kunt u ook gemakkelijk decimale getallen toelaten met eventueel tolerantie. In het *Algorithm* zijn de twee antwoorden hier apart voorbereid (\$antw1 en \$antw2). De grootste met het Maple-commando max en de kleinste met het Maple-commando min. In het eerste invulveld worden beide antwoorden gevraagd en daar is de volgorde niet van belang. De *Grading Code* is dan als volgt:

```
aantal:=nops({$RESPONSE} intersect {$opl}): if StringTools[Search]("solve", "$RESPONSE")>0
then 0 else evalf(aantal/2) end;
```

In deze code wordt van het antwoord van de student een verzameling gemaakt door er accolades omheen te zetten. Het aantal elementen van de doorsnede (intersect) van de verzameling van de student en de correcte verzameling wordt geteld. Als resultaat geeft dit aantal gedeel door 2 dan een getal tussen 0 en 1 wat overeenkomt met de waardering voor deze vraag als er bijvoorbeeld twee antwoorden waren. Maar dan mag er natuurlijk niet het Maple-commando "solve" in het antwoord van de student staan.

TIP: U moet er natuurlijk wel van verzekerd zijn dat het correcte antwoord beslist uit twee elementen bestaat, maar daarin kan voorzien worden bij de keuze van de variabelen in het *Algorithm*.

TIP: Geef voor het eerste antwoordveld geen instelling met *Formula*, want dan krijgt de student in de *Preview* haakjes om het antwoord te zien en dat moet niet. Echter als u de instelling van *Maple syntax* met *text entry only* hanteert, dan kan de student de Maple-opdracht "solve" gebruiken in zijn antwoord. In dat geval moet u nog extra programmeren met StringTools, dat "solve" niet in het antwoord mag voorkomen, zoals hier boven. Als u dat niet wilt programmeren, kunt u uitwijken naar de instelling *Symbol entry only* en dan volstaat de volgende *Grading Code*:

```
aantal:=nops({$RESPONSE} intersect {$opl}): evalf(aantal/2) ;
```

TIP: Als het gaat om decimale getallen, converteer dan de verzamelingen eerst naar verzamelingen met echte breuken, want dan gaat het matchen beter. Eventueel voorafgegaan door een afronding met evalf[#].

```
evalb(convert({$roots},rational)=convert({$RESPONSE},rational)) and
evalb(StringTools[Search]("solve", "$RESPONSE")=0) ;
```

Hierin is de variabele \$roots als rij van oplossingen reeds in het *Algorithm* voorbereid. Hiervan wordt dan een verzameling gemaakt en de getallen worden geconverteerd naar rationale getallen.

Als alternatief kunt u de elementen van de verzameling bijvoorbeeld met 100 vermenigvuldigen en daarna afronden, zoals behandeld in paragraaf *Verzamelingen* (page 72).

TIP: Kijk ook eens in paragraaf *Lijsten en Vectoren met Formula* (page 134) voor uitgebreidere informatie over geordende en ongeordende lijsten.

TIP: voor meer informatie over lijsten, zie in de handleiding *Randomiseren* waarin een aparte paragraaf over lijsten.

TIP: voor gedeeltelijk goedrekenen.

```
if evalb(StringTools[Search]("solve", "$RESPONSE")=0) then
```

```
`if (member($ant1,{$RESPONSE}),0.3,0)
```

```
+
```

```
`if (member($ant2,{$RESPONSE}),0.3,0)
```

```
+
```

```
`if (member($ant3,{$RESPONSE}),0.4,0)
```

```
else 0 end if;
```

1.4.16 Ongelijkheden

Ongelijkheden testen vereist enige kennis van het computeralgebrasysteem Maple omdat we toch daaraan veel zullen uitbesteden.

De student krijgt hier bijvoorbeeld een waardering 1 voor het teken en 2 voor het getal dat hij moet invullen.
Het *Algorithm* is als volgt opgesteld:

```
$a:=range(1,7);
$b:=switch(rint(4),2,4,5,6);
$ongleft=mathml("(3*x-$a)/3");
$ongright=mathml("(x-8)/$b");
$index:=rint(2);
$teken:=switch($index,">","<");
$tekenfout:=switch($index,"<",">");
$rechtslinks:=switch($index,"lhs","rhs");
$ongl:=maple("(3*x-$a)/3 $teken (x-8)/$b");
$opl:=maple("solve($ongl,{x})");
$Getal:=maple("$rechtslinks($opl[1])");
```

Bij het definiëren van het linkerlid en het rechterlid van de ongelijkheid is rekening gehouden met het feit dat er geen omslag van het teken plaatsvindt tijdens het oplossen van de ongelijkheid.

Het definiëren van het linker en rechter lid als mathml-code brengt met zich mee dat de ongelijkheid in niet-vereenvoudigde vorm aangeboden kan worden met \$ongleft \$teken \$ongright.

Afhankelijk van het teken geeft Maple de oplossing weer als $\{getal < x\}$ of als $\{x < getal\}$ omdat Maple altijd het kleinerdan-teken hanteert, vandaar de definitie van \$Getal. Hoewel dat ook op een andere manier opgelost kan worden door niet de ongelijkheid op te lossen, maar de vergelijking met: $\$getal:=maple("solve((3*x-\$a)/3 = (x-8)/\$b,x)");$

De variabele \$teken, kan niet alleen gebruikt worden bij het presenteren van de opgave, maar ook bij het definiëren van het goede antwoord. Het foute antwoord voor het teken in het invulveld van het type List, wordt gedefinieerd als \$tekenfout.

Bij het definiëren van het numerieke antwoordveld wordt gematcht met de variabele \$Getal en geopteerd voor *Accept arithmetic* zodat de student ook het exacte antwoord kan invullen als hij niet afrondt. De tolerantie wordt op 0.01 gezet voor het geval de student op twee decimalen gaat afronden.

Het volgende voorbeeld overhoort de oplossing van een ongelijkheid in zijn geheel.

Question Name: 19 ongelijkheid simpel

Geef de oplossing van de volgende ongelijkheid in de meest simpele vorm en gebruik geen decimale getallen.

$$6x + 1 > -2$$



Figure 1.127: Simpele ongelijkheid

De student moet dan bijvoorbeeld iets als "x>-1/2" invullen wat door Maple opgevat wordt als "-1/2<x". Maple ziet dus van de ongelijkheid "x>-1/2" het linker lid als -1/2 en het rechterlid als x. De student had dus net zo goed ook "-1/2<x" kunnen invullen. De correcte oplossing en die van de student worden met elkaar vergeleken doormiddel van het matchen van de rechterleden en de linkerleden afzonderlijk.

De oplossing van de student wordt altijd vertaald naar een ongelijkheid met het kleinerdan-teken.

Het *Algorithm* van deze opgave is als volgt:

```

$a=switch(rint(2),range(-6,-1),range(1,6));
$b=switch(rint(2),range(-6,-1),range(1,6));
$c=switch(rint(2),range(-6,-1),range(1,6));
$teken=switch(rint(2),"<",">");
$ong=maple("($a)*x+($b) $teken ($c)");
$displayongl=maple("printf(MathML[ExportPresentation](($a)*x + ($b)))");
$displayongr=mathml("$c");
$opl=maple("solve($ong,{x})");
$displayopl=maple("printf(MathML[ExportPresentation]($opl[1]) )");

```

| Variable | Value |
|-------------|-------------|
| a | 4 |
| b | -3 |
| c | 5 |
| teken | < |
| ong | $4x < 8$ |
| displayongl | $4x - 3$ |
| displayongr | 5 |
| opl | $\{x < 2\}$ |
| displayopl | $x < 2$ |

Figure 1.128: title of the figure

TIP: Het linker en rechterlid worden onafhankelijk van elkaar gedefinieerd in het *Algorithm* om de presentatie van de ongelijkheid in de vraag goed te krijgen.

TIP: Let op dat de oplossing \$opl als verzameling gegeven wordt. De grading kunt u doen met de volgende *Grading Code*:

```
evalb( ($opl[1])=($RESPONSE) );
```

Let op de haakes om de ongelijkheid zoals (\$RESPONSE). En let ook op dat u matcht met \$opl[1], dus het eerste element van de oplossing \$opl die als verzameling gegeven wordt. In de rubriek *Answer* moet dan ook \$opl[1] staan.

U kunt ook de volgende *Grading Code* gebruiken:

```
evalb( $opl={$RESPONSE} );
```

of als u wel decimale getallen toestaat:

```
evalb(convert( {$RESPONSE},rational)=convert( $opl,rational) );
```

Het volgende voorbeeld overhoort de oplossing van een ongelijkheid ook weer in zijn geheel. In de *Grading Code* wordt dan gebruikgemaakt van het matchen van verzamelingen.

Question Name: 19c ongelijkheid 3

Los de volgende ongelijkheid op:

$$0 < x^2 - 11$$

Geef het antwoord in de vorm {x<...},{x>...} en rond af op 2 decimalen.

De oplossing is

Figure 1.129: Kwadratische ongelijkheid

Bij deze vraagstelling moet de student een verzameling of een rijtje verzamelingen invullen. De aanwijzing hoe iets ingevuld moet worden is daarvan afhankelijk. Er is hier gekozen voor een numeriek antwoord en dat ligt gevoelig bij het matchen van verzamelingen, vergelijkingen en ongelijkheden.

In het *Algorithm* bereiden we een en ander voor:

```
$index=rint(2);
$gk=switch($index,">","<");
$a=range(2,17);
$ong=maple("x^2-$a $gk 0");
$displayOng=maple("printf(MathML[ExportPresentation]($ong))");
$opl=maple("evalf[3]({solve($ong,{x}))");
$oplossing=maple("if $index=0 then $opl[1],$opl[2] else $opl[1] end");
$aanwg="Geef het antwoord in de vorm {x<...},{x>...}";
$aanwk="Geef het antwoord in de vorm {x<...,x>...}";
$aanwijzing=switch($index,"$aanwg","$aanwk");
```

| Variable | Value |
|------------|--|
| index | 0 |
| gk | > |
| a | 11 |
| ong | $0 < x^2 - 11$ |
| displayOng | $0 < x^2 - 11$ |
| opl | $\{3.32 < x\}, \{x < -3.32\}$ |
| oplossing | $\{3.32 < x\}, \{x < -3.32\}$ |
| aanwg | Geef het antwoord in de vorm {x<...},{x>...} |
| aanwk | Geef het antwoord in de vorm {x<...,x>...} |
| aanwijzing | Geef het antwoord in de vorm {x<...},{x>...} |

Figure 1.130: Oplossing van ongelijkheid met verzamelingen matches

Met de variabele \$gk kunt u variëren tussen groter-en kleinerdan-teken.
Met de variabele \$displayOng laten we het hier aan Maple over hoe de presentatie van de ongelijkheid wordt vormgegeven. Een alternatief is het werken met mathml en linker en rechterlid apart te definiëren zoals in het voorbeeld van Figure 1.126 (page 107) gedaan is.

Vervolgens wordt de oplossing gegenereerd en afgerond op 3 significante cijfers met `evalf[3](...)`. De range van het getal `$a` is zó gekozen dat dit neerkomt op twee cijfers achter de decimale punt. Dit resulteert in een oplossing die de vorm heeft van een verzameling bestaande uit meer verzamelingen óf één enkelvoudige verzameling, zoals dat ook bij de inleiding van deze paragraaf vermeld is. De student krijgt een aanwijzingen hoe het antwoord ingevuld moet worden (afhankelijk van de situatie). De variabele `$oplossing` is die welke de student dient in te vullen en kan bij het invulveld bij *Answer* worden gebruikt. Het zal duidelijk zijn dat hier tolerantie in het numerieke antwoord niet mogelijk is.

De *Grading Code* eist hier nog even aandacht omdat bij het matchen van verzamelingen het erg nauw steekt met getallen. Bijvoorbeeld de verzamelingen $\{x > 4.0\}$ en $\{x > 4\}$ worden niet als dezelfde gezien. Vandaar dat de hele verzameling in één keer geconverteerd wordt naar rationale getallen. Dat kan met `convert(...,rational)` in één keer zonder map te hoeven gebruiken. De *Grading Code* is dan de volgende:

```
evalb(convert({$RESPONSE},rational)=convert($opl,rational));
```

Let op dat `$opl` dus een verzameling is bestaande uit één of twee verzamelingen met afgeronde oplossingen. Het antwoord van de student moet ook nog die vorm hebben `{$RESPONSE}` om te kunnen matchen. De student moet ook beslist afronden en tolerantie in het antwoord is niet mogelijk.

In het volgende voorbeeld wordt er gewerkt met intervallen om de oplossing van een ongelijkheid weer te geven. Er kan dan gewerkt worden met multiple choice, adaptief voorafgegaan door een berekening van de snijpunten van twee grafieken. Vervolgens kan de student met het correcte antwoord de oplossing van de ongelijkheid vertalen naar het correcte interval van de variabele.

Question Name: 19d ongelijkheid met intervallen



Bepaal de snijpunten van de grafieken van de volgende twee functies:

$$f(x) = x^2 + x - 2$$

$$g(x) = -2x + 2$$

Geef het antwoord in de vorm bijvoorbeeld:

$[x=20, y=30], [x=3, y=5]$

zonder af te ronden met de rekenmachine.

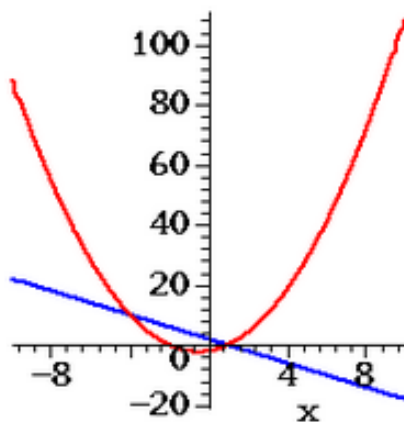
$[x=1, y=0], [x=-5, y=10]$



Correct response: $[x = -4, y = 10], [x = 1, y = 0]$

Voor welke waarden van x is $f(x)$ groter dan $g(x)$?

- ☐ $[-4, 1]$
- ☐ $[1, \rightarrow)$
- ☐ $(-4, 1)$
- ☐ $(\leftarrow, -4]$
- ☐ $(\leftarrow, -4]$ en $[1, \rightarrow)$
- ☐ $(\leftarrow, -4)$ en $(1, \rightarrow)$



Attempt 1 of 1

Verify

Figure 1.131: Ongelijkheid met intervallen

In het eerste deel van de vraag wordt gevraagd naar de snijpunten van twee grafieken. Dat komt neer op het oplossen van een stelsel vergelijkingen. Zie daarvoor in paragraaf *Stelsels Vergelijkingen* (page 99).

Het tweede deel van de vraag is in de vorm van multiple choice. De student krijgt het antwoord van de eerste vraag in ieder geval te zien, mocht dat niet goed zijn geweest en op deze adaptieve manier kan de student toch laten zien dat hij de oplossing van de ongelijkheid kan vertalen naar intervalnotatie.

In dit geval zijn de alternatieven voor de multiple choice-vraag in het *Algorithm* voorbereid in de vorm van de variabelen \$test1 t/m \$test6 waarvan de laatste dus de juiste keuze is.

```
$a=range(-2,4,3);
$f=maple("y=x^2+x-2");
$g=maple("y=$a*x+2");
$oplossing=maple("solve(rhs($f)=rhs($g),x)");
$oplmin=maple("min([$oplossing])");
$oplmax=maple("max([$oplossing])");
$plot=plotmaple("plot([rhs($f),rhs($g)],x=-10..10,color=[red,blue],thickness=2,plotoptions='height=250, width=250')");
$test1="[$oplmin,$oplmax]";

$test2="#←,$oplmin";

$test3="[$oplmax,→#";

$test4="#$oplmin,$oplmax#";

$test5="#←,$oplmin] en [$oplmax,→#";

$test6="#←,$oplmin# en #$oplmax,→#";
```

De variabelen \$test kunnen aangeroepen worden bij het maken van de multiple-choce-vraag echter deze html-code (tekst) kan ook direct ingevuld worden bij de alternatieven van het multiple-choice-invulveld.

Het pijltje naar rechts maakt u met `&rarr`; en het pijltje naar links met `&larr`;

De linker en rechter haken voor open interval # en # maakt u met respectievelijk de html-codes `&lang`; en `&rang`;

(Bij *Refresch* ziet u dat deze html-code direct veranderd is in het pijltje en de haken en eventueel kunt u dan nog de *Returns* weghalen in de code.)

Zie voor het maken van de dynamische figuur in paragraaf *Dynamische figuren* (page 138).

1.4.17 Griekse letters

Bij de *Maple graded*-vragen is het mogelijk om ook formules met Griekse letters te overhoren.

Question Name: greek

test!

1) Vul in $\frac{1}{2} d(\alpha + \beta)^2$

Hier is geprogrammeerd dat eventueel de spatie achter de d weggelaten mag worden. De grie worden. Rechter muisklikken voor de paletten. In principe altijd spaties of een * tikken tusser Anders worden lettercombinaties als één variabele gezien.

Equation Editor
[Help](#)

Figure 1.132: Griekse letters invoeren

Het antwoord en ook de vraag worden voorbereid in het *Algorithm* zoals in dit voorbeeld hieronder te zien is.

Er zijn twee goede antwoorden geprogrammeerd en de Grading Code kan dan zijn:

```
is(($antw1)-($RESPONSE) = 0) or is(($antw1a)-($RESPONSE) = 0);
```

Het beste is dan om voor de instelling *Maple syntax Symbol entry only* te kiezen waarmee de student dan een Editor aangeboden krijgt en de Griekse letters aan kan klikken. (In versie 9 ook enkele hoofdletters.)

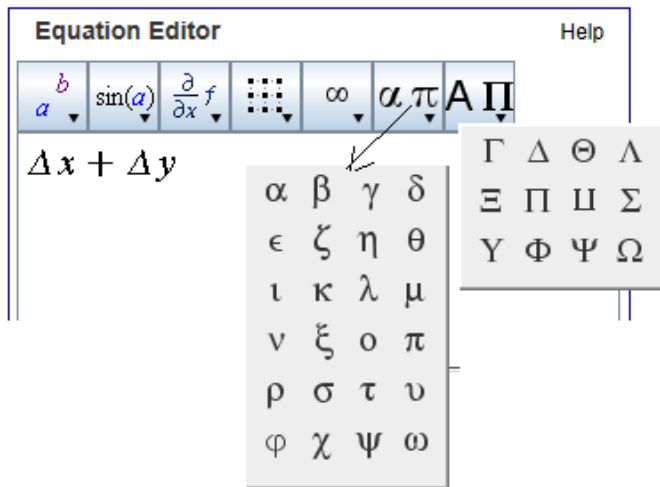


Figure 1.133: Griekse letters invoegen in het antwoordveld

In het *Algorithm* worden de formules als volgt voorbereid:

```
$antw1=maple("d(alpha+beta)^2/2!");
$antw1a=maple("d*(alpha+beta)^2/2!");
$displayantw1=maple("printf(MathML[ExportPresentation]($antw1))");
```

TIP: Eventueel kunnen de instellingen ook op *text entry only* gezet worden, maar dan moet de student weten hoe de Griekse letters ingetikt moeten worden zoals alpha en theta en Pi en dergelijke. Echter in dit soort situaties is het aanbieden van een Editor wat minder riskant.

TIP: Let we op dat als in de formule een ζ of een π gehanteerd worden dat deze letters met begin-hoofletter genoteerd worden: Zeta en Pi. Maple kent namelijk aan deze letters bepaalde functies of waarden toe.

TIP: De letter epsilon doet het niet! MapleSoft is daarover reeds benaderd.

TIP: De Griekse hoofdletters kunt u in de formules voorbereiden met bijvoorbeeld DELTA en THETA en SIGMA bijvoorbeeld.

TIP: Als u Δx wilt overhoren, kan dat alleen maar als u het goede antwoord programmeert met Delta*x want de combinatie Deltax wordt niet gezien als een Griekse letter met daarachter een x. Het ziet er dan uit als Δx . Dit is ook iets wat de studenten moeten trainen.

1.4.18 Differentialen

Voor het overhoren van differentialen en integralen is het vaak handig om de *Editor* ter beschikking te stellen. Echter niet altijd, zoals we straks zullen zien (Figure 1.135 (page 116) waar uitgelegd wordt dat symbolische differentialen niet met de Editor overhoord kunnen worden).

Kennis van enige syntax voor student als ook voor de item-makers is hierbij wel vereist.

TIP: Als u de afgeleide van een functie of de integraal overhoort waarbij de student deze moet berekenen, hou er dan rekening mee dat de student ook gebruik kan maken van de Maple-commando's "diff" en "int" en dat het antwoord dan goedgerekend wordt. Dat

geldt niet alleen bij de instellingen *Text entry only* maar ook voor het gebruik van de Editor. Als de student een differentiaal of een integraal uit het palet neemt is de response namelijk de waarde van de differentiaal of de waarde van de integraal. Met StringTools is de string van de response van de student te controleren op gebruik van deze commando's met bijvoorbeeld de *Grading Code*:

```
evalb(StringTools[Search]("diff", "$RESPONSE")=0);
```

Dit geldt dus zowel voor de instellingen voor *Text mode* als ook *Symbol mode*.

Als de student een differentiaalvergelijking moet intoetsen, stel dan de *Editor* beschikbaar met de instellingen *Maple syntax* met *Symbol entry only*. De student kan dan beschikken over de operator $\frac{d}{dx}$.

Equation Editor



Figure 1.134: De student tikt een differentiaalvergelijking in

De *Grading Code* zou kunnen zijn:

```
evalb(dsolve($RESPONSE,v(t),explicit)=dsolve($answ1,v(t),explicit));
```

Waarmee dus de expliciete oplossing van de DV die de student intikt (\$RESPONSE) gematcht wordt met de expliciete oplossing van het correcte antwoord (\$answ1).

Het *Algorithm* van deze vraag is als volgt:

```
$answ1=maple("diff(v(t),t)=k/m*v(t)");
$answ2=maple("m*diff(v(t),t)=k*v(t)");
$displayansw1=maple("printf(MathML[ExportPresentation]($answ1))");
$displayansw2=maple("printf(MathML[ExportPresentation]($answ2))");
```

Dit is een vrij eenvoudige vraag en de differentiaalvergelijking kan op meer manieren genoteerd worden, maar dat is niet belangrijk, want in de *Grading Code* worden toch de oplossingen ervan gematcht.

De laatste regels in het *Algorithm* zijn ten behoeve van de rubriek *Feedback*.

TIP: Met de *Editor* kan uitstekend de differentiaalvergelijking overhoord worden. De functie moet dan wel beslist genoteerd worden met zijn onafhankelijke variabelen erbij. Dus hier is dat $v(t)$. Alleen met v volstaan, zal niet lukken.

Met de *Editor* heeft het namelijk geen zin om uitdrukkingen te overhoren zoals $\frac{d}{dx} y$. Immers deze uitdrukking heeft de betekenis 0 omdat y gezien wordt als onafhankelijk van x . Evenzo $\text{diff}(v,t)$.

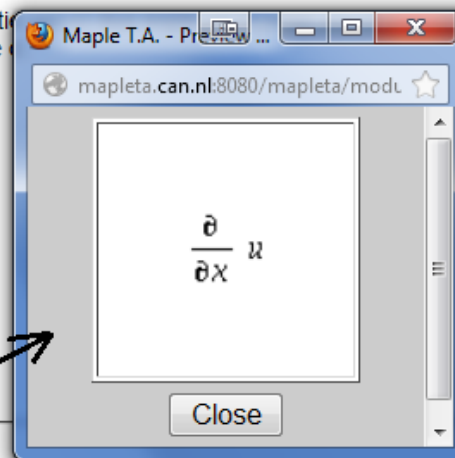
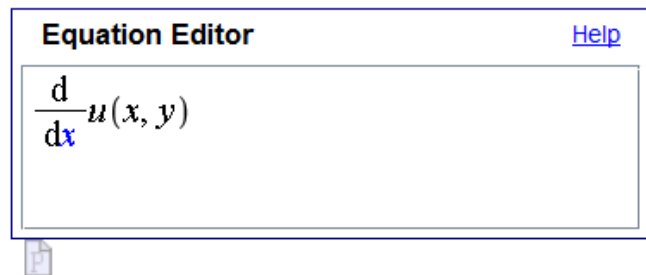
TIP: Neem eventueel uw toevlucht tot de instelling *Text entry only* als u symbolische differentiaal wilt overhoren zoals $\frac{\partial}{\partial x} y$ en train de student in het intikken van de inerte vorm $\text{Diff}(y,x)$ die gemakkelijk te graden is met $\text{is}((\$antw)-(\$RESPONSE)=0);$

als het antwoord voorgeprogrammeerd is als $\$antw=\text{maple}("Diff(u,x)");$

In de *Preview* van het tekstveld komt dan ook de partiële afgeleide te staan met de kromme ∂ zoals in de volgende figuur te zien is bij het tweede antwoordveld.

2) Vul in $\frac{\partial}{\partial x} u(x, y)$

Hier kan gebruik gemaakt worden van de editor als bij de functie $u(x, y)$ de variabelen deze afhankelijk is. In de Editor is echter niet voorzien in de kromme



3) Vul in $\frac{\partial}{\partial x} u$

Diff(u,x)

Hier is geen Editor aangeboden, maar moet de student wel degelijk de syntax kennen. Er moet ingetikt worden: Diff(u,x) met een hoofdletter. Dat is de inerte vorm en dus symbolisch. Als er diff(u,x) ingevuld wordt, dan wordt dit in de Preview (vergrootglasje) wel goed weergegeven echter dan wordt er gerekend met 0. Immers de afgeleide van een constante is 0 en dan had er net zo goed Diff(y,x) ingevuld kunnen worden.

Figure 1.135: Differentiaal symbolisch met text invoer

Als de student tikt diff(y,x) dan komt er in de Preview hetzelfde te staan, echter de waarde ervan is dan gelijk aan 0; de response wordt dan gezien als zijnde 0. Daarmee wordt gerekend en daarmee wordt ook de grading gedaan. De student moet dus in dit geval wel degelijk goed geïnstrueerd worden wat betreft de syntax. Beter is het de student te instrueren dat hij Diff(y,x) invoert in het tekstveld zodat u de grading daarop aan kunt passen. Immers de student had ook diff(u,x) kunnen tikken met hetzelfde resultaat dat de response gelijk is aan 0 terwijl Diff(u,x) en Diff(y,x) door Maple als twee verschillende dingen wordt gezien.

TIP: Zie ook in paragraaf *Symbol Mode* (page 7) voor de bediening van de Editor waar studenten nog wel degelijk wat instructie voor moeten krijgen.

Hogere afgeleiden

4) Vul in $\Delta x^3 \frac{\partial^3}{\partial x^3} u$

Delta*x^3*Diff(u,x,x,x)

Hier is de instelling ook text entry only omdat het de beschikking over deze hoofdletter Delta maar ingevuld te worden: Delta*x^3*Diff(u,x,x,x)

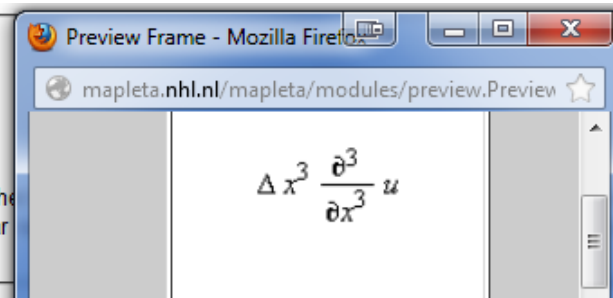


Figure 1.136: Hogere afgeleiden met het tekstveld

Voor de symbolische notatie is het dus beter om een tekstveld aan te bieden en de student goed te instrueren dat het hier gaat om de symbolische notatie. Immers de derde afgeleide van u (een constante) naar x is natuurlijk 0 zodat de grading niet echt uitsluitend zal geven als u niet de symbolische notatie hanteert met het inerte commando Diff.

Voor de functie u afhankelijk van x en y is de derde afgeleide naar x te noteren als het juiste antwoord geformuleerd is als:

```
$antw:=maple("diff(u(x,y),x,x,x)");
```

Deze formule kan wel met de *Editor* overhoord worden, maar natuurlijk ook met de instelling *text entry only* waarbij de student kan previewen. Ook hier is het handig als de student enigszins getraind is in het hanteren van de Editor of op de hoogte is van de syntax.

9) Vul in $\frac{\partial^3}{\partial x^3} u(x, y)$

Equation Editor

$$\frac{d}{dx} \left(\frac{d}{dx} \left(\frac{d}{dx} u(x, y) \right) \right)$$

Figure 1.137: Hogere afgeleiden met de Editor

TIP: Een alternatief is met subscript te werken voor de afgeleide. Dan kan wél weer van de *Editor* gebruik gemaakt worden: y_x .

Ook kan de student met underscore werken om subscript te bewerkstelligen dus bijvoorbeeld u_{xxx} . In de Editor komt dan vanzelf subscript te staan. Echter de formule van het antwoord in Maple-notatie zou dan in het *Algorithm* voorbereid moeten zijn als:

```
$antw6:=maple("(1-M^2)*Delta*x*u[xxx]-Delta*y^2*u[yyy]");
```

In Maple is subscript dus met vierkante haken.

6a) Vul in $(1 - M^2) \alpha x u_{xxx} - \frac{1}{12} \alpha y^2 u_{yyy}$

Equation Editor [Help](#)

$$(1 - M) \alpha x u_{xxx} - \frac{1}{12} \alpha y^2 u_{yyy}$$



In deze situatie is het handig om subscript te gebruiken en dan kan de Editor ook ingezet worden. Subscript kan verwezenlijkt worden door de paletten te gebruiken, maar de student kan met het toetsenbord ook u_{xxx} tikken om de subscript te bewerkstelligen. De Griekse letters kunnen uit de paletten gehaald worden.

Figure 1.138: Subscript in de Editor

1.4.19 Integralen

1.4.19.1 Onbepaalde integraal

Bij het herleiden van de onbepaalde integraal vraagt u in feite de primitieve van de integrand aan de student.

Question Name: 18c primitiveren 5

Herleid de volgende onbepaalde integraal

$$\int (\sin(3x) - 6 \cos(6x)) dx$$

This question accepts formulas in Maple syntax.
[Plot](#) | [Help](#) | [Preview](#)

Figure 1.139: Primitiveren, de onbepaalde integraal

De formule die door de student ingetikt dient te worden (in de tekst mode) kan gemakkelijk gematcht worden door middel van het differentiëren van het antwoord van de student met de *Grading Code*: `evalb(simplify(diff($antwoord,x)-(diff($RESPONSE,x)))=0) and evalb(StringTools[Search]("int", "$RESPONSE")=0);` Als u kiest voor de Editor, met *Symbol entry only*, is de laatste restrictie `and evalb(StringTools[Search]("int", "$RESPONSE")=0)` óók nodig want de student kan uit de paletten van de Editor de integraal nemen en dan wordt het antwoord ook goedgekeurd.

Omdat er bij onbepaald integreren altijd nog een willekeurige integratieconstante gegeven kan worden, matchen we hier juist de afgeleide van het correcte antwoord met de afgeleide van het antwoord van de student. De integratieconstante doet er dan niet toe. In het *Algorithm* hebt u bijvoorbeeld geprogrammeerd:

```
$a=range(1,9);
$b=range(2,10);
$c=range(2,10);
$vraag=maple("Int(sin($a*x)-$b*cos($c*x),x)");
$displayvraag=maple("printf(MathML[ExportPresentation]($vraag))");
$antwoord=maple("value($vraag)");
$displayantwoord=maple("printf(MathML[ExportPresentation]($antwoord))");
```

TIP: Let op dat u de vraag formuleert met het inerte commando met de hoofdletter (Int). Met het Maple-commando value, kan de waarde van deze onbepaalde integraal gegenereerd worden (de integratieconstante wordt bij Maple op 0 gesteld).

1.4.19.2 Bepaalde integraal

Als de student de uitkomst van de bepaalde integraal moet berekenen, kunt u gewoon het juiste antwoord met dat van de student matchen met de *Grading Code*: `evalb(simplify($antwoord-$RESPONSE)=0)` ;

Question Name: 18d integreren 4

Bereken de volgende bepaalde integraal met pen en papier.

$$\int_1^2 \left(8 a x^3 + \frac{1}{3 x^8} \right) dx$$

Je kunt het antwoord intikken zoals je dat ook op de rekenmachine doet, dus met sterretjes en dakjes en gedeeld door met een schuine streep. Dan heb je geen extra rekenmachine nodig voor deze opgave.

This question accepts numbers or formulas.

Plot | [Help](#) | [Change Math Entry Mode](#) | [Preview](#)

Figure 1.140: De bepaalde integraal berekenen

In het *Algorithm* programmeert u het volgende:

```
$a=range(1,9);
$b=range(1,5);
$integrand=maple("a*$a*x^($b)+1/($b*x^($a))");
$vraag=maple("Int($integrand,x=1..2)");
$displayvraag=maple("printf(MathML[ExportPresentation]($vraag))");
$antwoord=maple("simplify(value($vraag))");
```

TIP: Let hier eens op de instelling *Formula*. Dan is ook geen restrictie nodig in de *Grading Code* om te voorkomen dat de student gebruikmaakt van Maple-codes. Er wordt dus wel een exact antwoord gevraagd, dus een formule als antwoord. In feite is het beter om *Maple syntax* te hanteren maar dan moet u natuurlijk wel in de *Grading Code* programmeren dat "int" niet in het antwoord van de student voorkomt.

TIP: Bij dit soort vragen kan ook het numerieke vraagtype gehanteerd worden als het antwoord bestaat uit een getal zonder variabelen. Er kan in dat geval eventueel ook nog een tolerantie worden meegegeven als u afrondingen wilt accepteren.

1.4.19.3 Meervoudige integralen

In dit voorbeeld wordt dezelfde manier van doen als in het voorbeeld van de enkelvoudige bepaalde integraal gehanteerd. Komt er een enkel getal uit zonder variabelen, dan kunt u overgaan op het vraagtype numeric en opteren voor het exacte antwoord of zelfs afrondingen accepteren.

Question Name: 18e dubbelintegraal 3

Bereken de volgende dubbel integraal

$$\int_1^2 \int_y^3 (7 - x) dx dy$$

Je kunt het antwoord intikken zoals je dat ook op de rekenmachine doet, dus met sterretjes en dakjes en gedeeld door met een schuine streep. Dan heb je geen extra rekenmachine nodig voor deze som.

This question accepts numbers or formulas.

Plot | [Help](#) | [Change Math Entry Mode](#) | [Preview](#)

Figure 1.141: Dubbelintegraal berekenen

Hier is weer gekozen voor de instelling *Formula*, dan hoeft er in de *Grading Code* ook geen restrictie bij geprogrammeerd te worden dat het commando "int" niet mag voorkomen in het antwoord van de student. Let dan wel op dat de student het exacte antwoord dient te geven.

De vraag en het goede antwoord worden voorbereid in het *Algorithm*:

```
$a=range(1,9);
$vraag=maple("Int(Int(a*($a)*x,x=2*y..3),y=1..2)");
$displayvraag=maple("printf(MathML[ExportPresentation]($vraag))");
$antwoord=maple("value($vraag)");
$displayantwoord=maple("printf(MathML[ExportPresentation]($antwoord))");
```

1.4.19.4 De student voert de integraal in

Als u wilt dat de student de integraal opstelt en ook als integraal intikt, dan kunt u de *waarde* van de ingevoerde integraal matchen met die van de correcte integraal. Als de waarde van de correcte integraal in het *Algorithm* geprogrammeerd is als \$antw, kunt u gebruikmaken van de volgende *Grading Code*: `evalb(simplify($antw-($RESPONSE))=0)` and `evalb(StringTools[Search]("int", "$RESPONSE")>0)`;

Op deze manier kan de student ook een andere formulering van de integraal hanteren die dezelfde waarde heeft én u eist dat er een integraal ingevoerd is. Als u de instelling op *Symbol entry* hebt staan, dan wordt de ingevoerde integraal altijd vertaald naar het Maple-commando "int". In de string van de respons staat dan dus "int".

$\int_1^{10} x^2 dx$ (0%)

 "int(x^2,x = 1 .. 10)"

 "\$RESPONSE"

 Symbol entry only

Figure 1.142: De analyse van het antwoord van de student in de vorm van een integraal in Symbol mode

Echter de \$RESPONSE is de *waarde* van de integraal en daarmee wordt in de *Grading Code* gematcht.

Dat is hetzelfde als u niet de *Symbol mode* kiest voor de instellingen maar *Text entry only*. De student tikt de integraal in en dat is ook te controleren met behulp van StringTools en verder wordt de *waarde* van de integraal gebruikt om mee te matchen in de *Grading Code*.

Text entry only

```
int(x^2,x=1..10)
```

```
"int(x^2,x=1..10) "
```

"\$RESPONSE"

Text entry only

```
int(x^2,x=1..10)
```

```
333
```

\$RESPONSE

```
Int(x^2,x=1..10)
```

```
Int(x^2,x = 1 .. 10)
```

Figure 1.143: De analyse van het antwoord van de student in de vorm van een integraal in Text mode

In deze figuur zien we dat het invoeren van een integraal in het invulveld met *Text entry only* in de string van het antwoord letterlijk wordt vertaald naar wat de student invoert. Echter bij het tweede invulveld wordt de pure response (\$RESPONSE) direct vertaald naar de *waarde* van de integraal en bij het derde invulveld tikt de student "Int" met een hoofdletter. De response is in dat laatste geval niet meer de *waarde* van de integraal maar de integraal zelf. U kunt in dat geval ook afzonderlijk de integrand en apart de variabele met de grenzen overhoren. In feite wordt in de *Grading Code* altijd gematcht met de \$RESPONSE al zijn er genoeg mogelijkheden om in de *Grading Code* ook iets op te nemen wat terug te voeren is op de string van het antwoord.

TIP: In beide gevallen wordt in de *Preview* als de student "int" of "Int" intikt in het textveld, de integraal zelf weergegeven en niet de waarde ervan.

TIP: Als u van uw studenten eist dat ze "Int" hanteren in hun invoer in het textveld, kunt u twee dingen doen in de *Grading Code*:
`evalb(simplify($antw-value($RESPONSE))=0) and evalb(StringTools[Search]`
`("Int", "$RESPONSE")>0);`

Maar u kunt de integrand en variabele en grenzen apart beoordelen als de variabele \$antw in het *Algorithm* voorbereid is als de juiste integraal met de volgende *Grading Code*:

```
is((op(1,$antw))=(op(1,$RESPONSE)) )
and is( (op(2,$antw))=(op(2,$RESPONSE)) )
and evalb(StringTools[Search]("Int", "$RESPONSE")>0);
```

TIP: In het *Algorithm* kunt u het een en ander voorbereiden.

Hieronder ziet u wat Maple doet met de inerte integraal als daarvan de operanden opgevraagd worden

```
> op(1,Int(x^2,x=0..10));
```

$$x^2$$

(1.72)

```
> op(2,Int(x^2,x=0..10));
```

$$x=0..10$$

(1.73)

In het volgende voorbeeld wordt gevraagd de integraal op te stellen.

Question Name: 18f Volume Paraboloide

Geef de integraal om het volume te berekenen van een trechtervorm met een hoogte die evengroot is als de straal $R = 4$ van het grondvlak door de wortelfunctie (halve parabool) te wentelen om de verticale as.

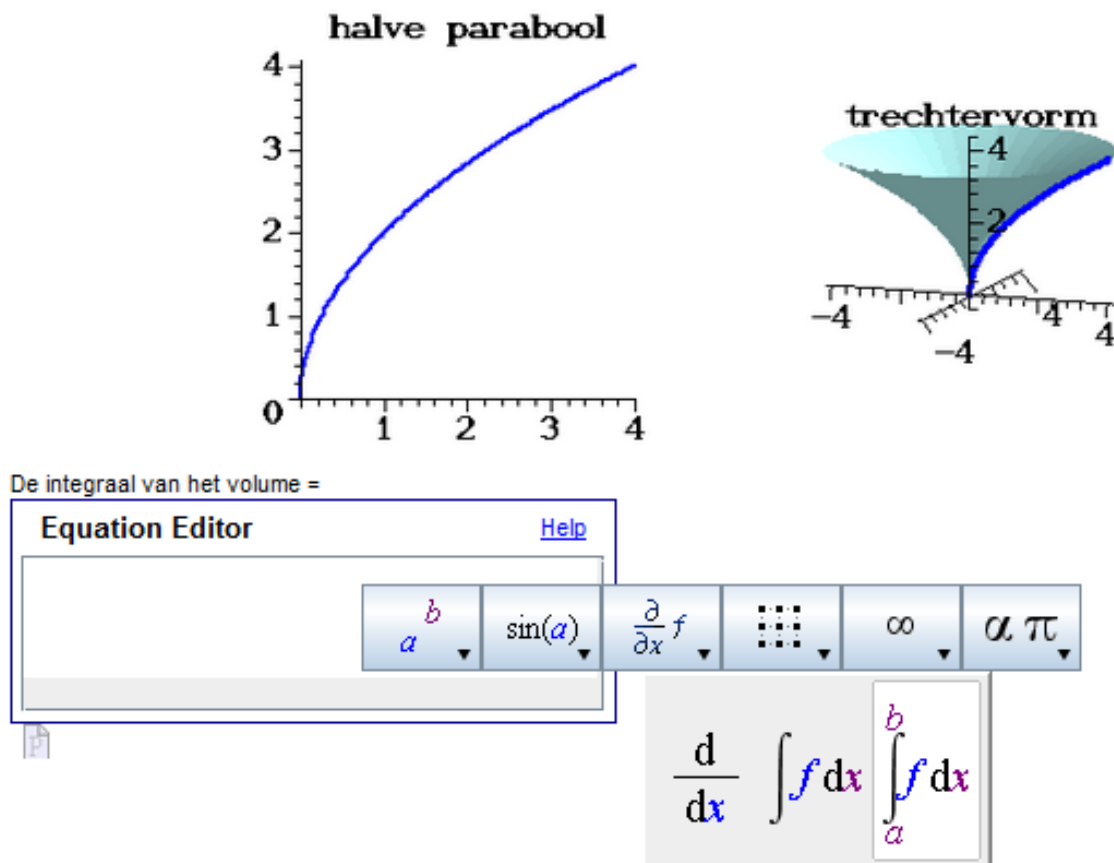


Figure 1.144: Invoeren van een integraal

TIP: Het mooiste is hier natuurlijk te opteren voor *Symbol Entry* en niet voor *Text entry*. Hoewel dat niet veel uitmaakt, maar de student moet in het geval van de *Text entry mode* wel de officiële *Maple syntax* kennen van het invoeren van de integraal met het commando `int` (of `Int`).

TIP: Let op dat de student dus wél een integraal invoert en niet alleen het antwoord. Dat is te checken bij de instelling *Symbol* met de extra restrictie in de *Grading Code*: `and evalb(StringTools[Search]("int", "$RESPONSE")>0);`

TIP: Echter als u de instelling *Text entry* hanteert, kan de student gebruikmaken van "int" maar ook van "Int".

In de *Grading Code* moet u dan nog wat zwaarder programmeren:

```
evalb(simplify($antw-value($RESPONSE))=0) and evalb(StringTools[Search]
("int", "$RESPONSE")>0) or evalb(StringTools[Search]("Int", "$RESPONSE")>0);
```

In het volgende voorbeeld moet de student ook een integraal invoeren. Echter deze integraal is symbolisch en in dat geval is het lastig om de waarde ervan te matchen met die van de student als de waarde van de integraal niet berekend kan worden.

7) Vul in $\int_0^1 (-w_x u - w u - w f) dx$

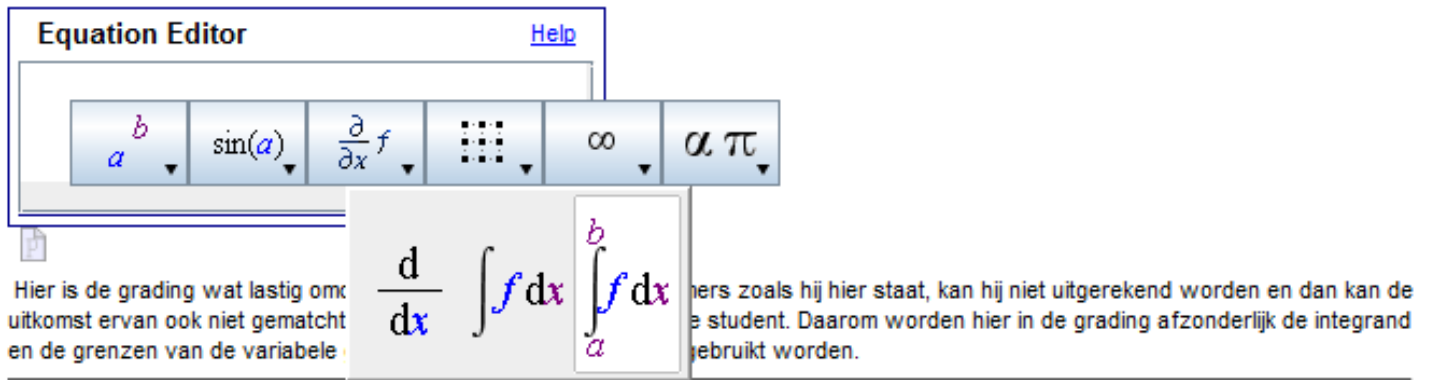


Figure 1.145: Invoeren van een integraal symbolisch

In het volgende kunt u met Maple uitproberen of deze symbolische integraal een waarde heeft.

U ziet een voorbeeld waarin het niet mogelijk is de waarde van de integraal te berekenen.

De x in de subscript zal roet in het eten gooien en de integraal wordt onverrichter zake weer teruggegeven.

```
> int(-w*u-w*u-w*f,x=0..1);
```

$$-fw - 2uw \quad (1.74)$$

```
> int(-w[x]*u-w*u-w*f,x=0..1);
```

$$\int_0^1 (-fw - uw - uw_x) dx \quad (1.75)$$

```
> op(Int(-w[x]*u-w*u-w*f,x=0..1));
```

$$-fw - uw - uw_x, x=0..1 \quad (1.76)$$

Bij het opvragen van de operanden van een integraal gebruikt u het inerte commando `Int`, zodat keurig de integrand en de variabele met de grenzen weergegeven worden. Echter hier had dat niet uitgemaakt als de waarde van de integraal toch niet berekend kan worden met `int` (immers de integraal wordt dan gewoon weer teruggegeven).

In het voorbeeld (Figure 1.145 (page 123)) is de grading wat lastig omdat de integraal symbolisch is. Immers zoals hij hier staat, kan hij niet uitgerekend worden en dan kan de waarde ervan ook niet gematcht worden met de waarde van het antwoord van de student. Daarom worden hier in de *Grading Code* afzonderlijk de integrand en de grenzen van de variabele gematcht.

Het juiste antwoord programmeert u in het *Algorithm* met `$antw7:=maple("Int(-w[x]*u-w*u-w*f,x=0..1)");`

U kunt hier prima de Editor aanbieden met de instelling *Symbol entry only*, maar ook als de student de Maple-syntax kent van integralen en subscript, kunt u *Text entry only* instellen met dezelfde *Grading Code*:

```
is( (op(1,$antw7))=(op(1,$RESPONSE)) )
and is( (op(2,$antw7))=(op(2,$RESPONSE)) )
and evalb(StringTools[Search]("int","$RESPONSE")>0);
```

Er moet in het laatste geval dan misschien nog wel gecontroleerd worden of "`Int`" of "`int`" voorkomt in het antwoord van de student. Het komt er dus op neer dat u de eerste operand (de integrand) van het antwoord van de student matcht met die van het correcte antwoord en evenzo de tweede operand: de integratievariabele met de juiste grenzen.

TIP: Deze manier van doen werkt dus alleen als de integraal niet berekend kán worden. Immers als u de instellingen op *Symbol entry* hebt gezet, wordt de integraal uit het palet vertaald naar het niet-inerte commando "int". Dus als de integraal wél te berekenen was, dan kan er niet gewerkt worden met de operanden van de integraal, zoals de integrand en de variabele met de grenzen.

TIP: Als u van uw studenten verlangt om expliciet de integraal in te voeren en daarbij wilt checken of zowel de integrand als ook de variabele met eventuele grenzen afzonderlijk gecontroleerd worden, zit er niets anders op dan dat u de instellingen zet op *Text entry only*. De student moet dan beslist het inerte commando "Int" hanteren om de integraal in te tikken. De *Preview* geeft dan geen verschil tussen "Int" en "int" maar de response van de student wordt in het geval van "int" vertaald naar de waarde van de integraal en in geval van "Int" vertaald naar de pure integraal.

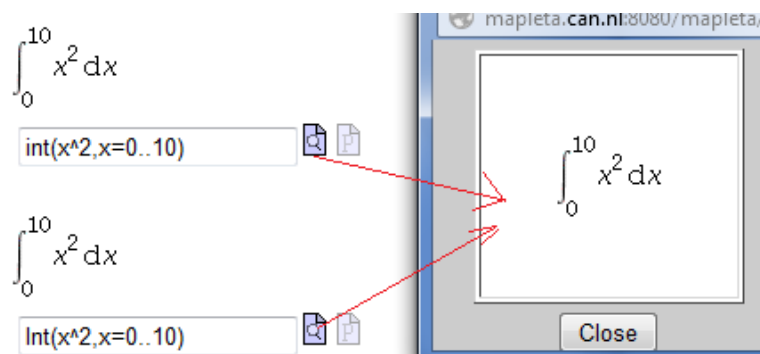


Figure 1.146: Verschil in invoer met Int en int en geen verschil in de Preview

1.4.20 Matrices en Vectoren

1.4.20.1 Inleiding

Bij Matrices en Vectoren is de *Grading Code* niet echt voor de hand liggend. Maar verder is het met de instellingen van *Maple syntax* mooi om te opteren voor de *Symbol Mode* om de studenten met behulp van een Editor de Matrix of Vector te laten invoeren.

Als twee matrices of twee vectoren gelijk zijn aan elkaar, dan zijn *alle* overeenkomstige elementen gelijk aan elkaar. Dit wordt gecheckt door Maple met de volgende *Grading Code*:

```
LinearAlgebra[Equal]($RESPONSE,$antw);
```

Hierin is \$RESPONSE steeds per definitie de respons van de student en \$antw is het in het *Algoritme* voorbereide antwoord. Deze *Grading Code* kunt u behalve voor Matrices ook voor Vectoren gebruiken.

TIP: U kunt ook eisen dat enkele elementen van de matrix of vector een bepaalde waarde hebben zoals met de *Grading Code*:

```
is($RESPONSE[2,1]=0) and is($RESPONSE[3,1]=0);
```

gecheckt wordt of het element van de tweede rij en de eerste kolom én het element van de derde rij en de eerste kolom van de respons gelijk zijn aan 0.

TIP: Als u decimale getallen hanteert (twee decimalen bijvoorbeeld) is wellicht de volgende *Grading Code* handig om te gebruiken.

```
LinearAlgebra[Equal](map(round,100*($RESPONSE)),map(round,100*($antw)));
```

Hiermee bereikt u dat eerst alle elementen van de matrix of vector met 100 vermenigvuldigd worden en vervolgens afgerond. Het steekt namelijk nogal nauw dat met Equal uit het LinearAlgebra pakket de Matrices en Vectoren ook exact overeenkomen wat betreft de elementen.

TIP: Bij het overhoren van Matrices en Vectoren moeten studenten goed opgevoed worden met name bij het kiezen van de juiste dimensies van de Matrix. In het veld van de Equation Editor met de rechter muisknop klikken en dan komt er een menu tevoorschijn. (MapleSoft heeft in de volgende versie de knoppen er op voorhand al ingezet.) Zie ook onderstaande figuur.

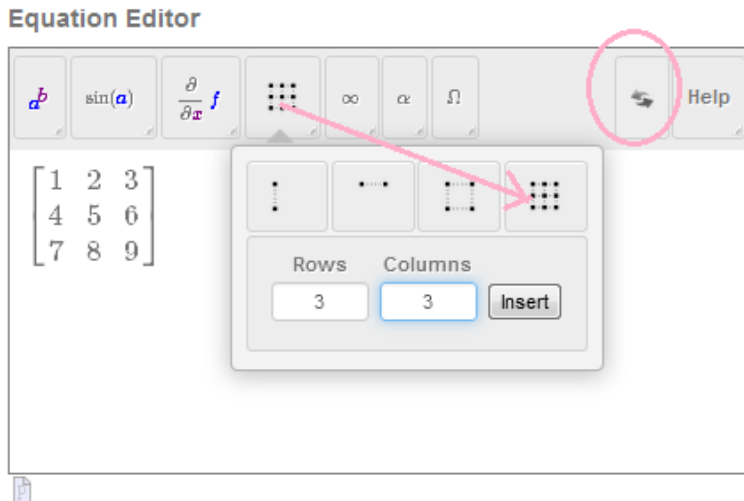


Figure 1.147: Het invoeren van een Matrix door de student in de Editor

TIP: Willen studenten iets verbeteren, laat ze dan alles selecteren en dit met de dubbele pijl, die hier in de figuur omcirkeld is, weghalen en niet met de *Backspace*-knop, want dan kan het zijn dat er onbedoelde code in de Editor blijft staan en dan zal het systeem moeite hebben met het matchen. De student kan daarmee het hele veld schoonvegen.

Bij het matchen van Matrices en Vectoren is het lastig om toleranties te hanteren. Laat de student dan ook zo mogelijk het exacte antwoord intikken of geef goed aan hoe de afronding moet zijn en bereid het correcte antwoord op die manier voor in het *Algorithm*.

TIP: Let op dat een kolomvector door het systeem niet als een Matrix wordt gezien. Het goede antwoord moet dan ook als kolomVector voorbereid zijn (in het *Algorithm*) om mee te matchen. Let dus op dat bij gebruik van de Editor die u aanbiedt bij de instellingen *Maple syntax* met *Symbol entry only* dat als de student opteert voor een matrix van 1 kolom, dat het systeem deze dan sowieso altijd ziet als Vector! bij het matchen moet dus het voorbereide antwoord ook als Vector gedefinieerd zijn!

TIP: Omdat u bij dit soort vragen in feite altijd de Editor zult gebruiken, is het handig om in het veld van het correcte antwoord te programmeren, zodat de student in de feedback bij het correcte antwoord ook altijd de tweedimensionale presentatie ziet:

```
printf(MathML[ExportPresentation]($antw));
```

1.4.20.2 Analyse van de invoer

In de *Grading Code* zult u enerzijds gebruikmaken van de \$RESPONSE van de student om te matchen met het correcte antwoord, anderzijds is het van belang nog extra gebruik te kunnen maken van de string van het antwoord van de student om te kijken wat de student precies heeft ingetikt. Met StringTools kunt u dan verder programmeren hoe u de invoer van de student wilt hebben door te kijken naar de string van het antwoord van de student: "\$RESPONSE".

In onderstaande figuur ziet u in enkele voorbeelden van invoer in de Editor hoe deze vertaald wordt naar een string.


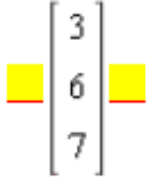
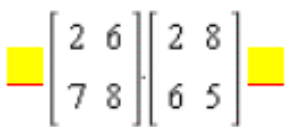
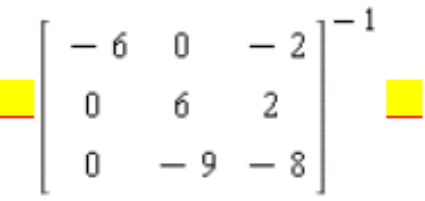
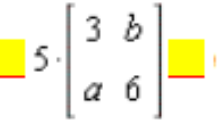
| Your response | String response |
|---|--|
|  | <code>"Matrix(2,2,{(1, 1) = 2, (1, 2) = 5, (2, 1) = u, (2, 2) = m},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])"</code> |
|  | <code>"Vector[column] (3,{1 = 3, 2 = 6, 3 = 7},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])"</code> |
|  | <code>"Typesetting:-delayDotProduct(Matrix(2,2,{(1, 1) = 2, (1, 2) = 6, (2, 1) = 7, (2, 2) = 8},datatype = anything,storage = rectangular,order = Fortran_order,shape = []),Matrix(2,2,{(1, 1) = 2, (1, 2) = 8, (2, 1) = 6, (2, 2) = 5},datatype = anything,storage = rectangular,order = Fortran_order,shape = []))"</code> |
|  | <code>"1/Matrix(3,3,{(1, 1) = -6, (1, 3) = -2, (2, 2) = 6, (2, 3) = 2, (3, 2) = -9, (3, 3) = -8},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])"</code> |
|  | <code>"5*Matrix(2,2,{(1, 1) = 3, (1, 2) = b, (2, 1) = a, (2, 2) = 6},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])"</code> |

Figure 1.148: Invoer van matrices vertaald naar de string

In de figuur hierboven is te zien dat een matrix met één kolom ingevoerd, vertaald wordt naar een Vector[column].

Als de student twee matrices invoert met een (gewone) punt ertussen, dan worden deze twee matrices opgevat als het inwendig product (DotProduct) van de twee matrices. U kunt bijvoorbeeld met StringTools controleren hoeveel matrices de student heeft ingevoerd. Als u bijvoorbeeld per se wilt dat de student de twee matrices zelf met elkaar vermenigvuldigt, mag er natuurlijk maar één matrix ingevoerd worden. U controleert dan bijvoorbeeld dat er slechts één recht haakje "[" in het stringantwoord voorkomt met

```
LinearAlgebra[Equal]($RESPONSE,($\ntw)) and evalb(StringTools
```

```
[CountCharacterOccurrences]("$RESPONSE","[")=1);
```

In de figuur hieronder is te zien dat de \$RESPONSE (dus niet de string, maar het antwoord van de student) soms door het systeem al uitgerekend is. In de Editor kan de student geen commando's tikken echter de student kan wel gebruikmaken van (matrix)vermenigvuldigingen, inversen, optellen en aftrekken en dergelijke en dat is soms niet uw bedoeling.

| Your response | response |
|---|--|
| $\begin{bmatrix} 2 & 6 \\ 5 & 4 \end{bmatrix} \begin{bmatrix} 2 & 7 \\ 4 & 3 \end{bmatrix}$ | <pre>Matrix(2,2,{(1, 1) = 28, (1, 2) = 32, (2, 1) = 26, (2, 2) = 47},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])</pre> |
| $\begin{bmatrix} 3 & 8 \\ 7 & 9 \end{bmatrix}^{-1}$ | <pre>Matrix(2,2,{(1, 1) = -9/29, (1, 2) = 8/29, (2, 1) = 7/29, (2, 2) = -3/29},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])</pre> |
| $5 \cdot \begin{bmatrix} a & k \\ 7 & 8 \end{bmatrix}$ | <pre>Matrix(2,2,{(1, 1) = 5*a, (1, 2) = 5*k, (2, 1) = 35, (2, 2) = 40},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])</pre> |

Figure 1.149: Invoer van matrices

In bovenstaande figuur is te zien dat de student een matrixvermenigvuldiging invoert met behulp van de gewone punt tussen de twee matrices. De response (\$RESPONSE) wordt dan vertaald naar de uiteindelijke matrix die deze vermenigvuldiging oplevert. Hiermee wordt dan het juiste antwoord vergeleken met bijvoorbeeld de *Grading Code*: `LinearAlgebra[Equal]($RESPONSE, ($antw))`;

Ook is te zien dat de student de matrix tot de macht -1 kan invoeren om een juiste beoordeling te krijgen als naar de inverse wordt gevraagd. Dit gebeurt ook als de student $\frac{1}{\begin{bmatrix} 3 & 8 \\ 7 & 9 \end{bmatrix}}$ had ingevoerd.

De response is dan al door het systeem uitgerekend. U kunt natuurlijk hier weer extra programmeren met `StringTools` of de student deze truc ook toepast en eisen dat in de string bijvoorbeeld "1/Matrix....." niet voorkomt met de *Grading Code*:

```
LinearAlgebra[Equal](simplify($RESPONSE),$ans) and evalb(StringTools[Search]
```

```
("1/M","$RESPONSE")=0);
```

De scalaire vermenigvuldiging die de student invoert met behulp van een sterk (de student ziet een middot in de Editor) wordt door het systeem vertaald naar de resulterende matrix waar alle elementen met deze scalar vermenigvuldigd zijn. Hetzelfde geldt voor optelling, aftrekking en vermenigvuldiging van matrices.

Als u van dit soort dingen op de hoogte bent, kunt u er in de *Grading Code* mee rekening houden.

1.4.20.3 Voorbeeld

In het volgende voorbeeld moet er een 3×4 -matrix ingevoerd worden door de student.

06 Lineair systeem matrix

Gegeven zijn de volgende drie vergelijkingen met de onbekenden x , y en z .

| |
|-------------------------|
| $-13x + 4y - 10z = -11$ |
| $15x - 4z - 4y = 5$ |
| $-13x + 10y - 7z = -10$ |

Geef de aangevulde 3 bij 4 matrix van het lineaire systeem ten opzichte van de onbekenden $[x,y,z]$.

In dit voorbeeld maakt het niet uit of de volgorde van de vergelijkingen verwisseld wordt of dat de vergelijkingen vereenvoudigd zijn of niet. De coëfficiëntenmatrix aangevuld met de bijbehorende rechterlid-kolom geeft een eenduidige oplossing van het systeem.

Equation Editor

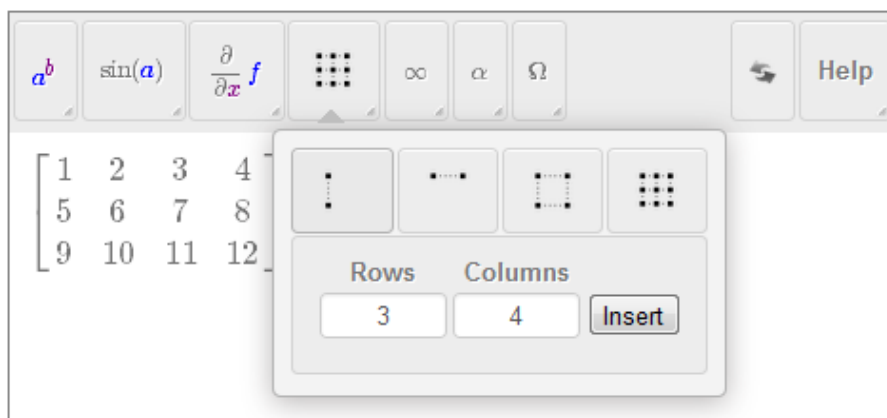


Figure 1.150: Matrix invoeren

Het bijbehorende *Algorithm* hiervan is:

```
$a=range(-15,15);
$b=range(-15,15);
$c=switch(rint(2),range(-15,-1),range(1,15));
$d=range(-15,15);
$e=switch(rint(2),range(-15,-1),range(1,15));
$f=range(-15,15);
$g=switch(rint(2),range(-15,-1),range(1,15));
$h=range(-15,15);
$i=range(-15,15);
$j=range(-15,15);
$k=range(-15,15);
$l=range(-15,15);
$index=rint(3);
$lijstA=maple("[1,2]");
```



```

$A=maple("StringTools[Randomize]():combinat[randperm]($lijstA)");
$ad1=maple("$A[1]");
$ad2=maple("$A[2]");
$x=switch($index,"x","y","z");
$y=switch($index+$ad1,"x","y","z","x","y","z","x");
$z=switch($index+$ad2,"x","y","z","x","y","z","x");
$e1=maple("($a)*$x+($b)*$y+($c)*$z=$j");
$E1=maple("printf(MathML[ExportPresentation]($e1))");
$e2=maple("($d)*$x+($e)*$z+($f)*$y=$k");
$E2=maple("printf(MathML[ExportPresentation]($e2))");
$e3=maple("($g)*$x+($h)*$y+($i)*$z=$l");
$E3=maple("printf(MathML[ExportPresentation]($e3))");
$M=maple("LinearAlgebra[GenerateMatrix]([$e1,$e2,$e3],[x,y,z],augmented=true)");
$displayantw=maple("printf(MathML[ExportPresentation]($M))");
$test=maple("LinearAlgebra[ReducedRowEchelonForm]($M)");
$displaytest=maple("printf(MathML[ExportPresentation]($test))");

```

In bovenstaande wordt eerst een aantal coëfficiënten voorbereid ten behoeve van de vergelijkingen.

Verder zijn de variabelen van de vergelijkingen flink door elkaar gegooid met behulp van \$index en \$ad1 en \$ad2.

Zie in de *Handleiding Randomiseren* over de functie switch.

De vergelijkingen worden gecodeerd met MathML ten behoeve van de presentatie in de vraag.

De matrix \$M wordt gegenereerd door Maple. De variabelen \$displayantw en \$displaytest zijn ter controle waarbij de variabele \$test gebruikt wordt in de *Grading Code*.

In de rubriek van het correcte antwoord communiceert u `printf(MathML[ExportPresentation]($M))`; ten behoeve van een mooie feedback.

In de *Grading Code* programmeert u als volgt:

```
LinearAlgebra[Equal]($test,LinearAlgebra[ReducedRowEchelonForm]($RESPONSE));
```

Deze gereduceerde vorm van de aangevulde matrix is onafhankelijk van de rekenwijze en de volgorde van de vergelijkingen die de student hanteert.

In het *Algorithm* is de variabele \$test, de gereduceerde matrix, in feite al voorbereid.

```
$test=maple("LinearAlgebra[ReducedRowEchelonForm]($M)");
```

TIP: Wilt u toch afrondingen toelaten bij het matchen van de Matrices en Vectoren, probeer dan in de *Grading Code* iets dergelijks als:

```
LinearAlgebra[Equal](LinearAlgebra[ReducedRowEchelonForm]
(map(round,100*($RESPONSE)),LinearAlgebra[ReducedRowEchelonForm](map(round,100*$antw)));
```

Hiermee wordt het afgeronde antwoord van de student gematcht met het afgeronde antwoord van het correcte antwoordmodel.

Er wordt dus gekeken of de aangevulde coëfficiëntenmatrix het systeem van de gegeven vergelijkingen dekt, waar \$M het juiste antwoord is. Het maakt dan niet uit in welke volgorde de vergelijkingen staan.

1.4.20.4 Random matrix

Om te beginnen programmeren we een random Matrix. Dat kan ook door alle coëfficiënten stuk voor stuk als range te definiëren, maar deze gaat wel zo gemakkelijk.

```

$n=3;
$A=maple("randomize():LinearAlgebra[RandomMatrix]($n,$n,density=0.75,generator=rand(-9..10))");
$displayA=maple("printf(MathML[ExportPresentation]($A))");

$B=maple("randomize():LinearAlgebra[RandomVector](3,generator=rand(-9..9))");

```

1.5 Vraagtype Formula

Het zelfstandige vraagtype *Mathematical Formula* biedt een aantal mogelijkheden om extra dingen met formules te doen hoewel *Maple-graded* vragen erg veel aankunnen. Dit vraagtype is ook ondergebracht binnen de *Question Designer* en heet dan *Formula*, maar biedt voornamelijk dezelfde mogelijkheden als het zelfstandige vraagtype *Mathematical Formula*.

TIP: Het is met dit vraagtype ook mogelijk om in verschillende situaties getallen en formules te overhoren met eventueel een tolerantie daarbij. Tik dan bij het goede antwoord een vraagteken erbij met de tolerantie. In deze paragraaf zijn wel voorbeelden daarvan, zoals in *Figure 1.159 (page 136)* wordt getoond.

Een mogelijkheid voor een formule is $(-17.78 \pm 0.01) + (0.56 \pm 0.01) * F$.

U hebt dan de mogelijkheid om in elk van de numerieke delen van de formule een tolerantie aan te brengen. (Werk wel met haakjes voor elk deel. Echter u kunt dan geen formules met lettercombinaties gebruiken en studenten mogen de sterren voor vermenigvuldigen ook weglaten.

Eventueel kunt u ook een algoritmische variabele gebruiken voor de tolerantie.

TIP: Let op dat duizendtalseparatoren beslist niet geaccepteerd worden in dit vraagtype.

Hieronder is te zien welke subtypen er mogelijk zijn voor de vraagstelling binnen dit vraagtype.

Bij het zelfstandige vraagtype Mathematical Formula lijken er meer mogelijkheden te zijn dan bij het vraagtype Formula binnen de Question Designer maar dat is maar schijn.

| Mathematical Formula | Question Designer Formula |
|---|---|
| Formula - e.g. $x^2 \sin(x^2)$ | Formula - e.g. $e^x \sin(x^2)$ |
| Formula without logs and trig - e.g. $x^2/(x^2+1)$ | Formula that matches responses to within +C |
| Formula with physical units - e.g. $-9.8 (1/2)t^2 \text{ m/s}^2$ | Vector of formulas - e.g. $(2t, t+1, 3t^2)$ |
| Formula that matches responses to within +C | Equation - e.g. $x+3y=1$ |
| Equation - e.g. $y=x^2-1$ | Formula without simplification |
| Unordered list of formulas - e.g. $1 + \sqrt{5}$; $1 - \sqrt{5}$ | Chemical Equation - e.g. $2H_2 + O_2 \rightarrow 2H_2O$ |
| Ordered lists of formulas - e.g. $2x^3, 4x$ | |
| Vectors of formulas - e.g. $(2t, 3t+1, 4)$ | |
| Chemical equation - e.g. $2H_2 + O_2 \rightarrow 2H_2O$ | |

Figure 1.151: Subtypes van het vraagtype Formula

In de volgende paragrafen worden de mogelijkheden besproken.

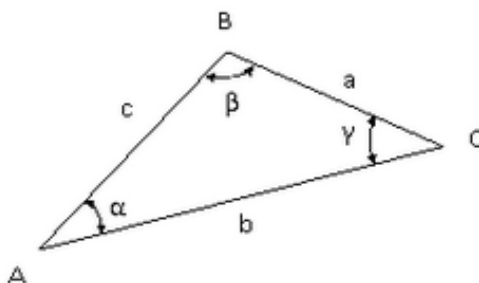
1.5.1 Formula

De instelling met *Formula* bij het vraagtype *Mathematical Formula* is zogenaamd "gebruiksvriendelijk" maar LET OP: lettercombinaties worden steeds als vermenigvuldigingen gezien. De lettercombinatie ab wordt bijvoorbeeld altijd als $a*b$ vertaald en p^2 wordt als $p*2$ opgevat.

Bij het getal π maakt het niet uit of Pi of pi wordt ingetikt. Beide keren wordt het als de bedoelde letter opgevat en dus niet als $p*i$ gezien. Berekeningen $\sin(\pi/2)$ of $\sin(Pi/2)$ of direct het antwoord 1 wordt allemaal goedgerekend.

De student kan bij sommige instellingen (hier bij *Formula*) kiezen voor de stijl waarin het antwoord ingevuld moet worden met de link *Change Entry Style*. Dat wil zeggen kiezen tussen tekst-invoer of met behulp van een Editor.

12 drie zijden



Gegeven de driehoek met de zijden

zijde $a = 13$

zijde $b = 14$

zijde $c = 3$

Bereken hoek α in graden en rond af op 1 decimaal.

This question accepts numbers or formulas.

Help | [Switch to Equation Editor](#) | [Preview](#)

Equation Editor

 a^b
 $\frac{a}{b}$
 \sqrt{a}
 $|a|$
 π
 \sin

This question accepts numbers or formulas.

Help | [Switch to Text Entry](#)

Figure 1.152: Change Entry Style

Bij dit vraagtype kan de student dus steeds zelf kiezen voor het invoeren van getal of formule in het antwoordveld als tekst (dus als ééndimensionale formule met een *Preview*-functie), of met behulp van een Editor. Het omwisselen van *Text Mode* en *Symbol Mode* gaat met betreffende link. De *Preview*-functie als de student tekst invoert, is echter zeer beperkt en voldoet niet in alle opzichten (voor exponentiële functies is deze ronduit slecht). Echter er vindt géén automatische vereenvoudiging plaats zoals bij de *Preview* van de *Maple-graded* vraag met *Maple syntax Text entry only* instelling en dat kan soms weer een voordeel zijn.

Als u voor dit vraagtype kiest in de *Question Designer*, dan kan de student middels het *Sigma*-knopje onder het invulveld wisselen *Symbol Mode* en *Text Mode*. De Editor geeft niet veel mogelijkheden. In de text-mode kan op *Preview* geklikt worden maar deze *Preview* is niet erg goed, maar geeft in ieder geval geen automatische vereenvoudiging en dat is soms een voordeel.

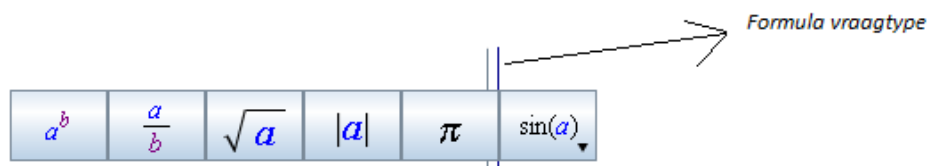
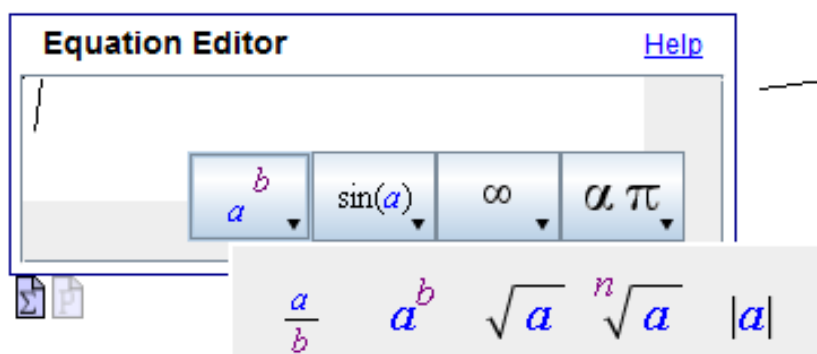


Figure 1.153: Vraagtype Formula in de Question Designer

Vergelijk ook eens de Editor van het *Maple-graded* vraagtype met de instelling *Formula*. Deze Editor is iets uitgebreider maar er is nog steeds geen mogelijkheid om lettercombinaties en andere geavanceerde formules in te vullen. Hier ook weer de mogelijkheid om met het *Sigma*-knopje onder het invulveld te wisselen tussen *Symbol Mode* en *Text Mode*.

$$3pq - 14x + 5x - 7pq + 3$$



*Maple-graded vraagtype
met Formula instelling*

Figure 1.154: Vraagtype Maple-graded met instelling Formula

TIP: Het vraagtype *Formula* kan ook gebruikt worden om getallen te overhoren met tolerantie zoals de vraag van Figure 1.152 (page 131). Hier wordt het goede antwoord \$antw in het *Algorithm* voorbereid en bij de instellingen voor het correcte antwoord van het antwoordveld: \$antw ? 0.1 . Het betekent dus een tolerantie van 0.1 in het antwoord.

Het aardige is dat voor de tolerantie ook een algoritmische variabele gebruikt kan worden!

Maar let op dat duizendtal separators niet geaccepteerd worden bij dit vraagtype. Zorg er dan voor dat in de feed back en ook in de vraag zelf geen duizendtal separators worden gehanteerd. Wilt u deze dan toch gebruiken, neem dan uw toevlucht tot het vraagtype *Numeric*.

1.5.2 Formula without Simplification

Bij dit vraagtype *Formula* binnen de *Question Designer* is nog een extra mogelijkheid om te kiezen voor sub-type *Formula without simplification*. Dit houdt in dat het antwoord niet als berekening, maar echt in de vorm gegeven moet worden waarin ook het correcte antwoord gegeven is. Bij deze instelling is bijvoorbeeld de berekening niet goed maar een breuk met dezelfde waarde wel. Is bijvoorbeeld het antwoord $\pi/2$, dan wordt bij de instelling *Formula without simplification* $5*\pi/10$ niet goed gerekend, maar bij de instelling met *Formula* zou dit wel goedgerekend worden. Het is raadzaam dit soort dingen even uit te proberen. Bij de instelling met *Formula* kan bijvoorbeeld goedgerekend worden: $\pi/2 = \pi/2 = 5*\pi/10 = 5\pi/10$ enz.

TIP: Bij de instelling met *Formula* kan steeds een tolerantie opgegeven worden met behulp van het vraagteken, maar bij de instelling *Formula without simplification* natuurlijk niet!

Liever gebruiken we niet dit vraagtype om formules te overhoren, maar in bijzondere gevallen kan het wel van pas komen.

In het volgende voorbeeld wordt de limiet overhoord met wortels. Het is de bedoeling dat de eventuele wortelvorm die als antwoord gegeven moet worden zo goed mogelijk vereenvoudigd wordt en daarom gebruiken we speciaal dit vraagtype *Formula without simplification*.

Description: 11b limits problem 7a1

Jump To: [Question](#) | [Information Fields](#)

Question:

Evaluate:

$$\lim_{x \rightarrow \infty} \left(\frac{7x+6}{\sqrt{8x^2+4x+7}} \right) = \text{7/4*sqrt(2)}$$

Enter the answer in the simplified form and no decimal numbers.

Close

$$\frac{7}{4} \sqrt{2}$$

Close

Figure 1.155: Limieten met wortelvorm

In de figuur hierboven ziet u waarom dit vraagtype voor formules in feite zeer ongeschikt is. De student kan het zelfs presteren om in te voeren: 7/4sqrt2!!! Maar dit wordt toch naar de juiste formule vertaald volgens de *Preview*. En in beide situaties wordt het als correct beoordeeld. Echter dit soort dingen zijn niet aanbevelenswaardig.

Afgezien daarvan moet u bij de vraagstelling dus wel expliciet communiceren dat het antwoord in **vereenvoudigde vorm** gegeven moet worden. Deze vereenvoudigde vorm bereid u voor in het *Algorithm*. Bij wortels is dat wat lastig, maar u moet er voor zorgen dat het antwoord in de **vereenvoudigde vorm met sqrt** voorbereid wordt (en niet met $^{(1/2)}$).

Bijvoorbeeld op de volgende manier:

```
$antwoord=maple("$d/$a1*sqrt($a1)");
```

Let hierbij op de quotes om te voorkomen dat Maple er iets met $^{(1/2)}$ van maakt.

Dan maakt u in de *Question Designer* het antwoordveld aan met vraagtype *Formula* en subtype *Formula without simplification* als volgt.

Choose Question Type

- Formula**
- Maple
- Multiple Choice
- Numeric
- List
- Essay

Formula:

Weighting: 1

Sub-type: Formula without simplification

Answer: \$antwoord

Figure 1.156: Vraagtype Formula met wortels

Dit vraagtype laat toe dat studenten zelf beslissen of ze met de Editor werken of niet. De student moet de wortelvorm met behulp van sqrt invoeren en krijgt in de *Preview* dan een wortelteken te zien zoals *Figure 1.155* (page 133) laat zien. Echter de invoer 7/sqrt(8)

wordt hier als fout beoordeeld, omdat de instellingen op *Formula without simplification* staan en het antwoord als vereenvoudigde vorm is voorbereid.

In de Editor kan de student invoeren: $\frac{7}{4}\sqrt{2}$ of $\frac{7}{4}\cdot 2^{\frac{1}{2}}$, maar bijvoorbeeld niet $\frac{7}{\sqrt{8}}$ of $\frac{7}{8^{\frac{1}{2}}}$ of een andere niet vereenvoudigde

vorm. Dat alles alléén op voorwaarde dat het juiste antwoord in vereenvoudigde vorm is voorbereid met sqrt (dus met behulp van quotes). Als u dat niet doet, moet de student beslist het antwoord met de vorm $^{(1/2)}$ invoeren en dan zou de student in de *Preview* iets kunnen krijgen wat hem helemaal op het verkeerde been zet.

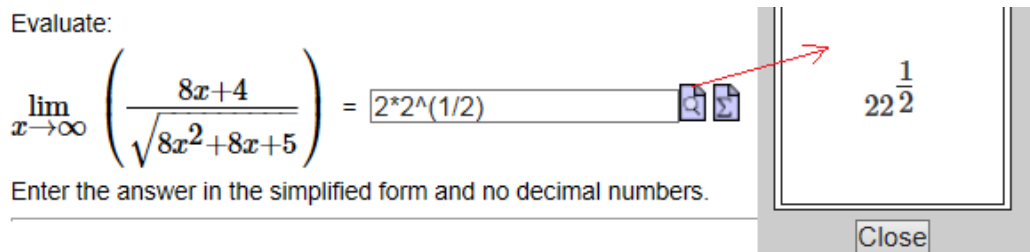


Figure 1.157: Preview met wortels

TIP: Deze instelling met sub-type *Formula without simplification* kan ook met het zelfstandige vraagtype *Mathematical Formula* gemaakt worden, maar dan kunnen de instellingen met *Formula without simplification* alleen bewerkstelligd worden als er in de broncode nog de volgende regel wordt meegegeven.
grading=form@

TIP: Eventueel overheert u het antwoord in bijvoorbeeld twee delen met een apart antwoordveld voor het getal en nog een apart antwoordveld voor de wortel.

1.5.3 Formula that matches responses to within +C

Het sub-type *Formula that matches responses to within +C* wil zeggen dat op een constante na de formule als goed beoordeeld wordt. Dus $x+3$ en $x+6$ zouden beide goedgekeurd kunnen worden met deze instelling.

1.5.4 Lijsten en Vectoren met Formula

Met het zelfstandige vraagtype *Mathematical Formula* is het mogelijk om geordende lijsten van getallen of formules te overhoren met de instelling *Ordered list of formulas*. U geeft dan gewoon het rijtje getallen of formules op als juiste antwoord, gescheiden door komma's.

Ook een ONgeordende lijst kan overhoord worden met *Unordered list of formulas*. Echter dan zijn de verschillende getallen of formules gescheiden door puntkomma's.

Dan is er ook nog de mogelijkheid om vectoren (een rijtje getallen of formules met haakjes eromheen en gescheiden door komma's) te overhoren.

Bij ditzelfde vraagtype binnen de *Question Designer* zitten deze drie soorten subtypen opgesloten in één soort namelijk het subtype *Vector of formulas*. Zie Figure 1.158 (page 135).

TIP: Bij het zelfstandige vraagtype *Mathematical Formula* staat er in de vraag voor de student wel bij hoe het antwoord ingetikt dient te worden, maar bij dit vraagtype binnen de *Question Designer* wordt dat niet vermeld. Het is raadzaam om dit dan in de tekst van de vraag op te nemen, zodat de student weet hoe hij het antwoord moet intikken.

17 Lijsten/Vectoren

Source

x_2 x^2

Font Size

Σ

Vul in de vector (3,4) (mag met en zonder haakjes).

Mathematical formula

Geef de kentallen van de vector in willekeurige volgorde gescheiden door puntkomma.

Mathematical formula

Geef in willekeurige volgorde de lijst met vectoren (3,4) en (5,7) en scheid deze vectoren door middel van een puntkomma.

Mathematical formula

Vul de vector in (3.1,5.7) met een

Mathematical formula

Cancel

Edit Response Area

Choose Question Type

- Essay
- Free Body Diagram
- List
- Maple-graded
- Math App
- **Mathematical formula**
- Multiple Choice
- Numeric

Weighting 1

Sub-type: Vector of formulas - e.g. (2t, t+1, 3t^2)

Answer (3,4) ; (5,7)

Figure 1.158: Question Designer Lijsten en Vectoren

Als dit vraagtype binnen de *Question Designer* wordt gehanteerd, dan is de enige mogelijkheid te kiezen voor het sub-type *Vector of formulas*. In deze instelling zit zowel de ongeordende lijst als ook de geordende lijst opgesloten. Een ongeordende lijst kan dus ook overhoord worden in dit vraagtype binnen de *Question Designer*, maar dan moet de puntkomma als separator getikt worden. Er kan zelfs een ongeordende lijst met vectoren overhoord worden! Of een geordende lijst met ongeordende lijstjes (vectoren) zoals in *Figure 1.158* (page 135) te zien is.

TIP: De student hoeft de haakjes om de vectoren niet beslist in te tikken, maar wel de komma.

TIP: Geef aan in de vraag hoe het antwoord ingetikt dient te worden.

TIP: Als onderdelen van de geordende of ongeordende lijst niet goed zijn, is het hele antwoord fout. Er is in dit vraagtype geen mogelijkheid tot gedeeltelijk goedrekenen.

TIP: Ook geordende lijsten of vectoren met formules kunnen ingevoerd worden, maar let dan wel op dat de formules niet al te geavanceerd moeten zijn want lettercombinaties worden als vermenigvuldigingen gezien en dergelijke. Bovendien kan de student bij deze instellingen NIET kiezen voor de Editor.

TIP: U kunt ook nog tolerantie aanbrengen in het antwoord met het vraagteken.
(3.1?0.1,5.7?0.1)

Heel belangrijk bij dit vraagtype is de instelling voor het soort formules dat overhoord gaat worden: Het is zelfs mogelijk om chemische formules te overhooren, geordende of ongeordende lijsten met formules of getallen en zelfs vergelijkingen in een willekeurige vorm.

Zie ook paragraaf *Vergelijkingen met Formula* (page 136).

Edit Response Area

Choose Question Type

- Essay
- Free Body Diagram
- List
- Maple-graded
- Math App
- **Mathematical formula**
- Multiple Choice
- Numeric
- Sketch

Mathematical formula:

Weighting: 1

Sub-type: Vector of formulas - e.g. (2t, t+1, 3t²)

Answer: (3.1?0.1,5.7?0.1)

Formula - e.g. $e^x \sin(x^2)$
 Formula that matches responses to within +C
Vector of formulas - e.g. (2t, t+1, 3t²)
 Equation - e.g. $x+3y=1$
 Formula without simplification
 Chemical Equation - e.g. $2H_2 + O_2 \rightarrow 2H_2O$

Figure 1.159: De subtypen van Formula in de Question Designer

TIP: Helaas is bij de beantwoording van deze vraag de waardering 0 of 100%. De student kan geen deelpunten krijgen als het voor een deel goed beantwoord is.

1.5.5 Vergelijkingen met Formula

Voor het gebruik van het sub-type *Equation* van het vraagtype *Formula* binnen de *Question Designer* is even wat extra informatie nodig.

Als u het goede antwoord invult, moet dat beslist in expliciete vorm staan! Het is wat misleidend als er staat *Equation* e.g. $x+3y=1$. Op deze manier het correcte antwoord formuleren, zal niet tot de gewenste resultaten leiden. Maar als u het correcte antwoord in expliciete vorm geeft, dan kan de student daar van alles van maken als het een gelijkwaardige vergelijking is. Dus vul voor het correcte antwoord bijvoorbeeld in: $x=1-3y$ of $y=(1-x)/3$, dan kan de student elke vergelijking intikken die hieraan gelijkwaardig is en dat wordt dan als correct beoordeeld. Dit lukt alleen als een van de variabelen wel degelijk in expliciete vorm geschreven kan worden, dus niet bij de vergelijking $x^2 + y^2 = 9$ of iets dergelijks. Neem dan uw toevlucht tot het *Maple-graded* vraagtype (paragraaf *Maple Graded Vraagtype* (page 45)).

Het enige nadeel van dit vraagtype is dat de student dus geen sterren hoeft te tikken voor vermenigvuldiging (maar het mag wel) en dat daardoor dus lettercombinaties niet overhoord kunnen worden. Er is wel een *Sigma*-knopje voor de student beschikbaar voor het openen van de Editor.

stion Preview

$$x = \frac{5-2y}{3}$$

Choose Question Type

- Essay
- Free Body Diagram
- List
- Maple-graded
- Math App
- **Mathematical formula**
- Multiple Choice
- Numeric

Mathematical formula:

Weighting: 1

Sub-type: Equation - e.g. $x+3y=1$

Answer: $y=(5-3*x)/2$

De volgende vergelijking moet ingetikt worden $2y+3x=5$ maar mag op verschillende manieren.

$x=(5-2*y)/3$

Figure 1.160: Vergelijkingen met het vraagtype Formula

In bovenstaande figuur moet in de *Answer code* de vergelijking in expliciete vorm ingevoerd zijn.

Dus mocht ook in de vorm $x=(5-3*x)/2$. De student mag dan elke gelijkwaardige vorm invullen om een goede beoordeling te krijgen.

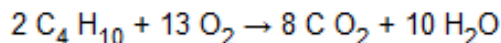
1.5.6 Chemische formules met Formula

Met de instelling *Chemical Equation* e.g. $2H_2+O_2 \rightarrow 2H_2O$ kan het correcte antwoord dus op deze manier ingevoerd worden. Het pijltje kan eenvoudig met het toetsenbord gemaakt worden en de subscripts met underscore. De student kan bij deze instelling kiezen voor tekst invoer of met behulp van de Editor. Als de student opteert voor de *Symbol Mode* (dus met de Editor), heeft dit vraagtype een goede Editor om subscript en pijltjes te maken.

De Editor is niet zo uitgebreid, maar voor scheikundeformules biedt het goede tools met de pijltjes en de subscripts.

Question Name: 05 Scheikunde - formules intikken

Tik de volgende reactievergelijking over



Equation Editor

$2 \text{C}_4\text{H}_{10} + 13$

a_b

a^b

•

→

←

↔

This question accepts chemistry formulas and equations.
[Help](#) | [Change Entry Style](#)

Figure 1.161: Chemische formules

1.6 Vraagtype Numeric

Zie *Handleiding Maple T.A. Items Maken Deel A*.

1.7 Dynamische figuren

Dit onderdeel staat nu in *Handleiding Maple T.A. Items Maken Deel C*

1.8 Maple-commando's

Over alle onderstaande commando's staat informatie in het boek

Handleiding Maple 16, Metha Kamminga

ISBN: 9789039526750

Uitgeverij Academic Service

1.8.1 Algemene commando's

- `abs` De absolute waarde van een getal.
- `allvalues` Soms wordt de oplossing van een vergelijking gegeven in code. Om alle oplossingen te krijgen gebruik je dan `allvalues`.
- `assign` Om van een vergelijking een toekenning te maken.
- `coeff(expr, x, n)` Om de coëfficiënt te krijgen van x tot de macht n van een uitdrukking.
- `coeffs(expr)` geeft alle coëfficiënten van een uitdrukking in meer variabelen. Daarvan kunt u een verzameling maken en dan kijken of alle coëfficiënten in de verzameling overeenkomen. Zie ook bij verzamelingen.

- `ceil(getal)` rond naar boven af (zie ook bij `floor`).
- `collect` Om termen samen te nemen.
- `combinat[permute]([0,1,2])` geeft alle permutaties van het opgegeven lijstje.
- `combinat[randperm]([0,1,2,3])` geeft een random permutatie van de elementen van het lijstje.
- `combinat[randcomb]([0,1,2,3],2)` geeft een random combinatie van twee elementen uit het lijstje.
- `convert(formule,string)` waarmee u een formule naar een string kunt converteren. Er zijn nog meer conversies mogelijk!!!
- `degree(f,x)` geeft de graad van de polynoom in x .
- `denom(breuk)` Geeft de noemer van de breuk.
- `Digits:=5` Betekent dat Maple met 5 significante cijfers moet rekenen.
- `diff(f,x)` Differentieert de functie f naar x .
- `discrim(kwadratischevorm,x)` berekent de discriminant van de kwadratische vorm in x .
- `dsolve(dv,onbekende functie)` Lost een differentiaalvergelijking op naar de onbekende functie.
- `dsolve({dv,randvoorwaarde}, onbekende functie)`
- `eval(F,x=1)` evalueert F voor de waarde van x gelijk aan 1.
- `evalb` Evalueer boolean dus true of false bijvoorbeeld.
- `evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","a")==0)`
Hiermee controleer je of het aantal aatjes in de respons gelijk is aan 0.
- `evalb(0=StringTools[Search]("factor","$RESPONSE"))`
Hiermee controleer je of er ook de lettercombinatie "factor" in de string van de respons van de student voorkomt. Als de combinatie niet voorkomt levert dat 0 en als de combinatie wel voorkomt levert Search een getal dat aangeeft op welke plek in de string de lettercombinatie begint.
- `stringresponse:=StringTools[Remove](" ", "$RESPONSE"):`
`evalb(SubString(stringresponse, 1..-1)=$stringantwoord);`
`of`
`with(StringTools); stringresponse:=Remove(IsSpace, "$RESPONSE");`
`evalb(stringresponse=$stringantw);`
Dit commando zorgt ervoor dat eerst de spaties uit de respons van de student gehaald worden en vervolgens wordt het resultaat vergeleken met het juiste antwoord als string. De spaties kunnen ook met `DeleteSpace(string)` verwijderd worden. En `SubString(.., 1..-1)` betekent het eerste karakter tot en met het laatste karakter.
- `evalc` Evalueert een complex getal naar de vorm $a+ib$ in de veronderstelling dat alle variabelen in de formule reëel zijn.
- `evalf` Evalueert een getal naar een decimaal getal (met floating point).
- `factor` Factoriseert een uitdrukking.
- `frac(breuk)` Geeft het gedeelte van de breuk dat overblijft nadat de helen er afgehaald zijn.
- `DecimalPlaces:=(x,n)->Float(round(10^n*x),-n);`
- `floor(getal)` rondt naar beneden af op geheel getal.
- `fsolve(vergelijking, onbekende, range)` Lost de vergelijking numeriek op binnen een bepaalde range.
- `igcd(getal1,getal2,...)` Betekent "greatest common divisor of integers" = ggd = grootste gemene deler voor getallen.
- `ilcm(getal1,getal2,...)` Betekent least common multiple of integers = kgv = kleinste gemene veelvoud voor getallen.
- `if $a>$b then 1 else 0 end if` Hiermee kunnen indices gemaakt worden ten behoeve van switch.
- `if $a<0 then 0 elif $a=0 then 1 else 2 end if` Hiermee kunnen indices gemaakt worden ten behoeve van switch.

- `ifactor(125)` geeft de factoren van het getal (integer)
- `Int(f,x)` of `int(f,x)` Integreert de functie f naar x . De eerste met hoofdletter geeft de integraal, de tweede geeft de geevalueerde integraal.
- `intersect` gebruik je om de doorsnede van twee verzamelingen te krijgen, dus $\{A\} \cap \{B\}$ geeft de doorsnede van de verzamelingen $\{A\}$ en $\{B\}$ (zie ook bij `union`).
- `isprime(a)` vraagt of het een priemgetal is (boolean), `nextprime(a)` geeft het volgende priemgetal, `prevprime(a)` geeft het vorige priemgetal.
- `ithprime(n)` Geeft het n -de priemgetal (het eerste is 2).
- `lcm(getal1, getal2, ...)` Geeft het kgv=least common multiple.
- `Limit(f,x=0)` of `limit(f,x=0)` Geeft de limiet voor x nadert tot 0.
- `LinearAlgebra[Equal](M1,M2)` om matrices met elkaar te vergelijken true of false.
- `LinearAlgebra[Determinant](matrix)` Om de determinant van een matrix te berekenen.
- `LinearAlgebra[GenerateMatrix]([s1,s2,s3],[x,y,z],augmented=true)` berekent van de drie vergelijkingen s_1 en s_2 en s_3 in de variabelen x , y en z de aangevulde matrix. (3 rijen en 4 kolommen).
- `LinearAlgebra[Rank](matrix)` Om de Rank van een matrix te berekenen.
- `LinearAlgebra[ReducedRowEchelonForm](matrix)` Om de gereduceerde matrix te berekenen.
- `LinearAlgebra[Transpose](A)` Geeft de getransformeerde van een matrix A .
- `maple("evalf(ScientificConstants[Constant](R))")` ; geeft de waarde van de universele gasconstante. De eenheid kun je opvragen met
-
- `map(round,A)` Geeft de afronding van alle elementen van A (matrix, vector of verzameling of lijst). Deze actie is ook te bewerkstelligen met behulp van de elementwise postfix (`~`) dus met `round~(A)`.
- `map(x->100*x,A)` Vermenigvuldigt alle elementen van A met 100, handig voor een verzameling waar een scalarvermenigvuldiging niet werkt. Zo kan elke functie gebruikt worden om alle elementen te behandelen. Deze actie is ook te bewerkstelligen met behulp van de elementwise postfix (`~`) dus met `100*~A`.
- `Matrix([a,b],[c,d])` Geeft een 2×2 matrix.
- `max(a,b,c)` bepaalt het maximum van een lijst met getallen.
- `min(a,b,c)` bepaalt het minimum van een lijst met getallen.
- `minus` om twee verzamelingen van elkaar af te trekken (zie ook `union` en `intersect`).
- `nops` geeft het aantal operanden van een object (verzameling, lijst, uitdrukking).
- `numer(breuk)` Geeft de teller van de breuk.
- `numtheory[factorset](G)` geeft de verzameling factoren waaruit het gehele getal G bestaat.
- `op(...)` geeft de operanden van een uitdrukking.
- `op(1..4,expressie)` geeft de operanden 1 t/m 4 van een uitdrukking.
- `op([1,3],expressie)` neemt eerst de eerste operand van een uitdrukking en geeft vervolgens de derde operand daarvan. Het moet gezien worden als het nesten van operanden van een uitdrukking. Dit is hetzelfde als `op(3,op(1,expressie))` ofwel de derde operand van de eerste operand van een expressie.
- `plot` Plotopdracht.
- `printf(MathML[ExportPresentation](antwoord))` Geeft de MathML code van een formule.
- `printf("tekst")` om tekst te communiceren waarbij de quotes worden weggelaten door middel van `printf`.

- `randomize():LinearAlgebra[RandomVector](6,generator=rand(1..5))` om een random vector te maken.
- `randomize():LinearAlgebra[RandomMatrix]($n,$n,density=0.75,generator=rand(-9..10))` om een random matrix te maken.
- `remove(has,[seq(seq(i*k^2,i=2..10),k=2..10)],[seq(k^2,k=2..31)])`
Van alle getallen in de lijst $i \cdot k^2$ waarbij $i = 2$ t/m 10 en $k = 2$ t/m 10 , wordt met "has" gecontroleerd of er elementen in zitten van de lijst getallen k^2 waarbij k loopt van 2 t/m 31 en daarna worden deze elementen verwijderd.
- `max(remove(has,[$g1,$g2,$g3,$g4,$g5,$g6],$grootste))` Als eerst `$grootste` is gedefinieerd als grootste van betreffende lijst getallen, dan kan deze grootste eruit weggehaald worden en vervolgens kan dan weer de grootste gekozen worden.
- `rhs(verg) lhs(verg)` Geeft de rechterkant respectievelijk de linkerkant van een vergelijking.
- `round` Rondt een getal af tot een geheel getal.
- `seq` Om een rij te maken.
- `series(f,x=0,n)` om de reeksontwikkeling van de functie f in de buurt van $x = 0$ te maken, optioneel als extra argument tot aan welke graad ontwikkeld moet worden.
- `simplify(vorm)` Vereenvoudigen van een uitdrukking.
- `solve(verg, x)` of `solve(verg, {x})` Lost een vergelijking op naar x .
- `sort` Kan een uitdrukking sorteren op volgorde van aflopende machten of bij lijsten naar volgorde van grootte (alleen van rationale getallen, met wortels gaat niet).
- `sqrt` afkorting van de square root dus de vierkanstswortel
- `StringTools[CountCharacterOccurrences]("$RESPONSE","a")`
- `StringTools[Remove]("a","$RESPONSE")`
- `StringTools[Search]("factor","$RESPONSE")` geeft een getal waar de eerste de beste karaktercombinatie in de string gevonden wordt.
- `StringTools[SearchAll]("a","$RESPONSE")` geeft een rij van getallen waar alle gezochte karakters in de string gevonden worden.
- `subs(x=3,f)` Substitueert $x = 3$ in de vorm f .
- `Sum/sum` Voor sommeren
- `type($RESPONSE,expanded)` geeft true als de \$REPONSE van de vorm de haakjes weggewerkt zijn.
- `union` wordt gebruikt voor de vereniging van twee verzamelingen (zie ook bij `intersect` en `minus`)
- `value(..)` Geeft de waarde ergens van, bijvoorbeeld van een integraal
- `Vector([1,2,3])` Geeft een kolomvector
- `[seq(seq(i*k^2,i=2..10),k=2..10)]` Geeft een lijst van waarden achtereenvolgens $i \cdot k^2$ waarbij i loopt van 2 tot en met 10 en k loopt van 2 tot en met 10 .
- `Float(round(10*69.4567),-1)`; om een getal te krijgen met één decimaal
- `Float(10*round(24.7752809),-1)`; geeft 25.0

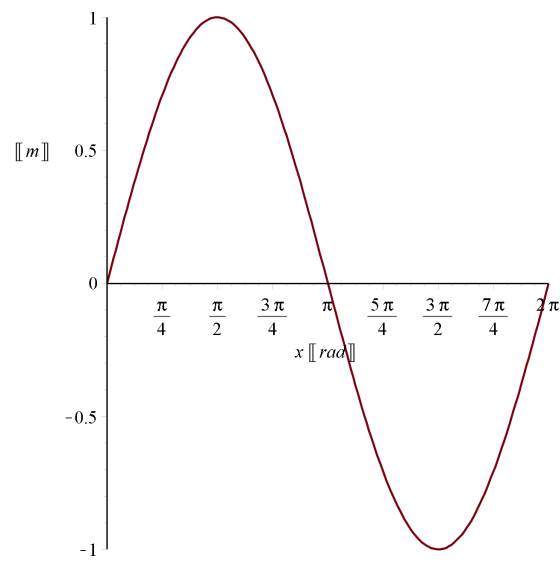
```
> zip(op,[2,1],[x^2+3,x^2+3]);
```

$$[3, x^2] \quad (1.77)$$

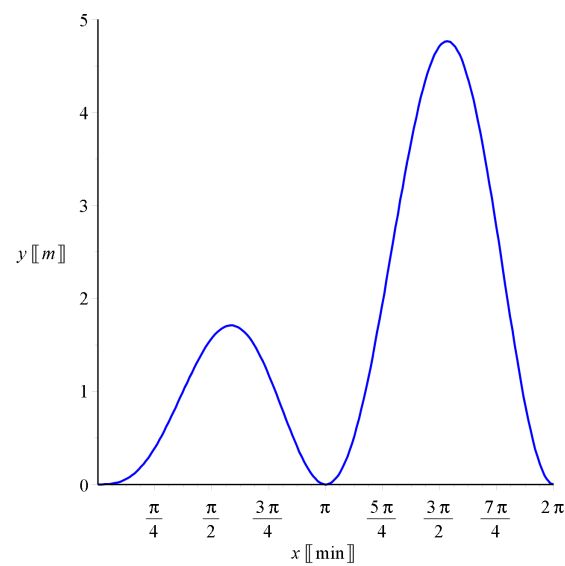
```
> zip((x,y)->x+y,[1,2,3,7],[4,4],5);
```

$$[5, 6, 8, 12] \quad (1.78)$$

```
> plot(sin(x)*Unit('m'), x = 0 .. 2*Pi*Unit('rad'));
```



```
> plot(sin(x)^2*x, x = 0 .. 2*Pi, y=0..5, useunits = [Unit('min'), Unit('m')], color = blue);
```



```
> ?zip
```

Index

