

Handleiding Möbius Items maken deel B

**Copyright © Metha Kamminga
januari 2023**

Handleiding Möbius Items maken deel B

Contents

1 Möbius Toets items maken deel B	1
1.1 Informatie en downloads	1
1.2 Formules met Möbius	1
1.2.1 Inleiding	1
1.2.2 Instellingen van het formule-invulveld	1
1.2.2.1 Inleiding bij de verschillende instellingen voor formules	1
1.2.2.2 Vraagtype Maple-graded. Instelling Maple syntax en Text entry only	2
1.2.2.3 Vraagtype Maple-graded Instelling Maple syntax en Symbol entry only	4
1.2.2.4 Vraagtype Maple-graded Instelling Formulatext	6
1.2.2.5 Vraagtype Formula instelling Formula-e.g. $e^x \sin(x^2)$	7
1.2.2.6 Belangrijke tip	9
1.2.3 Formules voorbereiden	9
1.2.3.1 Inleiding	9
1.2.3.2 Formules en getallen voorbereiden in het Algorithm	10
1.2.3.3 Formules in de tekst van de vraag	12
1.2.3.4 Wat is MathML-code	12
1.2.3.5 Vier manieren om MathML-code te maken	13
1.2.3.6 MathML-code programmeren in rubriek Algorithmtext	13
1.2.3.7 Automatische vereenvoudiging tegenhouden	18
1.2.3.8 Een functie definiëren en functiewaarden genereren	23
1.2.3.9 MathML met de Equation Editor	24
1.2.3.10 MathML-code met MathType	26
1.2.3.11 LaTeX	27
1.2.3.11.1 Variabelen, spaties en tekst	29
1.2.3.11.2 Superscript en subscript	30
1.2.3.11.3 Vectoren en onderstrepingen	30
1.2.3.11.4 Vectoren en matrices	31
1.2.3.11.5 Stelsels vergelijkingen (haken en uitlijning)	32
1.2.3.11.6 Functies en commando's cursief en niet cursief	32
1.2.3.11.7 Breuken en wortels	33
1.2.3.11.8 Ongelijkheden en speciale tekens	33
1.2.3.11.9 Limieten, sommatie en producten	34
1.2.3.11.10 Differentiëren en integreren	35
1.2.3.11.11 Stuksgewijze functies	36
1.2.3.11.12 Haakjes en pijltjes	36
1.2.3.11.13 Tekens en Griekse letters	37
1.3 Vraagtype Maple-graded	37
1.3.1 Inleiding	37
1.3.2 Algemene structuur van Maple-graded vraagtype	38
1.3.3 Algorithme van de gehele vraag	40
1.3.4 Question Text van het Maple-graded antwoordveld	41
1.3.5 Answer van het Maple-graded antwoordveld	41
1.3.6 Grading Code van het Maple-graded antwoordveld	42
1.3.7 Expression type van het Maple-graded antwoordveld	43
1.3.8 Maple Repository van het Maple-graded antwoordveld	43
1.3.9 Plotting Code van het Maple-graded antwoordveld	44
1.3.10 Custom Previewing van het Maple-graded antwoordveld	45
1.3.10.1 Andere mogelijkheden voor de custom preview:	46
1.3.11 Feedback van de gehele vraag	46
1.3.12 Feedback van het antwoordveld voor How did I do	47
1.4 Tips voor de Grading Code	49

1.4.1 Inleiding	49
1.4.2 Getallen overhoren met Maple-graded	51
1.4.3 Wanneer zijn formules gelijk aan elkaar?	51
1.4.4 Werken met StringTools	55
1.4.5 Exponenten Logaritmen en Pi	58
1.4.6 Automatische vereenvoudiging bij Preview tegenhouden	62
1.4.7 Complex numbers	62
1.4.8 Uitdrukkingen met de letter I	64
1.4.9 De Analyse van de Response	64
1.4.10 Verzamelingen	67
1.4.11 Uitdrukkingen met Coëfficiënten en operanden	69
1.4.12 Breuken	75
1.4.13 Wortelvormen	82
1.4.14 Vergelijkingen	90
1.4.15 Stelsels vergelijkingen	94
1.4.16 Ongeordende en geordende lijsten	99
1.4.17 Ongelijkheden	103
1.4.18 Symbolen en Griekse letters	106
1.4.19 Differentialen en differentiaalvergelijkingen	108
1.4.20 Integralen	113
1.4.20.1 Onbepaalde integraal	113
1.4.20.2 Bepaalde integraal en verschillende schrijfwijzen	114
1.4.20.3 Meervoudige integralen	117
1.4.20.4 De student voert de integraal in	118
1.4.21 Matrices en Vectoren	125
1.4.21.1 Inleiding	125
1.4.21.2 De matrix en de vector definiëren	126
1.4.21.3 Analyse van de invoer	128
1.4.21.4 Voorbeeld	130
1.4.21.5 Random matrix	133
1.4.21.6 Vectoren	133
1.5 Vraagtype Mathematical Formula	133
1.5.1 Inleiding	133
1.5.2 Formula - e.g. $e^x \sin(x^2)$	135
1.5.3 Formula without Simplification	135
1.5.4 Formula that matches responses to within +C	137
1.5.5 Geordende en ongeordende Lijsten en Vectoren met Formula	138
1.5.6 Eenvoudige vergelijkingen	139
1.5.7 Chemische formules met Formula	140
1.6 Vraagtype Numeric	141
1.7 Dynamische figuren	142
1.8 Maple-commando's	142
1.8.1 Algemene commando's	142
Index	149

List of Figures

Figure 1.1: Instellingen bij het Maple-graded vraagtype	2
Figure 1.2: De Preview ter controle van de invoer	3
Figure 1.3: Edit Response Area van Maple-graded vraagtype	3
Figure 1.4: Formuleinstelling Maple Syntax Symbolic entry only	4
Figure 1.5: title of the figure	6
Figure 1.6: De Preview met Formula-instelling	6
Figure 1.7: De Editor bij Formula instelling	7
Figure 1.8: Sub-types van het Mathematical Formula vraagtype	8
Figure 1.9: Preview van de Mathematical Formula vraag	8
Figure 1.10: Editor van het vraagtype Mathematical Formula	9
Figure 1.11: Formules in Algorithm	10
Figure 1.12: Negatieve waarden van de variabelen	11
Figure 1.13: Gedaante van getallen	11
Figure 1.14: MathML-code	13
Figure 1.15: Formule en MathML voorbereiden in Algorithm	14
Figure 1.16: Verschillende manieren om MathML-code te maken	15
Figure 1.17: MathML met subscript en lettercombinaties	16
Figure 1.18: MathML-Formules met Maple gemaakt	16
Figure 1.19: Gebruik van quotes bij de MathML conversie van Maple	17
Figure 1.20: MathML coderen met combinaties van methoden	17
Figure 1.21: MathML voorbereiden in gedeelten	18
Figure 1.22: Automatische vereenvoudiging tegenhouden	19
Figure 1.23: Gebruik van quotes bij de MathML conversie van Maple	19
Figure 1.24: MathML en quotes	20
Figure 1.25: MathML en quotes	21
Figure 1.26: Imaginaire eenheid	21
Figure 1.27: Custom PlottingCode	22
Figure 1.28: Formules met quotes	23
Figure 1.29: Een functie definiëren	24
Figure 1.30: De Equation Editor	25
Figure 1.31: Vernieuwde Equation Editor in Möbius 10	25
Figure 1.32: MathML-code aanpassen in de broncode van de vraag	26
Figure 1.33: Met MathType de MathML-code maken	27
Figure 1.34: LaTeX met de Equation Editor	28
Figure 1.35: voorbeeld met LaTeX	29
Figure 1.36: LaTeX in het Algorithme voorbereiden	29
Figure 1.37: LaTeX met variabelen spaties en tekst	30
Figure 1.38: LaTeX met superscript en subscript	30
Figure 1.39: LaTeX met vectoren en onderstreping	30
Figure 1.40: Kolomvectoren en matrices met LaTeX	31
Figure 1.41: Matrixvergelijking	32
Figure 1.42: Aangevulde matrix	32
Figure 1.43: Stelsel vergelijkingen uitlijnen	32
Figure 1.44: LaTeX met functies en commando's	33
Figure 1.45: LaTeX met cursief en niet-cursief	33
Figure 1.46: LaTeX met breuken en wortels	33
Figure 1.47: LaTeX met limieten	34
Figure 1.48: LaTeX met sommatie en product	34
Figure 1.49: LaTeX met gecentreerde formule	35
Figure 1.50: LaTeX met integralen	35
Figure 1.51: LaTeX met kringintegraal	35

Figure 1.52: LaTeX en differentiëren	36
Figure 1.53: partiële differentiatie	36
Figure 1.54: stuksgewijze functie	36
Figure 1.55: LaTeX en haakjes	36
Figure 1.56: LaTeX en tekens en Griekse letters	37
Figure 1.57: Voorbeeld van een Maple-graded vraag	38
Figure 1.58: Maple-graded vraag aanmaken	39
Figure 1.59: Dialoogscherf van de Maple-graded vraag	39
Figure 1.60: Het algoritme van een Maple graded vraagtype	40
Figure 1.61: Tekst van de Maple Graded vraag	41
Figure 1.62: Het correcte antwoord in 2D	41
Figure 1.63: Grading Code van de Maple-graded vraag	42
Figure 1.64: Instellingen voor Maple syntax bij Maple-graded vraagtype	43
Figure 1.65: Plotting Code in de Maple-graded vraag	44
Figure 1.66: Custom Previewing Code	45
Figure 1.67: Feedback bij de Maple-graded vraag	46
Figure 1.68: Grading van een Maple-graded vraag met feedback	47
Figure 1.69: Tabblad voor Feedback bij het invulveld	48
Figure 1.70: Feedback invoeren voor bij het invulveld	48
Figure 1.71: How did I do en feedback	49
Figure 1.72: Instellingen met Maple Syntax	55
Figure 1.73: Vereenvoudigen	56
Figure 1.74: Hoe ga je om met de exponentiële functie binnen maple	60
Figure 1.75: De logaritme binnen en buiten maple	60
Figure 1.76: Verschil tussen Pi en pi	61
Figure 1.77: Tegenhouden van automatische vereenvoudiging in de Preview door gebruikmaking van de Custom Previewing Code	62
Figure 1.78: Complex numbers	63
Figure 1.79: Instellingen voor het antwoord	65
Figure 1.80: Genereren van de string van de RESPONSE	65
Figure 1.81: Het zichtbaar maken van de "\$RESPONSE" met Text entry only	66
Figure 1.82: Het zichtbaar maken van de "\$RESPONSE" met Symbol entry only	66
Figure 1.83: Het zichtbaar maken van de "\$RESPONSE" met Symbol entry only	66
Figure 1.84: Het zichtbaar maken van de "\$RESPONSE met Formula	67
Figure 1.85: Een polynoom met decimalen matchen	71
Figure 1.86: Uitdrukking met decimale coëfficiënten	73
Figure 1.87: Breuken met instelling Text entry only	76
Figure 1.88: Breuken met instelling Symbol entry only	77
Figure 1.89: Breuken invoeren in de Editor	78
Figure 1.90: Breuken invoeren als text bij de instelling Formula	78
Figure 1.91: Breuken vereenvoudigen	80
Figure 1.92: Breuken met twee invulvelden voor teller en noemer	81
Figure 1.93: Hoe worden wortels vereenvoudigd	82
Figure 1.94: De respons en de stringrespons bij verschillende manieren van intikken van de wortels	82
Figure 1.95: Wortels vereenvoudigen	83
Figure 1.96: Wortels vereenvoudigen in Algorithm	84
Figure 1.97: De wortelvorm vereenvoudigen	85
Figure 1.98: Algorithm voor wortels	85
Figure 1.99: De operanden van een wortelvorm	86
Figure 1.100: Wortels vereenvoudigen	87
Figure 1.101: Operanden van een wortelvorm	88
Figure 1.102: Wortels met breuken vereenvoudigen	89
Figure 1.103: Wortels optellen	90

Figure 1.104: Lineair systeem, vergelijking invullen	91
Figure 1.105: Vergelijkingen met decimalen	92
Figure 1.106: Vergelijking van een ellips	93
Figure 1.107: Stelsel vergelijkingen	95
Figure 1.108: Stelsel vergelijkingen met verzameling en met lijst	95
Figure 1.109: Stelsel vergelijkingen met de variabelen apart	96
Figure 1.110: Stelsel met meer oplossingen	97
Figure 1.111: Stelsel vergelijkingen met meer oplossingen Algorithm	97
Figure 1.112: Stelsel vergelijkingen met meer oplossingen numeriek	98
Figure 1.113: Algorithm van een vraag over het berekenen van snijpunten van grafieken	98
Figure 1.114: title of the figure	99
Figure 1.115: Vector als geordende lijst	102
Figure 1.116: Lineaire ongelijkheid	104
Figure 1.117: Ongelijkheid met intervallen	105
Figure 1.118: Griekse letters invoeren	107
Figure 1.119: De student tikt een differentiaalvergelijking in	109
Figure 1.120: Differentiaal symbolisch met text invoer	110
Figure 1.121: Afgeleide met subscript	111
Figure 1.122: Het graden van de oplossing van een DV	111
Figure 1.123: De oplossing van de DV	112
Figure 1.124: Hogere afgeleiden met het tekstveld	112
Figure 1.125: Hogere afgeleiden met de Editor	113
Figure 1.126: Subscript in de Editor	113
Figure 1.127: Primitiveren, de onbepaalde integraal	114
Figure 1.128: De bepaalde integraal berekenen	115
Figure 1.129: Verschillende schrijfwijzen van de bepaalde integraal	116
Figure 1.130: LaTeX en de integraand met variabele 1	117
Figure 1.131: Dubbelintegraal berekenen	118
Figure 1.132: Student voert een integraal in	119
Figure 1.133: Integraal invoeren in tekstveld	120
Figure 1.134: Inhoud kegel met integreren	121
Figure 1.135: De analyse van het antwoord van de student in de vorm van een integraal in Symbol mode	122
Figure 1.136: De analyse van het antwoord van de student in de vorm van een integraal in Text mode	122
Figure 1.137: Invoeren van een integraal symbolisch	123
Figure 1.138: Verschil in invoer met Int en int en geen verschil in de Preview	125
Figure 1.139: Het invoeren van een Matrix door de student in de Editor	126
Figure 1.140: De matrix in het algorithm	127
Figure 1.141: De definitie van een vector in het algoritme	128
Figure 1.142: De respons als correcte antwoord bekijken	128
Figure 1.143: Invoer van matrices vertaald naar de string	129
Figure 1.144: Invoer van matrices	130
Figure 1.145: Matrix invoeren	131
Figure 1.146: Echelon matrix	132
Figure 1.147: title of the figure	134
Figure 1.148: Subtypes van het vraagtype Formula	135
Figure 1.149: Limieten met wortelvorm	136
Figure 1.150: Vraagtype Formula met wortels	137
Figure 1.151: Preview met wortels	137
Figure 1.152: Ongeordende en geordende lijsten bij vraagtype Mathematical Formula	138
Figure 1.153: Question Designer Lijsten en Vectoren	139
Figure 1.154: Vergelijkingen met het vraagtype Formula	140
Figure 1.155: Chemische formules	141

Figure 1.156: Correcte antwoord van een chemische formule	141
Figure 1.157: title of the figure	142

1 Möbius Toets items maken deel B

© Metha Kamminga
Update febr. 2022

1.1 Informatie en downloads

- Home page Metha Kamminga

Meer informatie over cursussen en nieuws over Möbius op de site van Metha Kamminga:
<https://www.methakamminga.nl>.

- Deze handleiding is onderdeel van een serie:

- Items maken deel A (zonder formules)
- Items maken deel B (met formules)
- Items maken deel C (bijzondere features)
- Randomiseren
- Zie voor algemene informatie over de *Content Manager* en het overzicht van een item in de handleiding *Items maken deel A*.

1.2 Formules met Möbius

1.2.1 Inleiding

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

Het gebruik van formules is nu helemaal onafhankelijk van Java 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.
- Het aanzien van de formules in de tekst van het item is aanzienlijk verbeterd.
- Het is zelfs mogelijk om formules te toetsen, waarbij studenten zelf formules kunnen invoeren in de invulvelden.
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

1.2.2.1 Inleiding bij de verschillende instellingen voor formules

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 49).

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

U maakt 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* vraagtypes ter beschikking. Het is het vraagtype *Maple-graded* en het vraagtype *Mathematical Formula*.

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

1.2.2.2 Vraagtype Maple-graded. Instelling Maple syntax en Text entry only

Bij het vraagtype *Maple-graded* met de instelling *Maple syntax* kan de student de formule intikken met de officiële Maple 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.

Bij het *Maple-graded* vraagtype zijn twee instellingen mogelijk: *Maple syntax* en *Formula*. Er is een groot verschil tussen beide instellingen en dat wordt in de volgende paragrafen uiteengezet.

Doe eerst de volgende instelling:

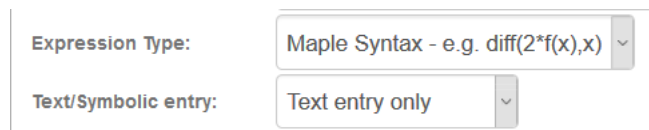


Figure 1.1: Instellingen bij het Maple-graded vraagtype

Deze instelling is qua betrouwbaarheid het meest geschikt maar heeft enkele nadelen.

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: Let op bij het previewen. U vraagt om de formule te vereenvoudigen. De student tikt de opgave over en in de *Preview* wordt de ingetikte formule automatisch vereenvoudigd. De student wordt hiermee voorgezegd! U kunt dit voorkomen door iets te programmeren in de *Custom Previewing Code*. Zie (page 42).

TIP: Als u geen maatregelen treft voor de *Grading Code* hebt u kans dat de overgetikte formule die de student niet vereenvoudigd heeft ook goedgerekend wordt.

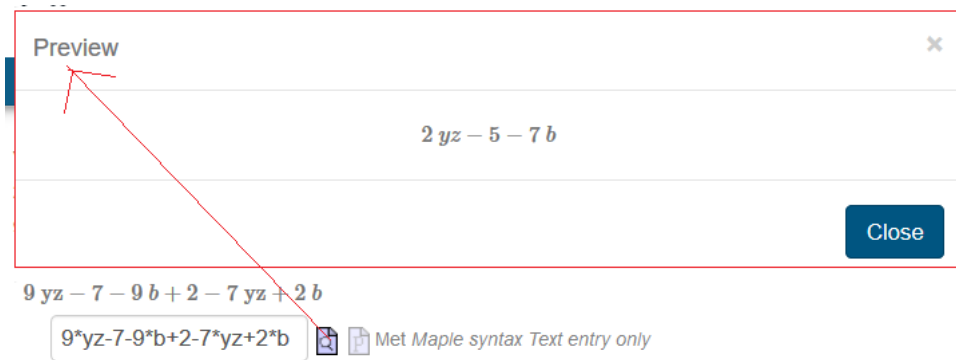


Figure 1.2: De Preview ter controle van de invoer

- De student klikt op *Preview* en ziet de vereenvoudigde formule wat niet altijd wenselijk is. Daarop moet u voorzieningen treffen in *Custom Previewing Code*. Zie (page 45)
- Met de *Grading Code* kunt u veel bereiken door te programmeren hoe het antwoord er precies uit moet zien. Zie (page 42).

Edit Response Area

Question Feedback

Maple-graded:

Weighting: 1

Answer: \$ans1
(referenced when grading as \$ANSWER)

Grading Code:
`evalb($RESPONSE-($ans1)=0) and
evalb(nops({StringTools[SearchAll]
("$z","$RESPONSE")})<2) and
evalb(nops({StringTools[SearchAll]
("$x" "$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: pri

Figure 1.3: Edit Response Area van Maple-graded vraagtype

In de figuur is te zien dat er nogal wat voorzieningen nodig zijn opdat de student in de *Preview* niet de vereenvoudigde vorm ziet en dat overtikken van de vraag niet tot een 100% score leidt.

TIP : voor het invoerveld: Als het antwoord erg lang is, voer dan bij de *Custom CSS* de volgende code in (te beginnen met een punt) zodat het invoerveld wat breder wordt.
`.response input {width:250pt;}`

1.2.2.3 Vraagtype Maple-graded Instelling Maple syntax en Symbol entry only

Bij *Expression Type* kiest u weer voor *Maple Syntax* maar nu met *Symbol entry only*.

Edit Response Area

Question Feedback

Maple-graded:

Weighting: 1

Answer: `printf(MathML[ExportPr`
(referenced when grading as \$ANSWER)

Grading Code: `evalb($RESPONSE=$ans1) and
evalb(nops({StringTools[SearchAll]
("$z", "$RESPONSE")})<2) and
evalb(nops({StringTools[SearchAll]
("$x" "$RESPONSE")})<2)`

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

Text/Symbolic entry: Symbolic entry only

Optional:

Maple Repository: `Maple Repository`

Plotting Code:

Custom Previewing Code:

Figure 1.4: Formuleinstelling Maple Syntax Symbolic entry only

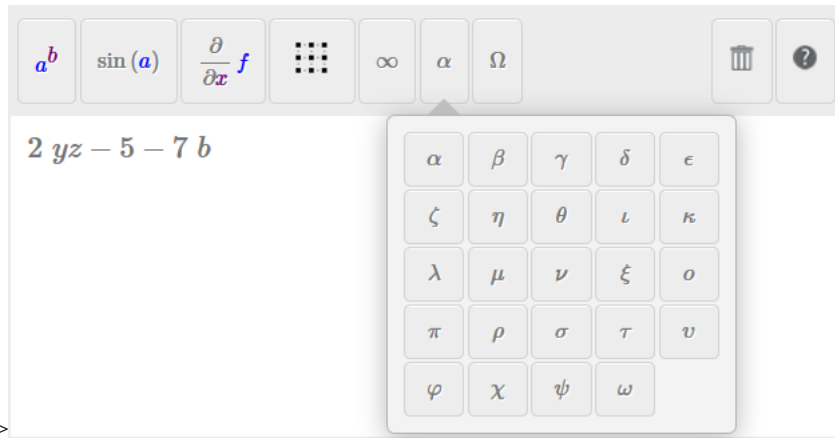
Hier ziet u dat de uitgebreide *Grading Code* ook weer gebruikt kan worden om te voorkomen dat het overgetikte antwoord ook goedgerekend wordt.

Echter bij *Custom Previewing Code* hoeft u niets in te vullen want de editor die aangeboden wordt met deze instellingen is van zichzelf al een editor. Er vindt dus geen automatische vereenvoudiging plaats.

TIP: Omdat de student geen codes met sterretjes en haakjes hoeft in te voeren biedt u voor het correcte antwoord bij *Answer* de 2D formule aan (in MathML) met de volgende code:

```
printf(MathML[ExportPresentation]($ans));
```

```
gureTitle>Maple syntax in Symbol Mode met Editor</
```



```
MFigureTitle>
MFigure>
```

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!

Een aantal tips bij gebruik van deze Editor

- De student moet wel degelijk óók leren omgaan met deze Editor. Integralen, differentiaal, subscript, matrices, Griekse letters en dergelijk is allemaal mogelijk.
- 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 yz 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 toepassingssfeer.
- 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 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.
Echter als de student $7b$ zonder spatie tikt, wordt dat in deze Editor weer wél als vermenigvuldiging gezien. In geval de student $b7$ intikt wordt dit juist weer níet als vermenigvuldiging opgevat en $b7$ kan dan ook als één geheel worden gebruikt in een formule.
- 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 de knop van het prullenmandje voor vernieuwen (rechts in de Editor) gebruikt om de Editor weer even schoon te vegen, zodat er geen onzichtbare codes meer in blijven zitten.
- 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.
Zie verder in paragraaf *Symbolen en Griekse letters* over het graden van formules met deze karakters (page 106).

- Alle standaardfuncties worden op de bekende manier ingevoerd zoals $\sin(x)$, $\exp(x)$ en $\ln(x)$, met haakjes dus.
- **TIP:** Een invoerveld met de editor van de Maple-syntax is vaak onnodig hoog. Neem dan de volgende code in de Custom CSS.
`.mwEquationEditor {height:100px !important}`

1.2.2.4 Vraagtype Maple-graded Instelling Formulertext

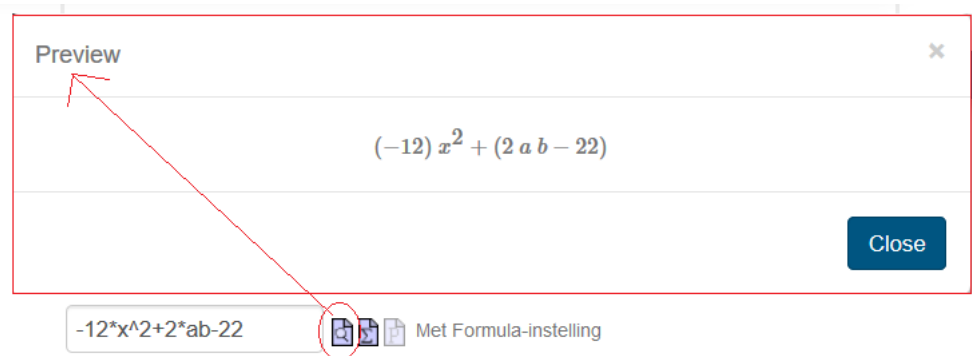
Bij het vraagtype *Maple-graded* kunt u de instelling voor het intikken van het antwoord op *Formula* zetten. U moet hierbij wel bedenken dat u deze instelling alleen gebruikt in bijzondere omstandigheden, want deze instelling kan beslist niet veel formules aan en is in de meeste gevallen zelfs af te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

Expression Type: Formula - e.g. e^x sin(x^2) ▼

Text/Symbolic entry: Student can choose ▼

Figure 1.5: title of the figure

Bij deze instelling ziet het er voor de student iets anders uit:



Preview

$$(-12)x^2 + (2ab - 22)$$

Close

-12*x^2+2*ab-22 Met Formula-instelling

Figure 1.6: De Preview met Formula-instelling

- In de *Preview* komen veel overbodige haakjes en ab wordt niet gezien als een lettercombinatie. Het systeem ziet ab automatisch als $a*b$ (het zet er ongevraagd zelf een ster tussen). Dus dit soort formules kunnen beslist niet overhoord worden met deze instelling.
- Het is hierbij mogelijk dat de student zelf kiest (Σ -knop) voor *Text Mode* óf voor *Symbol Mode* met de beschikbare knop.

Nadat de student op de Σ -knop heeft geklikt wordt de vraag opnieuw geladen en ziet hij de editor.

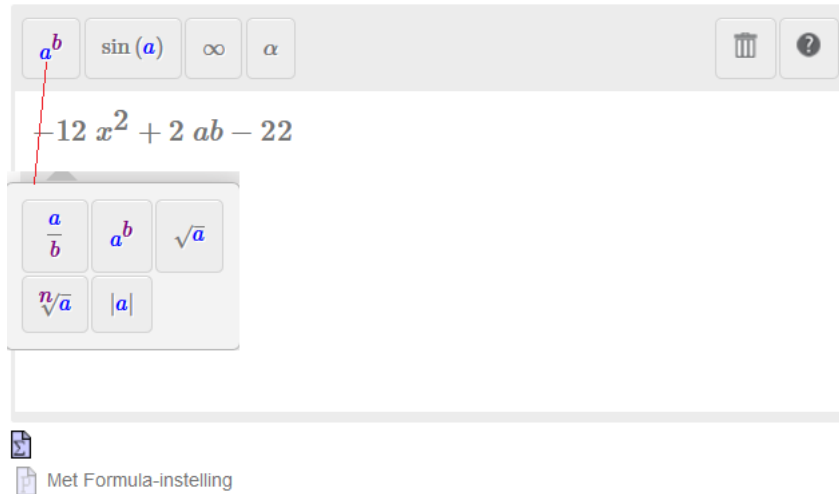


Figure 1.7: De Editor bij Formula instelling

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.

1.2.2.5 Vraagtype Formula instelling Formula-e.g. $e^x \sin(x^2)$

Als u formules wilt overhoren kiest u misschien voor het vraagtype *Mathematical Formula*. Echter dit vraagtype is daarvoor ongeschikt! Bij het vraagtype *Maple-graded* was de instelling met *Formula* al een stuk zwakker dan met *Maple Syntax*. Met het vraagtype *Mathematical Formula* doet u nóg een stap verder achteruit wat betreft de mogelijkheden voor formules.

The screenshot shows the 'Edit Response Area' window with the 'Question' tab selected. On the left, under 'Choose Question Type', 'Mathematical Formula' is highlighted. In the center, the 'Sub-type:' dropdown menu is open, showing various options like 'Formula - e.g. e^x sin(x^2)', 'Equation - e.g. x+3y=1', and 'Chemical Equation - e.g. 2H_2 + O2 -> 2H_2O'. The 'Formula' option is selected. The 'Weighting' is set to 1. At the bottom right, there are 'OK' and 'Cancel' buttons.

Figure 1.8: Sub-types van het Mathematical Formula vraagtype

Er is echter wel een aantal andere *sub-types* die de moeite waard zijn om te gebruiken.

De student ervaart het invulveld en de *Preview* bijna hetzelfde als bij *Maple-graded* vraagtype met instelling *Formula* namelijk weer een formule met veel overbodige haakjes.

The screenshot shows a 'Preview' window with the formula $(-12) x^2 + (2 a b - 22)$. Below the preview, the input field contains the text `-12*x^2+2*ab-22`. A red arrow points from the input field to the preview window. To the right of the input field, there is a small icon and the text 'Met vraagtype Formula'. A 'Close' button is located at the bottom right of the preview window.

Figure 1.9: Preview van de Mathematical Formula vraag

De karaktercombinatie *ab* wordt hier ook gezien als *a*b* (net als bij de instellingen *Formula* van de *Maple-graded* vraag).

Ook hier kan de student zelf beslissen om met de Σ -knop over te schakelen naar de editor. Hij ziet echter dan een editor met nóg minder mogelijkheden, zoals in de figuur hier onder.

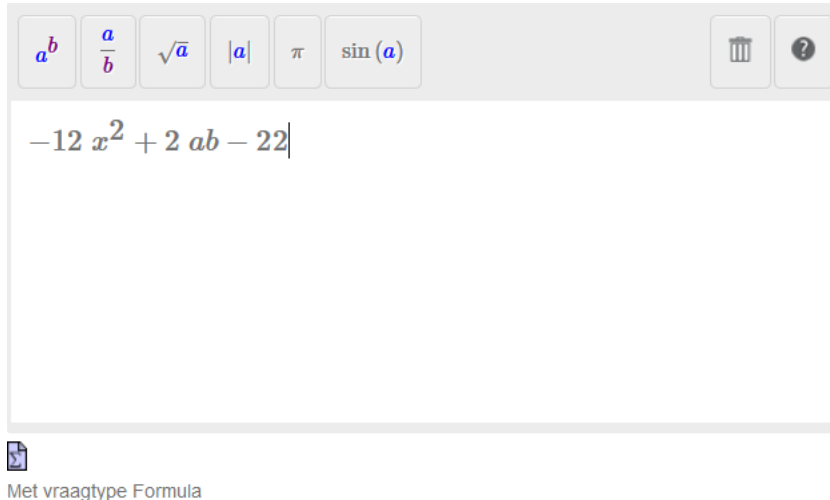


Figure 1.10: Editor van het vraagtype Mathematical Formula

De knoppen van deze editor zijn niet uitklapbaar en dit zijn alle knoppen die mogelijk zijn bij deze instelling.

Karaktercombinaties zijn hier ook niet mogelijk. Namelijk ab wordt gezien als $a*b$ ook al wordt er geen spatie of ster of stip tussen getikt.

1.2.2.6 Belangrijke tip

BELANGRIJKE TIP: De instelling met **Formula** voor het toetsen van formules in een *Maple-graded* vraagtype is eigenlijk **absoluut af te raden!**

Helemaal af te raden is vraagtype **Mathematical Formula**.

U kunt formules met lettercombinaties met deze instellingen niet toetsen! Ook wordt de student 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*.

Ook is er geen programmering met *Grading Code* mogelijk om dingen naar uw hand te zetten

1.2.3 Formules voorbereiden

1.2.3.1 Inleiding

Voor formules op het scherm werken we met HTML-code of MathML-code of 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 LaTeX-code te gebruiken. Zie voor meer informatie en voorbeelden in paragraaf *Formules met HTML* (page 12) en voor LaTeX in paragraaf *LaTeX* (page 27).

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 <https://www.methakamminga.nl/MapleTA/formules.pdf>.

In de volgende paragrafen wordt uiteengezet hoe u MathML-code genereert.

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.

1.2.3.2 Formules en getallen voorbereiden in het Algorithm

Formules

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	$-8+5\text{Mk}-6/-8\text{Q}$
formule2	$-8.0+(((5.0(\text{M}))*(\text{k}))(-0.75(\text{Q})))$
formule3	$-8+5\text{Mk}+3/4\text{Q}$

Figure 1.11: Formules in Algorithm

In bovenstaande figuur is het volgende op te merken:

- De variabele \$formule1 is gemaakt mét quotes zodat Mk als één geheel wordt beschouwd en de deling 6/8 wordt niet vereenvoudigd.
Daarbij is er wel een nadeel van de mintekens in -6/-8. Als het neerkomt op berekeningen hiermee is het handig om haakjes te zetten om de variabelen die eventueel negatief kunnen worden.
- 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 rekent 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 M_k 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).

TIP: 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.

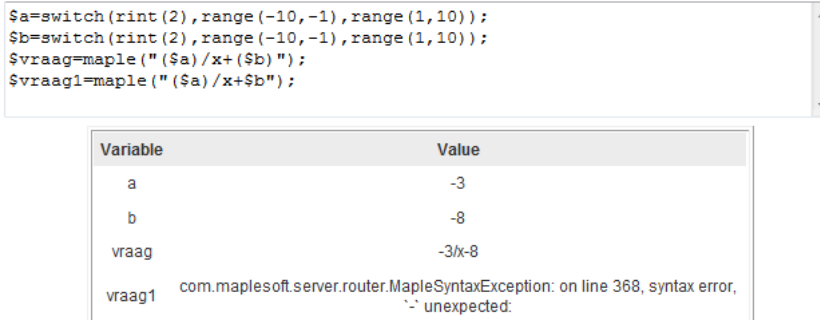


Figure 1.12: Negatieve waarden van de variabelen

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

Getallen

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.

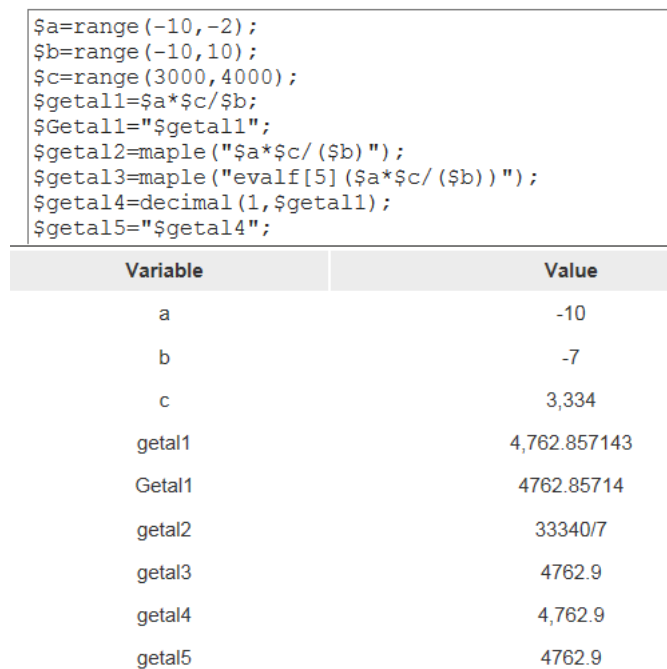


Figure 1.13: Gedaante van getallen

In bovenstaande figuur ziet u hoe u de duizendtal-separator kwijt kunt raken:

óf met Maple het getal genereren óf met quotes er omheen.

Met het commando `evalf` in Maple kan het aantal significante cijfers gestuurd worden (niet het aantal decimalen).

Het aantal decimalen is op verschillende manieren te sturen:

- Afdwingen van een bepaalde manier van decimalen noteren met `numfmt` (numeriek volgens format)
`$test1=numfmt("#.00", 20.9)` geeft als resultaat 20.90 (numeriek volgens format).
`$test2=numfmt("#0.000", 3/4)` geeft als resultaat 0.750.
- U kunt ook Maple inschakelen om de afronding te doen
`$test3=maple("Float(round(100*20.9), -2)");` dat 20.90 oplevert.
`$test4=maple("Float(round(1000*3/4), -3)");` dat 0.750 oplevert.
- Het systeem zelf kan ook afronden op de volgende manier:
`$test5=decimal(2, 20.90123);` geeft 20.9
`$test6=decimal(2, 20.9123);` geeft 20.91
LET OP: Deze manier van afronden geeft niet altijd het gewenste resultaat. Er is nog een alternatief voor een betere afronding met het volgende commando:
`$test7=decimal(3, 5*1.7^4);`
geeft **foute afronding**: 41.76 van het juiste getal 41.7605
`$test8=maple("MapleTA:-Builtin:-decimal(3, 5*1.7^4)");`
geeft de **juiste afronding** 41.761 van het getal 41.7605.

1.2.3.3 Formules in de tekst van de vraag

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 27) 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 *Quick Symbols* waarmee u verschillende tekens in de tekst kunt opnemen.

TIP: In de paragraaf *Getallen, tekens, links en lettertypen in de presentatie van de vraag* in de *Handleiding Möbius 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* maar met LaTeX: bijvoorbeeld `\(\alpha\)`. Zie paragraaf (page 27)

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 24) 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 27).

1.2.3.4 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.

Möbius 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><mfenced><mrow><mrow><mi>B</mi><mo>&InvisibleTimes;</mo><mi>M</mi></mrow><mo>+
</mo><mrow><mi>C</mi><mo>&InvisibleTimes;
</mo><mi>m</mi></mrow></mrow></mfenced></mrow><mo>&InvisibleTimes;
</mo><mi>h</mi></mrow></mfrac></mrow></math>
```

Figure 1.14: MathML-code

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

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

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 genereren die allemaal hieronder uitgelegd worden.

1.2.3.5 Vier manieren om MathML-code te maken

De MathML-code kunt u in Möbius op vier manieren maken.

- In de rubriek *Algorithm* met behulp van een Maple-commando (meest efficiënte manier).
`$displayvraag=maple("printf(MathML[ExportPresentation]($a/x+$b))");`
 Zie paragraaf MathML-code programmeren in rubriek *Algorithm text (page 13)*
- In de rubriek *Algorithm* met behulp van de mathml-functie van Möbius.
`$displayvraag=mathml("$a/x+$b");`
 Zie verder in paragraaf *MathML-code programmeren in rubriek Algorithm* Zie paragraaf MathML-code programmeren in rubriek *Algorithm text (page 13)*
- In de rubriek *Algorithm* met behulp van LaTeX. U kunt deze LaTeX-code tussen quotes in het Algorithm voorbereiden en testen:
`$displayvraag="\(\frac{$a}{x}+$b\)");`
 Maar u kunt ook deze code direct in de tekst van de vraag intikken (zonder de quotes).
 Zie paragraaf LaTeX (page 27) voor meer informatie.
- Met de *Equation Editor* is het vaak wat langdradig werk om een formule te bouwen. U kunt zelfs zonder deze editor.
 Bovendien is deze Editor niet heel erg uitgebreid.
 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 26).

1.2.3.6 MathML-code programmeren in rubriek Algorithmtext

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

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

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 met de volgende maple-opdracht:

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

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	$1/3 M^2 (-5 P v h + M H) / P v h$
displayopl	$\frac{1}{3} \frac{M (-5 P v h + M H)}{P v h}$

Figure 1.15: 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.

Bij het genereren van de oplossing \$opl, laat u 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 kunt u dan weer gebruiken 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 volgende commando: `mathml("...")` zoals in het volgende script.

Deze methode is nogal primitief en heeft maar weinig mogelijkheden.

```

$a=range(2,5);
$b=range(3,5);
$vraag="$a/x+$b";
$displayvraag=mathml("$vraag");
Probeer ook eens $displayvraag = mathml("$vraag","nosimplify");
om ongewenste vereenvoudigingen tegen te houden.

```

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 gewoon gerekend worden.

Echter 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:

1	<code>\$a:=range(3,10);</code>
2	<code>\$b:=range(3,10);</code>
3	<code>\$test1:=maple("printf(MathML[ExportPresentation](\$a+\$b/\$a*A))");</code>
4	<code>\$test2:=mathml("\$a+\$b/\$a*A");</code>
5	<code>\$test3:=mathml("\$a+\$b/\$a*A", "nosimplify");</code>
6	<code>\$test4:=mathml(\$a+\$b/\$a*A);</code>
7	

Variable	Value
a	8
b	6
test1	$8 + \frac{3}{4} A$
test2	$8 + \frac{3}{4} A$
test3	$8 + \frac{6}{8} A$
test4	$8 + 0.75 A$

Figure 1.16: 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 $a+b/a \cdot A$ 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 1.15 (page 14)*. Zie ook paragraaf *Formules in Algorithm (page 10)*.

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 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 differentiaal. 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 1.17: MathML met subscript en lettercombinaties

In de volgende figuur ziet u nog wat voorbeelden van andere formules waarbij de manier om MathML-code te genereren met behulp van Maple veel krachtiger is:

```
$testdiff1:=maple("printf(MathML[ExportPresentation](Diff(f(x),x))");
$testdiff2:=maple("printf(MathML[ExportPresentation](Diff(f,x))");
$testint1:=maple("printf(MathML[ExportPresentation](Int(f,x=0..10))");
$testwortel:=maple("printf(MathML[ExportPresentation](sqrt(x))");
$testmatrix:=maple("printf(MathML[ExportPresentation](Matrix([[2,3],
[a,b]]))");
$testvector:=maple("printf(MathML[ExportPresentation](Vector([2,3,a]))");
```

Variable	Value
testdiff1	$\frac{d}{dx} f(x)$
testdiff2	$\frac{\partial}{\partial x} f$
testint1	$\int_0^{10} f dx$
testwortel	$\sqrt{x}$
testmatrix	$\begin{pmatrix} 2 & 3 \\ a & b \end{pmatrix}$
testvector	$\begin{pmatrix} 2 \\ 3 \\ a \end{pmatrix}$

Figure 1.18: 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](...) ) ")
```

Bijvoorbeeld de grenzen van de integraal komen mooier boven en onder het teken te staan als u spaties gebruikt.

TIP: 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 1.19: Gebruik van quotes bij de MathML conversie van Maple

TIP: Soms is een *combinatie* van de functie `mathml("...")` én `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)))");
$steller:=maple("numer((2*x-$c)/($a+$c-$d))");
$noemer:=maple("denom((2*x-$c)/($a+$c-$d))");
condition:=not(eq($noemer,1));
$breuk:=mathml("($steller)/($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 1.20: 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. Neem eventueel uw toevlucht tot het maken van de formule met LaTeX-code.

TIP: Tegenwoordig is er nog een extra mogelijkheid gekomen om de automatische vereenvoudiging tegen te houden. Zie daarvoor paragraaf (page 18).

TIP: Nog een mogelijkheid als de combinatie van twee (of meer) formules in de tekst van de vraag bij de voorbereiding in het *Agorithm* 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:

$$\frac{11Q}{19} \times \frac{6}{10(Q+6)}$$

\$expr11 \times \$expr12

Figure 1.21: 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 kunt u dat oplossen.

(Het teken $\times$ maakt u met de toetscombinatie [Alt Gr][=].)

TIP: Heel veel oplossingen voor problemen met het presenteren van formules kunt u bereiken met LaTeX zie paragraaf over LaTeX (page 27).

1.2.3.7 Automatische vereenvoudiging tegenhouden

Er zijn meer manieren om automatische vereenvoudiging tegen te gaan van een formule die u wilt presenteren in uw item.

- 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 1.22: 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 1.23: Gebruik van quotes bij de MathML conversie van Maple

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

- 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 1.24: 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 met `maple("...")` wordt de evaluatie van deze berekening $5!$ tegen gehouden. Immers $5! = 120$.

- Vier manieren om de vereenvoudiging tegen te houden:
 - Met `mathml("...")`
Is vaak niet aan te bevelen. Er komen veel haakjes in de vorm voor.
 - Met de inert form:

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

 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.
 - Met quotes:

```
$test1:=maple("printf(MathML[ExportPresentation](3*x*``+( $a)*x))");
```

 Hier laat u maple geloven dat de termen niet gelijksoortig zijn door een van de termen met een "spatie" te vermenigvuldigen. Let dan wel op dat de spatie 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.
 - Als uiterste middel kunt u ook nog gebruikmaken van LaTeX. Daarbij is het niet zo handig om met variabelen te werken die zowel positief als negatief kunnen zijn. Zie voor meer informatie daarover in paragraaf (page 27).

```

$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 1.25: MathML en quotes

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 1.26 (page 21)). 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 27)).

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 1.26 (page 21)).

```

$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	$3P^*L/E/I`$
antw1display	$\frac{3PL}{EI}$
test1	$Re + R_e$

Figure 1.26: 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));`



Figure 1.27: Custom PlottingCode

Zie voor het rekenen en graden met complexe getallen in paragraaf *Uitdrukkingen met de letter I* (page 64).

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 1.28: Formules met quotes

1.2.3.8 Een functie definiëren en functiewaarden genereren

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");
$f=maple("x->$functie");
$fdisplay=maple("printf(MathML[ExportPresentation]($f))");
$f0=maple("apply($f,0)");
$f2=maple("apply($f,2)");
$g=maple("g:=x->$functie:[g(1),g(2),g(3)]");
$g2=switch(1,$g);

```

1	<code>\$a:=range(1,4);</code>
2	<code>\$b:=switch(rint(2),range(-4,-1),range(1,4));</code>
3	<code>\$functie:=switch(rint(5),"\$a*x^2+(\$b)", "\$b*x^2+\$a", "(\$a*x+(\$b))^2", "\$a*x^2+(\$b)", "x^2");</code>
4	<code>\$f:=maple("x->\$functie");</code>
5	<code>\$fdisplay:=maple("printf(MathML[ExportPresentation](\$f))");</code>
6	<code>\$f0:=maple("apply(\$f,0)");</code>
7	<code>\$f2:=maple("apply(\$f,2)");</code>
8	<code>\$g:=maple("g:=x->\$functie:[g(1),g(2),g(3)]");</code>
9	<code>\$g2:=switch(1,\$g);</code>
10	
11	

Variable	Value
a	1
b	3
functie	$(1*x+(3))^2$
f	$x \rightarrow (x+3)^2$
fdisplay	$x \rightarrow (x+3)^2$
f0	9
f2	25
g	[16, 25, 36]
g2	25

Figure 1.29: Een functie definiëren

In de figuur ziet u dat u een functie kunt definiëren als procedure (met pijltjesnotatie).

Als u binnen maple blijft is meteen een lijstje van functiewaarden te genereren zoals bij de functie \$g.

Maar als u eerst apart de functie definieert zoals met \$f. En daarna functiewaarden wilt genereren is het maple commando `apply` nodig om de functie toe te passen op een getal.

1.2.3.9 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 Möbius zoals in paragraaf *Werken met quotes* (page 18) is besproken.

We hebben altijd nog de mogelijkheid van de *Equation Editor* achter de hand, Figure 1.30 (page 25). 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*).

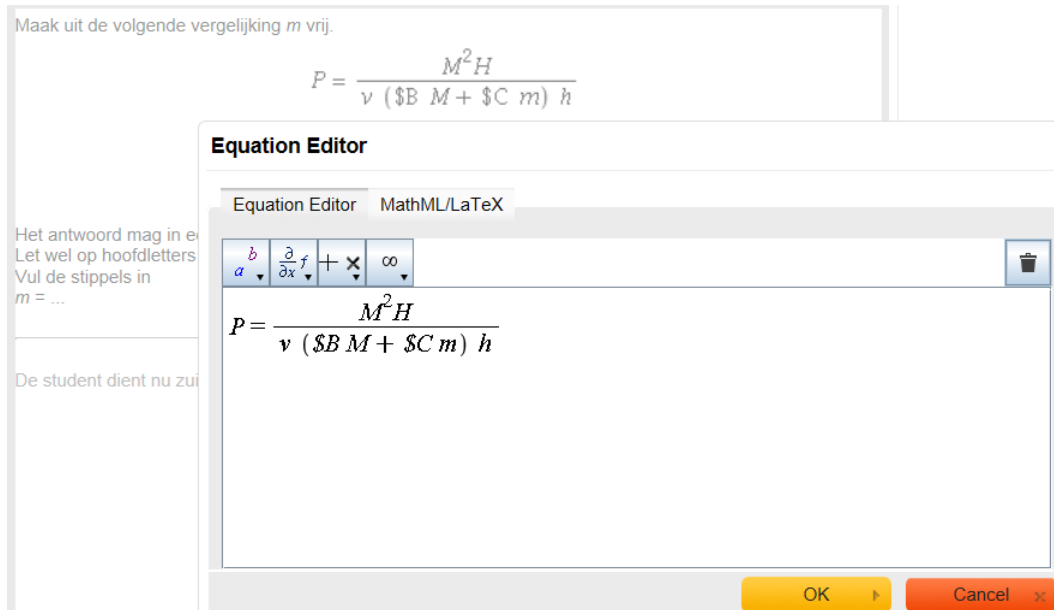


Figure 1.30: De Equation Editor

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.

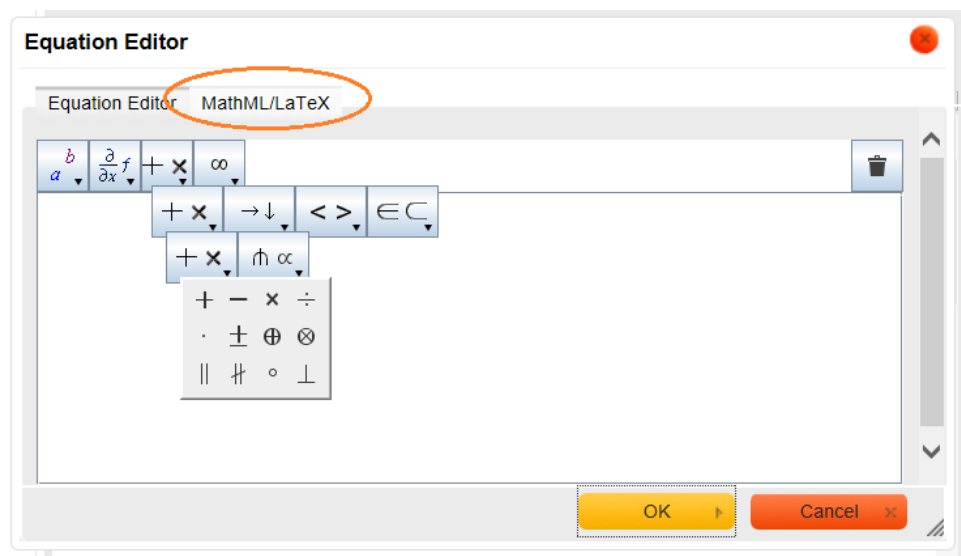


Figure 1.31: Vernieuwde Equation Editor in Möbius 10

TIP: Met het tabblad MathML/LaTeX in de *Equation Editor*, Figure 1.30 (page 25) 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 1.30* (page 25). 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>*</mo><mi>H</mi></mrow><mrow><mrow><mi>v</mi></m
i><mo>*</mo><mi>M</mi></mrow><mo>+</mo><mrow><mn>$C</mn><mo>*</mo><mi>H</mi></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 stippels in<br />
<em>m</em> = ...</p>
```

Figure 1.32: MathML-code aanpassen in de broncode van de vraag

1.2.3.10 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 formules 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.

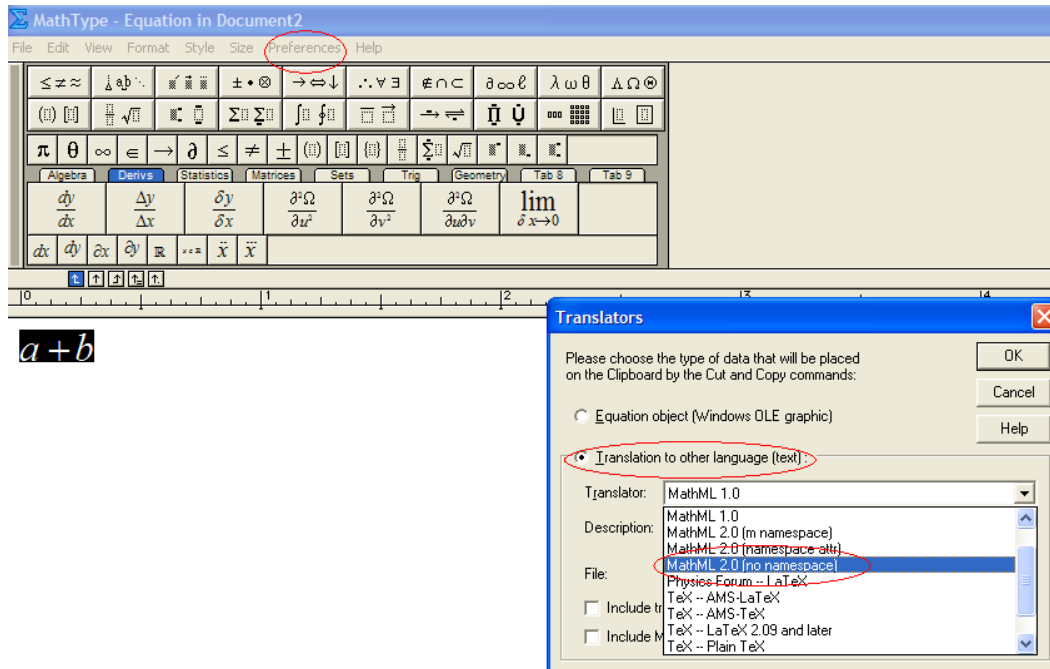


Figure 1.33: Met MathType de MathML-code maken

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

Zorg ervoor dat de volgende instellingen gedaan worden: zie *Figure 1.33* (page 27).

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 Möbius 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 Möbius 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 1.32* (page 26) 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.

1.2.3.11 LaTeX

LaTeX wordt ook ondersteund en er is 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 \ gevolgd door een haakje openen (en vervolgens aan het eind van de formule afsluiten met een backslash \ gevolgd door een haakje sluiten).

$$\backslash(.....\backslash)$$

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

$$\backslash[.....\backslash]$$

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 1.49 (page 35)* 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 bij de volgende website:

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

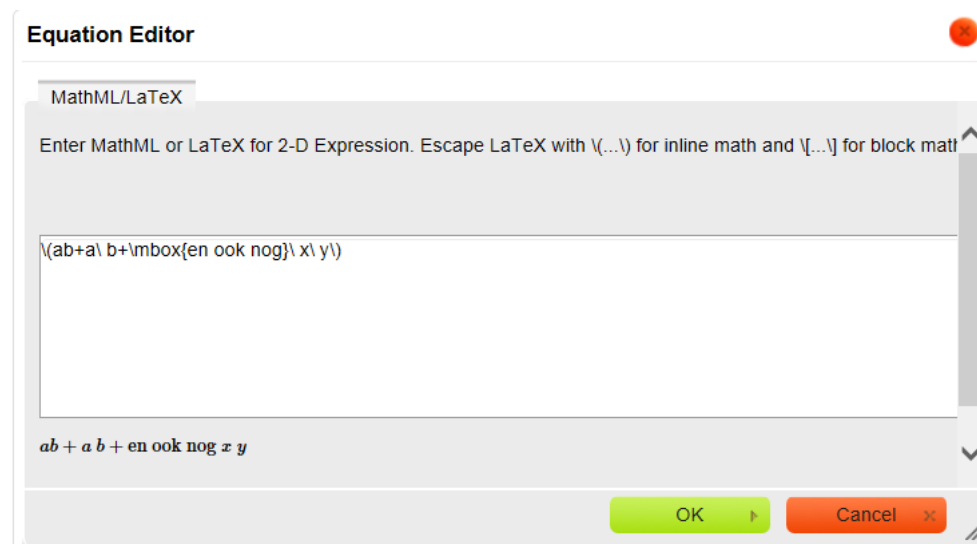


Figure 1.34: LaTeX met de Equation Editor

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="\(\ \frac{a}{b} \)";
(voor een klein breukje zijn geen accolades nodig).
$formulea="\(\ \frac{A_w}{L_{pp}}\cdot B_{mld} \)";
(voor een uitgebreide breuk staan teller en noemer apart tussen accolades.)
```

Met als resultaat:

$\begin{aligned} &\$test="\left(\frac{a}{b}\right)"; \\ &\$formulea="\left(\frac{A_w}{L_{pp} \cdot B_{mld}}\right)"; \end{aligned}$	
Variable	Value
test	$\frac{a}{b}$
formulea	$\frac{A_w}{L_{pp} \cdot B_{mld}}$

Figure 1.35: voorbeeld met LaTeX

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

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)
```

U kunt deze output in de tekst kopiëren als u deze tussen de haakjes plaatst: $\left(\dots \right)$.
Echter $\left$ en $\right$ zijn niet beslist noodzakelijk maar de haken worden dan zo nodig wel groter.
Het teken $\!$ haalt een spatie weg of maakt die iets kleiner.

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

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

Figure 1.36: LaTeX in het Algorithm voorbereiden

TIP: Een schat aan informatie vindt u op het web.

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

<https://en.wikibooks.org/wiki/LaTeX/Mathematics>

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

of

http://nl.wikibooks.org/wiki/LaTeX/Wiskundige_formules

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

1.2.3.11.1 Variabelen, spaties en tekst

Variabelen zijn altijd automatisch cursief.

Halve spaties ertussen krijgt u met `\,` (backslash gevolgd door een komma).

Een **hele spatie** krijgt u met een backslash gevolgd door een spatie.

Een **medium spatie** met `\:` (een backslash gevolgd door dubbele punt).

Een **grote spatie** met `\;` (een backslash gevolgd door puntkomma).

Een negatieve spatie met `\!` (een backslash gevolgd door een uitroepteken is om de ruimte die automatisch ontstaat weg te halen).

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

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 1.37: 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 enkele niet cursieve letters kunt u gebruiken (rm betekent roman) `\rm` niet cursief}

1.2.3.11.2 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 1.38: LaTeX met superscript en subscript

Let op dat u accolades gebruikt voor uitgebreidere sub- of superscript.

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

1.2.3.11.3 Vektoren en onderstrepingen

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

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

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

Figure 1.39: LaTeX met vektoren en onderstreping

Eventueel met stip of dubbele stip: $\dot{a} + \ddot{b}$

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

1.2.3.11.4 Vectoren en matrices

Een paar voorbeelden om kolomvectoren en zelfs matrices te maken met latex.

Twee dimensionale vectoren kan bijvoorbeeld met binom:

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

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

Met **vmatrix** krijgt u rechte strepen geschikt voor de determinant.

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

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

Het &-teken is ten behoeve van de uitlijning.

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

Figure 1.40: Kolomvectoren en matrices met LaTeX

Nog een voorbeeld:

```
"\(\begin{equation}
R^2 =
\begin{pmatrix} c & s \end{pmatrix}
\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}
\begin{pmatrix} c \\ s \end{pmatrix}
\end{equation}";
```

```
= c^2 + s^2
\end{equation}\)"
```

$$R^2 = \begin{pmatrix} c & s \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} c \\ s \end{pmatrix} = c^2 + s^2$$

Figure 1.41: Matrixvergelijking

Let op de uitlijningstekens met &.

Aangevulde matrix maak je met array:

```
"\(\left(\begin{array}{ccc|c}
a & b & c & d \\
e & f & g & h \\
i & j & k & l \\
m & n & o & p
\end{array}\right)\)"
```

Met als resultaat:

$$\left(\begin{array}{ccc|c} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{array} \right)$$

Figure 1.42: Aangevulde matrix

1.2.3.11.5 Stelsels vergelijkingen (haken en uitlijning)

Een voorbeeld van een setje vergelijkingen met een accolade ervoor en op =-tekens uitgelijnd.

```
"\(\left\{ \begin{array}{l}
1 + 2 = 3 \\
1 = 3 - 2
\end{array} \right.\)"
```

$$\left\{ \begin{array}{l} 1 + 2 = 3 \\ 1 = 3 - 2 \end{array} \right.$$

Figure 1.43: Stelsel vergelijkingen uitlijnen

De laatste accolade laat u weg door \right. (dus met een stip en niets er achter).

De &-tekens zijn voor de uitlijning.

De sterren achter align doet u als u geen nummering van de formules wilt.

1.2.3.11.6 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.

```
"\(\sin(x)+\sin(3x^2)\)"
"\(\log_{a}(x)+\ln(3\, y)+\Re(z)\, +\Re(z)+\mbox{Re}(z) +\mathrm{i}
+\log(x_z)\)"
```

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

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

Figure 1.44: LaTeX met functies en commando's

Het is wel mooi om een kleine spatie te maken tussen 3 en y bijvoorbeeld bij $\ln(3y)$.
Dat gaat dus met een backslash gevolgd door een komma.

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.

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

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

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

Figure 1.45: LaTeX met cursief en niet-cursief

1.2.3.11.7 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 1.46: LaTeX met breuken en wortels

1.2.3.11.8 Ongelijkheden en speciale tekens

TIP: Bij het tikken van een ongelijkheid in de tekst is het advies om niet de tekens van het toetsenbord te nemen voor $>$ en $<$, maar de officiële codes te nemen (let op de backslash).

$\geq$ \ge $\leq$ \le $<$ \lt $>$ \gt ∞ \infty $\cup$ \cup $\cap$ \cap

Meer tekens vindt u op het internet met de zoekopdracht latex symbols.

TIP: bij gebruik van de tekens $<$ en $>$ van het toetsenbord kan het zijn dat er errors ontstaan bij het migreren van content als de latex-code deze tekens bevat dus gebruik liever \lt en \gt.

1.2.3.11.9 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 0` 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 1.47: LaTeX met limieten

Ook de commando's `\sum` en `\prod` zijn bekende commando's en met subscript en superscript kunt u de grenzen meegeven.

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

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

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

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

Figure 1.48: 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.

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

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

Figure 1.49: LaTeX met gecentreerde formule

TIP: U kunt ook bereiken dat de grenzen voor integralen en sommatie echt boven en onder worden geplaatst door de extra optie met `\displaystyle{}`

```
"(\displaystyle{\sum^5_{i=2}\! 2^i} = 2^2 + 2^3 + 2^4 + 2^5 = 60\)"
```

1.2.3.11.10 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, maakt u daar weer math roman van met `\mathrm{d}`.

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

Maar hetzelfde effect geeft de volgende code:

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

Om de grenzen echt boven en onder het integraalteken te krijgen gaat met:

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

Of nog beter

```
\(\int \limits_a^b f(x) \, \mathrm{d}x\)
```

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

Figure 1.50: LaTeX met integralen

Een kringintegraal is ook aardig om te maken met `\oint_C f \, \mathrm{d}s`.

$$\oint_C f ds$$

Figure 1.51: LaTeX met kringintegraal

Dubbelintegraal maak je met:

```
\(\displaystyle{\iint\limits_G \mathrm{d}A}\)
```

en drievoudige integraal met

```
\(\displaystyle{\iiint\limits_G \mathrm{d}A}\)
```

Zie figuur *Figure 1.131 (page 118)*

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

$$\left(\frac{\mathrm{d}f(x)}{\mathrm{d}x}\right)$$

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

Figure 1.52: LaTeX en differentiëren

Oneindig symbool ∞ met `\infty`.

De stip (middot) met `\cdot`

Het officiële teken voor partiële differentiatie:

$$\frac{\partial z}{\partial x}$$

$\frac{dz}{dx}$	$\frac{\partial z}{\partial x}$
-----------------	---------------------------------

Figure 1.53: partiële differentiatie

TIP: Grotere integraalteken krijg je met:

$$\int \frac{1}{\sqrt{1-x^2}} \mathrm{d}x$$

1.2.3.11.11 Stuksgewijze functies

$$f(n) = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even} \\ -\frac{n+1}{2} & \text{if } n \text{ is odd} \end{cases}$$

Met `\quad` maakt u een spatie tussen formule en tekst.

Het `&`-teken is voor de uitlijning.

$$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ -(n+1)/2 & \text{if } n \text{ is odd} \end{cases}$$

Figure 1.54: stuksgewijze functie

1.2.3.11.12 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.

De accolades tikt u met het toetsenbord voorafgaand door een backslash. Dat is omdat accolades zonder backslash bedoeld zijn om dingen bij elkaar te houden.

De | streep tikt u met het toetsenbord.

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

In de volgende formule ziet u het effect van het schalen van de haken:

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

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

Figure 1.55: LaTeX en haakjes

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

Een pijltje maakt u met `\rightarrow`

TIP: Wilt u bijvoorbeeld alleen links een accolade dan doet u `\left\{ . . . \right.`, dus in plaats van de rechter accolade een stip.

1.2.3.11.13 Tekens en Griekse letters

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

`"\(\tan(\alpha)+\Delta+\delta\)"`

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

Figure 1.56: LaTeX en tekens en Griekse letters

1.3 Vraagtype Maple-graded

1.3.1 Inleiding

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 49).

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 Möbius 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 49).

TIP: voor het invoerveld: Als het antwoord erg lang is, voer dan bij de Custom CSS de volgende code in (te beginnen met een punt) zodat het invoerveld wat breder wordt.

```
.response input {width:250pt;}
```

Een invoerveld met de editor van de Maple-syntax is vaak onnodig hoog. Neem dan de volgende code in de Custom CSS.

```
.mwEquationEditor {height:100px !important}
```

1.3.2 Algemene structuur van Maple-graded vraagtype

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

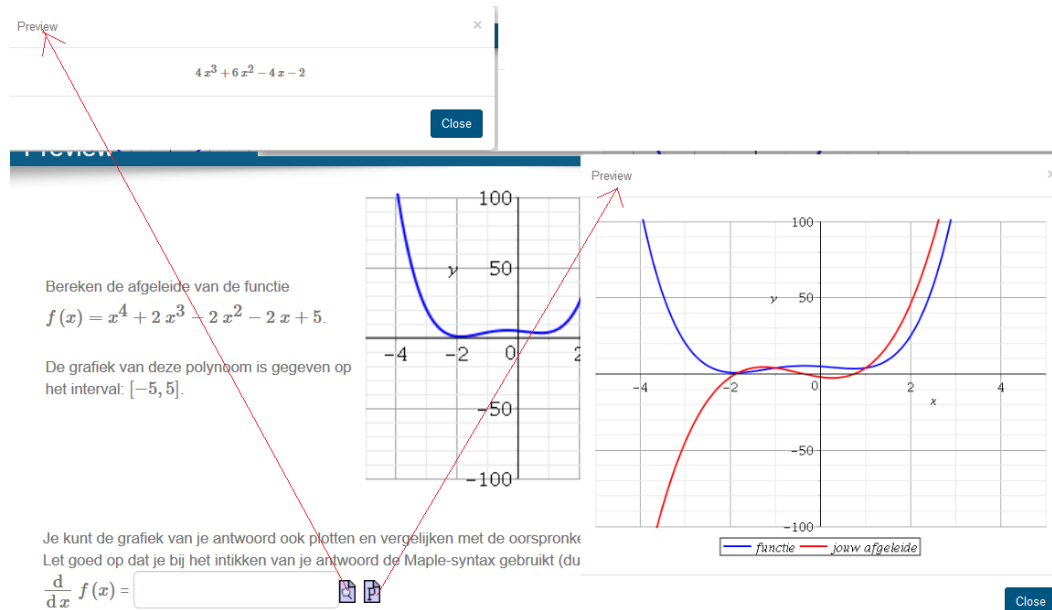


Figure 1.57: Voorbeeld van een Maple-graded vraag

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 verwezenlijkt worden wat betreft de presentatie van formule en grafiek in de tekst van de vraag. 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 1.57 (page 38) 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.

De student moet ook altijd op het vergrootglasje klikken om de formule middels *Preview* te controleren die hij heeft ingevoerd.

Het maken van een nieuwe vraag gaat met de blauwe knop *Create New - Question/Text*.

U ziet alle afzonderlijke rubrieken van deze vraag onder elkaar en deze zijn successievelijk open te klappen om te bewerken.

Begin met het geven van een naam aan de vraag. Deze naam ziet de student niet.

Dan klikt u *Question Text* open, u vult de vraag in en u brengt een invulveld aan met de knop van het vinkje in de checkbox zoals in de volgende figuur:

Figure 1.58: Maple-graded vraag aanmaken

Met het aanbrengen van een invulveld kunt u kiezen voor het vraagtype van uw keuze en in dit geval het vraagtype *Maple-graded*. U krijgt dan een bijpassend dialoogscherf (*Edit Response Area*).

Figure 1.59: Dialoogscherf van de Maple-graded vraag

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.

1.3.3 Algorithmen van de gehele vraag

Ga nu eerst naar het *Algorithm* om de variabelen aan te maken die nodig zijn voor de vraag. De variabelen die u hier aanmaakt zijn te gebruiken door de hele vraag heen, ik elk antwoordveld, in de Feedback en in de tekst van de vraag.... overal in de vraag waar nodig. Het *Algorithm* vormt het hart van de vraag.

Algorithm

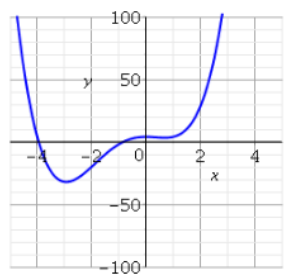
Edit the code for your algorithm in the text box to the right, or click "Show Designer" to use the algorithm designer. The algorithm designer tool allows you to define algorithms for your question by completing a form.

[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))");
$displaydfdxdx=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' ");
```

Variable	Value
p	$3x^3 - 4x^2 + 4$
poly	$x^4 + 3x^3 - 4x^2 + 4$
polydisplay	$f(x) = x^4 + 3x^3 - 4x^2 + 4$
antw	$4x^3 + 9x^2 - 8x$
displayantw	$4x^3 + 9x^2 - 8x$
displaydfdxdx	$\frac{d}{dx} f(x)$
displayinterval	$[-5, 5]$



figuur

Figure 1.60: 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))");
$displaydfdxdx=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 toe 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.

1.3.4 Question Text van het Maple-graded antwoordveld

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

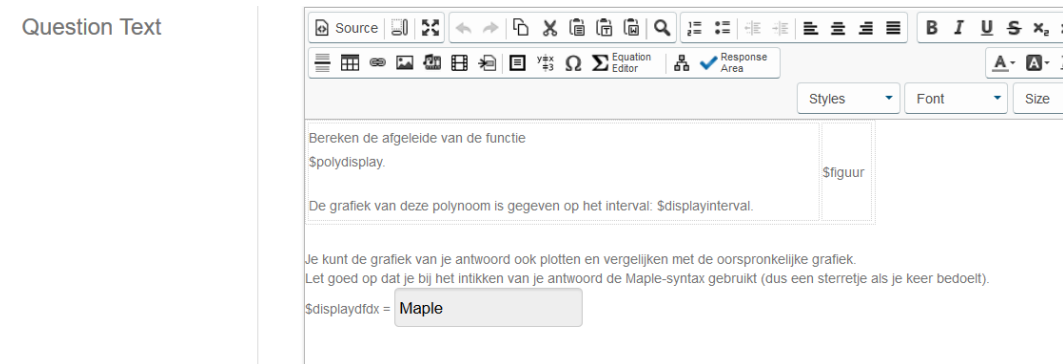


Figure 1.61: Tekst van de Maple Graded vraag

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

TIP: Door middel van een tabel (zonder borders) is de tekst naast de figuur geplaatst.

TIP: Dubbelklikken op het *Maple-graded* antwoordveld zal dit geopend worden en kunt u editen.

1.3.5 Answer van het Maple-graded antwoordveld

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.

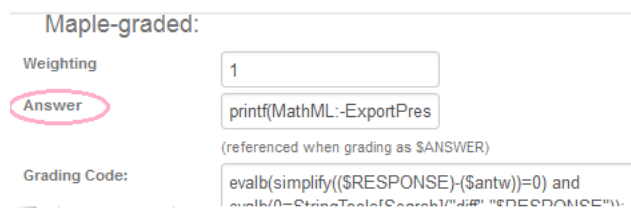


Figure 1.62: Het correcte antwoord in 2D

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));`

De student ziet de 2D-versie van het correcte antwoord op zijn scherm 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).

1.3.6 Grading Code van het Maple-graded antwoordveld

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 de *Grading Code* matcht u het antwoord van de student (\$RESPONSE) met het correcte antwoord dat u meestal in het *Algorithm* al hebt voorbereid (\$antw).

Voorbeeld van een Grading Code

The screenshot shows the 'Maple-graded' question configuration interface. It includes fields for 'Weighting' (set to 1), 'Answer' (set to '\$antw'), and 'Grading Code'. The 'Grading Code' field is highlighted with a red circle and contains the following Maple code:

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

Below the code field, there are dropdown menus for 'Expression Type' (set to 'Maple Syntax - e.g. diff(2*f(x),x)') and 'Text/Symbolic entry' (set to 'Text entry only').

Figure 1.63: Grading Code van de Maple-graded vraag

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 met `simplify`, moet dat 0 opleveren.

De opdracht `evalb` (evalueer boolean) geeft uitsluitsel (true of false).

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

De volgende toevoeging zorgt ervoor dat de student geen gebruik maakt van het beschikbare `maple-commando`.

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

TIP: Voor het geval dat u het correcte antwoord in mathml-code hebt gegeven in de rubriek *Answer* (Figure 1.62 (page 41)) 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: De 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 43)) 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 U ook nog moet programmeren dat het antwoord de karaktercombinatie "diff" niet bevat. Immers de student zou het antwoord met pen en papier zelf moeten kunnen berekenen. Dat de student dus niet "diff" in zijn antwoord mag gebruiken, kunt u geprogrammeren met de extra toevoeging

....and `evalb(StringTools[Search]("diff", "$RESPONSE")=0).`

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 formules 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*.

TIP: Houd er rekening mee dat bij de verschillende instellingen *Text Mode* en *Symbol Mode* er soms verschillende *Grading Codes* nodig zijn.

TIP: Zie voor meer mogelijkheden van *Grading Code* in paragraaf *Tips voor de Grading Code* (page 49).

1.3.7 Expression type van het Maple-graded antwoordveld

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*. U bepaalt dus als ontwerper van de vraag of er wel of niet een editor aangeboden wordt aan de student.

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

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

Text/Symbolic entry: **Text entry only**

Optional: **Text entry only**

Maple Repository: **Student can choose**

Figure 1.64: Instellingen voor Maple syntax bij Maple-graded vraagtype

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

1.3.8 Maple Repository van het Maple-graded antwoordveld

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*.

1.3.9 Plotting Code van het Maple-graded antwoordveld

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

Question Feedback

Maple-graded:

Weighting: 1

Answer: printf(MathML:-ExportPi)

(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

Optional:

Maple Repository:

Maple Repository

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

Custom Previewing Code: use InertForm:-NoSimpl in \$RESPONSE: end: prii

Figure 1.65: 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: In maple-opdrachten mogen geen dubbele quotes gebruikt worden. Hier ziet u de titel dus ook tussen backward quotes in plaats van dubbele quotes.

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.

1.3.10 Custom Previewing van het Maple-graded antwoordveld

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.

Edit Response Area

Question Feedback

Maple-graded:

Weighting: 1

Answer: printf(MathML:-ExportP
(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

Optional:

Maple Repository:

Maple Repository

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

Custom Previewing Code: use InertForm:-NoSimpl in \$RESPONSE: end: printf(InertForm:-ToMathML(%));

Figure 1.66: Custom Previewing Code

Als u niets invult, en de student klikt op *Preview* na het invullen van zijn formule, dan krijgt de student de ingetikte 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 in dit geval de *Preview* de niet-vereenvoudigde formule die de student heeft ingetikt weergeven in 2D.

In veel andere gevallen kunt u de *Preview* naar uw hand zetten om te bepalen wat de student te zien krijgt bij het drukken op de *Preview*-knop.

Zie in de volgende paragraaf (page 46) voor een aantal voorbeelden.

TIP: De Preview-knop is er alleen als u de instellingen voor text-invoer hanteert.

1.3.10.1 Andere mogelijkheden voor de custom preview:

- `printf(MathML[ExportPresentation](Vector($RESPONSE)))`;
In het geval dat de student een lijst met getallen moet invoeren bijvoorbeeld [3,4,5]. De student ziet dan in de Preview de echte vector met kentallen onder elkaar. Zie verder voor het overhoren van vectoren in paragraaf *Matrices en Vectoren* (page 125).
- `if evalb(StringTools[Search]("abs",$RESPONSE)=0) then
 printf("Vergeet niet de modulusstrepen bij de logaritme. Dat
 doe je met abs(..).") else printf(MathML[ExportPresentation]
 ($RESPONSE)) end if`;
Hier controleert u of in het antwoord van de de student wellicht geen abs voorkomt. In dat geval krijgt hij een melding dat hij abs niet moet vergeten.
- `use InertForm:-NoSimpl in $RESPONSE: end: printf(InertForm:-
 ToMathML(%))`;
Hier wordt in de Preview de niet-vereenvoudigde vorm gegeven die de student invoert. Hiermee kunt u dus de automatische vereenvoudiging tegenhouden.
- `if evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","[" >= 2) or
 evalb(StringTools[CountCharacterOccurrences]("$RESPONSE",",") > 1) then
 printf("tik een tweedimensionale vector en gebruik decimale punt") else
 printf(MathML[ExportPresentation](Vector($RESPONSE))) end if`;
Hier controleert u of de student slechts één vector invoert met twee kentallen. Er mag dus maar één vierkant haakje openen en één komma in het antwoord voorkomen.
- `if evalb(StringTools[Search]("*",$RESPONSE)=0) then printf("Vergeet niet een
 ster te tikken als je keer bedoelt.") else printf(MathML[ExportPresentation]
 ($RESPONSE)) end if`;
- `if evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","*") <= 2)
 then printf("Vergeet niet een ster te tikken als je keer bedoelt.") else
 printf(MathML[ExportPresentation]($RESPONSE)) end if`;
- `if evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","=")=0)
 then printf("Voer een vergelijking in x en y in.") else
 printf(MathML[ExportPresentation]($RESPONSE)) end if`;

1.3.11 Feedback van de gehele vraag

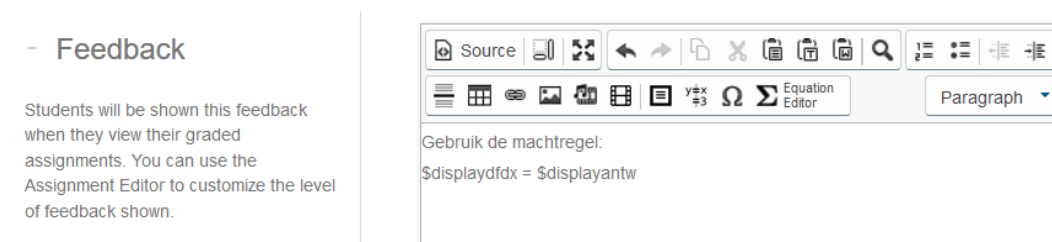


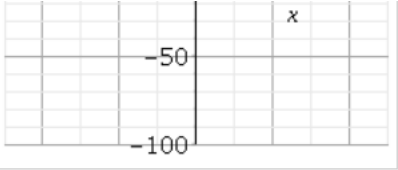
Figure 1.67: Feedback bij de Maple-graded vraag

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*.

De grafiek van deze polynoom is gegeven op het interval: $[-5, 5]$.



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).

$\frac{d}{dx} f(x) =$

Your response	Correct response
$4x^3 + 9x^2 + 8x - 5$	$4x^3 + 9x^2 + 8x - 5$

✓ Grade: 1/1.0

✓ Total grade: $1.0 \times 1/1 = 100\%$
 Feedback:
 Gebruik de machtregeel:
 $\frac{d}{dx} f(x) = 4x^3 + 9x^2 + 8x - 5$

Figure 1.68: 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 twee-dimensionale vorm weergegeven precies zoals de student het ook in de Editor had ingevoerd.

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

1.3.12 Feedback van het antwoordveld voor How did I do

Als u *Assignments* geeft of in Möbius werkt met *Lessons*, is er voor de studenten vaak de mogelijkheid om te klikken op *How did I do*.

Voor die situatie kunt u per antwoordveld ook *Feedback* voorbereiden.

Open de vraag en dubbelklik op het antwoordveld. U komt dan in *Edit Response Area* en u ziet bovenaan een tabblad *Feedback*.

Dit tabblad is speciaal voor *Feedback* van het antwoordveld in de situatie van *How did I do*. Deze feedback is niet te zien na afloop van de grading en ook niet in het *Gradebook*.

Edit Response Area

Question Feedback

Maple-graded:

Weighting 1

Answer `printf(MathML:-ExportPi`
(referenced when grading as \$ANSWER)

Grading Code: `evalb(simplify(($RESPONSE)-($antw))=0) and`

Figure 1.69: Tabblad voor Feedback bij het invulveld

Daarin kunt u uw Feedback opmaken.

Edit Response Area

Question Feedback

Source Undo Redo Copy Paste Link Unlink Search Bulleted List Numbered List Decrease Indent Increase Indent Bold Italic Underline Strikethrough Subscript Superscript

Paragraph Font Size

Gebruik de machtsregel voor differentiëren.

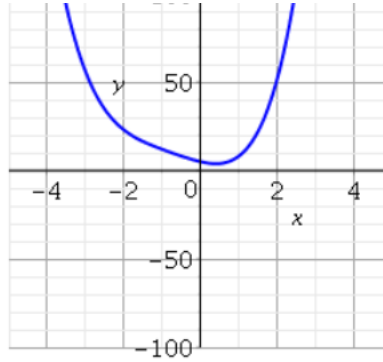
Figure 1.70: Feedback invoeren voor bij het invulveld

Als u de vraag *Previewt*, klikt u op *How did I do* om het effect te zien.

Bereken de afgeleide van de functie

$$f(x) = x^4 + 3x^3 + 4x^2 - 5x + 5.$$

De grafiek van deze polynoom is gegeven op het interval: $[-5, 5]$.



Je kunt de grafiek van je antwoord ook plotten en vergelijken met de grafiek van de correcte oplossing. Het is goed op dat je bij het intikken van je antwoord de correcte oplossing ziet.

$$\frac{d}{dx} f(x) = 4x^3 + 9x^2 + 8x - 5$$

Correct

Your Answer: $4x^3 + 9x^2 + 8x - 5$

Correct Answer: $4x^3 + 9x^2 + 8x - 5$

Feedback: Gebruik de machtsregel voor differentiëren.

Figure 1.71: How did I do en feedback

1.4 Tips voor de Grading Code

1.4.1 Inleiding

De *Maple-graded* vraag zoals besproken in paragraaf *Vraagtype Maple-graded* (page 37) 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 is iets dergelijks standaard voorgeprogrammeerd en staat er in de *Grading Code* reeds ingevuld:

```
evalb( ($ANSWER) - ($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:

```
is( ($ANSWER) - ($RESPONSE)=0 );
or
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 51) en ook in paragraaf *Ongeordende lijsten en geordende lijsten* (page 99) voor een voorbeeld daarvan.

TIP: 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*.

Als het antwoord reeds in het *Algorithm* voorgeprogrammeerd is met \$antw of iets dergelijks, dan kunt u dat in die rubriek invullen.

TIP: 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: u doet iets met het antwoord van de student, dat standaard \$RESPONSE is, en vervolgens maakt u een match met het correcte antwoord waarmee u hetzelfde doet.

Het correcte antwoord heet in de volgende paragrafen steeds \$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 formule, 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 de volgende paragrafen vindt u voorbeelden van de *Grading Code* in allerlei situaties.

TIP: Houd er rekening mee dat bij de verschillende instellingen *Text Mode* en *Symbol Mode* er soms verschillende *Grading Codes* nodig zijn.

1.4.2 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 (\$antw1 en \$antw2). 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(($antw1)-($RESPONSE))<0.1)
or evalb(abs(($antw2)-($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(($antw1)-($RESPONSE))<0.1) then 1 elif evalb(abs(($antw2)-($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.

1.4.3 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 `evalb(($RESPONSE)=$antw)`; dus iets krachtiger dan `evalb(($RESPONSE)-($antw)=0)`; en dit is weer krachtiger dan `evalb(simplify(($RESPONSE)-($antw))=0)`;

Deze laatste *Grading Code* is het meest robuust en 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*.

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 75) 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. Immers bij de instelling *Maple syntax Text entry only* wordt er vaak een vereenvoudiging aangeboden bij het klikken op *Preview*.

Wat u dan moet doen staat in de paragraaf over het tegenhouden van automatische vereenvoudiging (page 18).

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

Breuken

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.

Kijk ook eens naar het type uitdrukking. Als de uitdrukking als één breuk geschreven is, dan is het type uitdrukking een vermenigvuldiging.

```
> whattype(breuk1);
```

$$`+` \quad (1.8)$$

```
> whattype(breuk2);
```

$$`*` \quad (1.9)$$

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

$$breuk3 := \frac{a}{x} + 3a \quad (1.10)$$

```
> breuk4:=simplify(breuk3); breuk4a:=normal(breuk3);
```

$$breuk4 := \frac{3ax + a}{x}$$

$$breuk4a := \frac{a(3x + 1)}{x} \quad (1.11)$$

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

$$breuk5 := \frac{3ax + a}{x} \quad (1.12)$$

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

$$true \quad (1.13)$$

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

false (1.14)

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.

Zie in de paragraaf over *Breuken* (page 75) voor nog meer mogelijkheden.

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

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

false (1.15)

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

false (1.16)

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

$a(x-5) - ax + 5a$ (1.17)

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

0 (1.18)

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.

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 (1.19)

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

false (1.20)

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

true (1.21)

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

true (1.22)

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`.

Zie in de paragraaf over *Verzamelingen* (page 67) voor nog meer mogelijkheden.

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");
```

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

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

$$\text{true} \quad (1.24)$$

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

$$\text{"string1+s^2"} \quad (1.25)$$

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

Zie in de paragraaf over *StringTools* (page 55) voor nog meer mogelijkheden.

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);
```

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

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

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

```
> 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);
```

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

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

$$\text{true} \quad (1.27)$$

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.

Zie in de paragraaf over *Vergelijkingen* (page 90) voor nog meer mogelijkheden.

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

1.4.4 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 dubbele quotes omheen te zetten. Dit gaat het mooist als u de instellingen voor het invoeren van de formule hebt gezet op *Maple Syntax en Text entry only*.

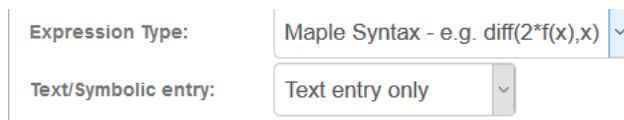


Figure 1.72: Instellingen met Maple Syntax

In paragraaf *De Analyse van de Response* (page 64) 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 óf het karakter niet voorkomt óf 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 *openvolging van karakters* wellicht niet voorkomt in de string.

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

Dit betekent dus dat de openvolging 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

(1.28)

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

Met het programmeren van de volgende regel in de *Grading Code* kunt u er dus voor zorgen dat bijvoorbeeld de combinatie x^2*y^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 ziet u bij verschillende instellingen (*Text entry only* of *Symbol entry only*) dat er gevraagd wordt naar een vereenvoudiging van een formule.

2

(1.29)

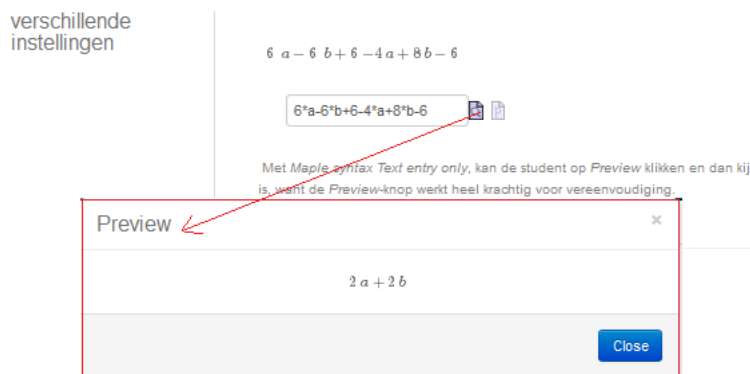


Figure 1.73: Vereenvoudigen

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

TIP: De *Preview*-functie bij *Maple syntax* is zó krachtig dat de vereenvoudiging wordt voorgezegd zoals in *Figure 1.73 (page 56)* 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 62)*.

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 voor de Grading Code controleren op het aantal specifieke karakters dat er in het antwoord van de student voorkomt. Zie ook paragraaf *De Analyse van de Response (page 64)*.

In de programmering van de *Grading Code* van de vraag zoals te zien is in *Figure 1.73 (page 56)* 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 overtekken 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 dus níet 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 nooit de instelling *Formula* tenzij het niet anders kan. Zie een voorbeeld in paragraaf *Breuken (page 75)* in *Figure 1.92 (page 81)*.

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("6/(5*x^2*y)");`

Een andere manier om in de feedback één breuk te krijgen is om het antwoord naar `mathml` weg te schrijven met `printf(MathML[ExportPresentation]($antw))` in te vullen in de rubriek *Answer*.

TIP: Hetzelfde krijgt u bij de wortel in de feedback.

Maak een variabele aan in het *Algorithm* bijvoorbeeld `$antw="sqrt(3)";` (Dus niet met Maple want die maakt er $3^{1/2}$ van).

U kunt deze variabele gewoon gebruiken in de *Grading Code*. Bij het correcte antwoord zet u dan niet `$antw` neer maar `printf("$antw");` (Dit commando haalt de quotes weg en schrijft precies over wat tussen quotes staat, meer niet.)

Een andere manier om in de feedback niet $3^{1/2}$ te krijgen is om het antwoord naar `mathml` weg te schrijven met `printf(MathML[ExportPresentation]($antw))` in te vullen in de rubriek *Answer*.

1.4.5 Exponenten Logaritmen en Pi

De exponentiële functie

Het is belangrijk twee situaties te onderscheiden:

- **Binnen maple** is de exponentiële functie e^x gedefinieerd met `exp(x)` en de letter `e` wordt niet gezien als het getal van Euler.
- **Buiten maple** is `exp(...)` hetzelfde als $e^{(...)}$ en kan dus beide gebruikt worden.
U ziet dat in het *Algorithm* dat beide variabelen als resultaat 2.72 geven waar Maple niet gebruikt wordt voor de berekening.
`$test3=decimal(2,e);`
`$test4=decimal(2,exp(1));`

TIP: In het **numerieke vraagveld** wordt `exp(...)` of $e^{(...)}$ niet geaccepteerd ook niet als *Accept arithmetic* aangevinkt is.

Binnen het Maple-graded vraagtype moet het correcte antwoord met $\exp(\cdot)$ worden gedefinieerd en de student moet ook $\exp(\cdot)$ invoeren als hij $e^{(\cdot)}$ bedoelt. Ook als de editor aangeboden wordt.

HINT: Bij de instelling *Formula* in het *Maple-graded* antwoordveld (is niet aan te bevelen), kan de student zowel $\exp(\cdot)$ als $e^{(\cdot)}$ invoeren voor een correcte beoordeling. Ook het antwoordveld van het type *Mathematical Formula* (is niet aan te bevelen), kan de student zowel $\exp(\cdot)$ als $e^{(\cdot)}$ invoeren voor een correcte beoordeling.

Als de exponent uit meer onderdelen bestaat, dan staan er haakjes om de exponent

In beide situaties voor *text entry only* en ook voor *symbol entry only* kunt u in de *Grading Code* van het Maple-graded vraagtype programmeren dat $\exp(\cdot)$ als ook $e^{(\cdot)}$ beide goedgerekend worden. Bij de instelling *maple-syntax Symbol entry only* is er namelijk geen knopje waarmee de student e invoegt. Dus de student tikt de e met het toetsenbord.

Voor beide situaties neemt u de volgende Grading Code:

```
resp:= "$RESPONSE":
fixed_resp:= StringTools[SubstituteAll](resp, "e^", "exp"):
evalb(( $ans) - (parse(fixed_resp))=0);
```

Het betekent dat u eerst in de string van het antwoord overal $e^$ vervangt door $\exp$.

Daarmee krijgt u een nieuwe string waarvan u met behulp van het maple commando `parse` weer een echte formule maakt om mee te rekenen.

TIP: Dat hier met `StringTools` wordt gewerkt voor de substitutie is gedaan omdat gewone substitutie niet werkt met $e^$: een karakter gevolgd door een operator.

Als de exponent uit één onderdeel bestaat

Bij de situatie dat e^x gezien moet worden als $\exp(x)$ en dus als beide goedgerekend moeten worden, kunt u niet de bovenstaande code gebruiken want daarmee zou e^x overgaan in $\exp x$ zonder haakjes.

Bij het ontwerpen van de *Grading Code* voor voor *text entry* en ook voor *symbol entry* moet u op voorhand kunnen weten welke exponentiële functie de student invoert en in moet voeren. Stel dat $\exp(x)$ en e^x beide als correct beoordeeld moeten worden kunt u nemen:

```
evalb(subs(e^x=exp(x), $RESPONSE) - exp(x)=0);
```

TIP: Hier is de string niet nodig. U kunt gewoon met substitutie werken en in het antwoord van de student e^x vervangen door $\exp(x)$.

TIP: Bij de instelling *Formula* van het Maple-graded vraagtype wordt de $e^$ automatisch omgezet naar $\exp$ en kunt u volstaan met de volgende *Grading Code*

```
evalb(( $ans) - ($RESPONSE) =0);
```

Het nadeel is wel dat bij de instelling *Formula* de student geen sterren hoeft te tikken. (Maar juist wel weer vóór de e bij $5 * e^{(5x)}$. Niet aan te bevelen dus.)

TIP: Let hier op de truc met de spatie naast de quotes bij de maple opdracht voor de MathML conversie $\$exprdisplay1$.

Zonder spatie wordt de opdracht overgenomen en met spatie wordt de opdracht eerst uitgevoerd waarbij er overgegaan wordt naar de natuurlijke logaritme.

TIP: Voor het numerieke antwoordveld werkt u buiten maple en is $\log(5)$ dus de Briggse logaritme terwijl $\ln(5)$ de natuurlijke logaritme is.

Het getal π

Ook hier is het weer belangrijk twee situaties te onderscheiden:

- **Binnen maple** geldt dat alleen het getal aangeduid met Pi de constante 3.14... is. Het getal pi is gewoon een variabele.
- **Buiten maple** geldt dat zowel Pi als pi opgevat wordt als de constante 3.14....
Dat wil zeggen dat in de vraagtypen *Numeric* en *Mathematical Formula* het niet uitmaakt of u Pi of pi hanteert.

Zodra u dus in het vraagtype *Maple-graded* zit, is er wel onderscheid, want dan zit u binnen maple met alles zoals *Grading Code* en *Answer* (correcte antwoord e.d.).

Bestudeer eens het volgende *Algorithm*:

```
$getal1="Pi";
$getal2="pi";
$getal1dec=decimal(2,$getal1);
$getal2dec=decimal(2,$getal2);
$getal3=maple("Pi");
$getal3dec=decimal(2,$getal3);
$getal4=maple("pi");
$getal4dec=decimal(2,$getal4);
$getal4mapledec=maple("evalf($getal4)");
```

Variable	Value
getal1	Pi
getal2	pi
getal1dec	3.14
getal2dec	3.14
getal3	Pi
getal3dec	3.14
getal4	pi
getal4dec	3.14
getal4mapledec	pi

Figure 1.76: Verschil tussen Pi en pi

TIP: Het is handig als u in de *Grading Code* van het *Maple-graded* vraagtype maatregelen treft om zowel Pi als pi goed te keuren.

U substitueert dan alle pi door Pi in het antwoord van de student en daarna matcht u met het juiste antwoord.

```
evalb($antw-subs(pi=Pi,$RESPONSE)=0);
```

of voor de zekerheid:

```
evalb(subs(pi=Pi,$antw-($RESPONSE))=0);
```

TIP: Bij de instelling met *Symbol entry only* kan dezelfde *Grading Code* gehanteerd worden, hoewel de knop voor π altijd vertaald wordt met Pi.

1.4.6 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.

Figure 1.77: Tegenhouden van automatische vereenvoudiging in de Preview door gebruikmaking van de Custom Previewing Code

Zie ook paragraaf *Custom Previewing Code* (page 45) .

Zie ook paragraaf *Automatische vereenvoudiging tegenhouden* (page 18) bij het voorbereiden van formules in het *Algorithm*.

1.4.7 Complex numbers

De imaginaire eenheid voor berekeningen in Maple is de hoofdletter I. Voor uw tekst kunt u een keuze maken uit de letters i of j die de student in de formule zal zien.

Laat de berekeningen met complexe getallen over aan Maple en gebruik dan de I. Met het resultaat van de berekening kunt u met substitutie overgaan naar de letters i of j voor de imaginaire eenheid.

De substitutie kunt u doen met het commando `subs` of met de opdracht `local I:=i`:

Ga uit van het volgende *Algorithm*:

```
$a1=switch(rint(2), range(-5, -1), range(1, 5));
$b1=switch(rint(2), range(-5, -2), range(2, 5));
$a2=switch(rint(2), range(-5, -2), range(2, 5));
$b2=switch(rint(2), range(-5, -2), range(2, 5));
```

```

$index=0;
$ij=switch($index,"i","j");
$vraag="($a1+($b1)*I)*($a2+($b2)*I)";
$vraagij="($a1+($b1)*$ij)*(($a2)+($b2)*$ij)";
$test=maple("printf(MathML[ExportPresentation]($vraagij))");
$vraagdisplay=maple("use InertForm:-NoSimpl in $vraagij: end:
  printf(InertForm:-ToMathML(%))");
$antw=maple("evalc($vraag)");
$antwoord=maple("local I:=$ij: $antw");
$antwoorddisplay=maple("printf(MathML[ExportPresentation]
($antwoord))");

```

Variable	Value
a1	-3
b1	4
a2	2
b2	5
index	0
ij	i
vraag	$(-3+(4)*I)*(2+(5)*I)$
vraagij	$(-3+(4)*i)*((2)+(5)*i)$
test	$(-3 + 4 i) (2 + 5 i)$
vraagdisplay	$(-3 + 4 i) (2 + 5 i)$
antw	$-26-7*I$
antwoord	$-26-7*i$
antwoorddisplay	$-26 - 7 i$

Figure 1.78: Complex numbers

In het *Algorithm* ziet u dat voor de imaginaire eenheid gekozen is voor de letter i. Met de variabele \$vraag wordt gerekend in maple. De variabele \$vraagij is voorbereid voor de tekst van de vraag en dus geconverteerd naar MathML (\$vraagdisplay).

De variabele \$antw is het berekende antwoord. In de *Grading Code* gebruikt u het correcte antwoord \$antwoord (met de letter i als de imaginaire eenheid).

Het correcte antwoord geconverteerd naar MathML (\$antwoorddisplay) kunt u gebruiken in de Feedback van de vraag.

HINT: In plaats van de twee regels in het *Algorithm*:

```

$antw=maple("evalc($vraag)");
$antwoord=maple("local I:=$ij: $antw");

```

kunt u ook volstaan met één regel:

```

$antwoord =maple("subs(I=$ij, evalc($vraag))");

```

De volgende *Grading Code* voor het *Maple-graded* vraagveld is:

```
evalb(($antwoord)-($RESPONSE)=0) and  
evalb(StringTools[CountCharacterOccurrences]  
("$RESPONSE", "$ij")<2);
```

In deze *Grading Code* gebruikt u \$ij welke hier staat voor de letter i waar de student deze mzximaal maar één keer mag gebruiken in zijn antwoord.

In *Answer* geeft u het correcte antwoord \$antwoord ten behoeve van de feedback.

In de *Custom Previewing Code* kunt u het volgende programmeren:

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

Enkele maple commando's voor complexe getallen

`evalc( . . )` geeft het complexe getal in de vorm $a+I*b$. Met dit commando `evalc` (evaluate complex) laat u maple weten dat de onbekende parameters reële getallen voorstellen.

`abs( . . )` geeft de absolute waarde van het complexe getal.

`argument( . . )` geeft het argument van het complexe getal.

`convert( . . , polar )` geeft de poolcoördinaten van het complexe getal.

`conjugate( . . )` geeft de complex geconjugeerde van het complexe getal.

1.4.8 Uitdrukkingen met de letter I

Als de letter I in een uitdrukking voorkomt die niet de imaginaire eenheid is, moet bij *Maple-graded* vraagtype iedere keer ervoor gezorgd worden dat Maple de letter I niet ziet als imaginaire eenheid. Dat kan met een extra commando:

`local I:=`I ``: Dus backquotes om de letter I inclusief de spatie.

- Het antwoord is bijvoorbeeld in het *Algoritme* voorbereid:
`$antw="3*P*L/(E*I)";`
`$antwdisplay=maple("local I:=`I``
``:printf(MathML[ExportPresentation]($antw))");`

In het *Maple-graded Response Area* invullen:

- *Answer*:
`local I:=`I `:printf(MathML[ExportPresentation]($antw));`
- *Grading Code*:
`local I:=`I `:evalb(($antw)-($RESPONSE)=0);`
- *Expression Type*: op *Maple Syntax* zetten.
- *Custom Previewing Code*:
`local I:=`I `:printf(MathML[ExportPresentation]($RESPONSE));`

1.4.9 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 55) 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* antwoordveld 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 mogelijkerwijs 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 1.79: Instellingen voor het antwoord

De drie mogelijke instellingen zijn:

- 1) *Maple syntax Text entry only* (**aanbevolen**)
- 2) *Maple syntax Symbol entry only*
- 3) *Formula* (**liever niet gebruiken**)

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 62). 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 55).

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 1.80* (page 65)).

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

Figure 1.80: 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.

Your response	string response
$(c-5)/c+4$	"(c-5)/c+4"
$3/(a*b)$	"3/(a*b)"

Maple syntax Text entry only

Figure 1.81: 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 formule, door de student ingetikt in de Editor, omgezet wordt naar een string, zoals in de volgende figuur te zien is.

Your response	string response
$\frac{(c-5)}{c+4}$	"(c-5)/(c+4)"
$\frac{3}{a \cdot b}$	"3/a/b"
$\frac{3}{a \cdot b}$	"3/a/b"
$\frac{3}{(a \cdot b)}$	"3/a/b"
$\frac{3}{ab}$	"3/ab"

Maple syntax Symbol entry only

Figure 1.82: 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:

Your response	string response
$\frac{3 \cdot 5 + a}{2(x \cdot y + y^2)}$	"(3*5+a)/(2*x*y+y^2)"
$\frac{3 \cdot 5 + a}{x(xy + y^2)}$	"(3*5+a)/x(xy+y^2)"
$a \cdot b$	"a*b"

Maple syntax Symbol entry only

Figure 1.83: 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.

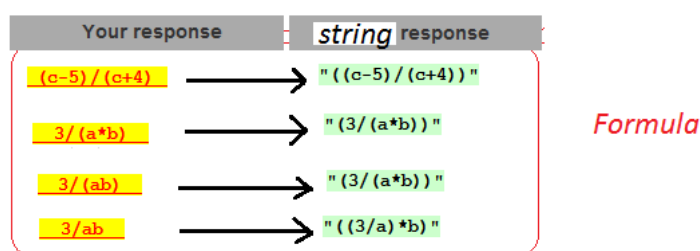


Figure 1.84: Het zichtbaar maken van de "\$RESPONSE met Formula

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 1.80* (page 65) 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 82) in *Figure 1.93* (page 82) en *Figure 1.94* (page 82).

Voorbeelden van de analyse van de respons van Matrices en Vectoren zijn te vinden in paragraaf *Matrices en Vectoren* (page 125).

Voorbeelden van de analyse van de respons van differentialen en integralen zijn te vinden in paragraaf *Integralen* (page 113).

1.4.10 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 (stelsels) vergelijkingen of iets dergelijks. Zie paragraaf (page 94) 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 138) waar het ook mogelijk is om ongeordende lijsten te overhoren.

TIP: Bij verzamelingen worden dubbele elementen als enkele gezien.

TIP: 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};`

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

$$B := \left\{ 3, \frac{1}{3}, 5.23 \right\} \quad (1.30)$$

> `evalb(A=B);`

$$false \quad (1.31)$$

> `convert(A, rational);`

$$\left\{ 3, \frac{1}{3}, \frac{523}{100} \right\} \quad (1.32)$$

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

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

$$true \quad (1.33)$$

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

$$\{33, 300, 523\} \quad (1.34)$$

Eventueel met de **elementwise operator** (de tilde `~`):

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

$$\{33, 300, 523\} \quad (1.35)$$

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

Of met de elementwise operator:

$$true \quad (1.36)$$

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

$$true \quad (1.37)$$

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 vind u in het voorbeeld van (page 103) 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 `{Sopl}` afzonderlijk te behandelen. Hetzelfde doet u dan met de verzameling `{RESPONSE}`.

```
evalb(map(x->round(100*x), {Sopl}) = map(x->round(100*x),
{RESPONSE}));
```

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);
```

$$\{2, 3, 43\} \quad (1.38)$$

```
> numtheory[divisors](258);
```

$$\{1, 2, 3, 6, 43, 86, 129, 258\} \quad (1.39)$$

U kunt ook het antwoord van de student controleren of de verzameling bepaalde elementen bevat.

U kunt aan elk van de drie elementen 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);
```

1.4.11 Uitdrukkingen met 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));
```

$$f := 3.x - 4.500000000x^3 + 2.025000000x^5 \quad (1.40)$$

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

$$-4.500000000 \quad (1.41)$$

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

$$3., -4.500000000, 2.025000000 \quad (1.42)$$

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

$$\{-450, 203, 300\} \quad (1.43)$$

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

$$\{-450, 203, 300\} \quad (1.44)$$

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

$$f1 := 5.936149480x + 17.37194834 \quad (1.45)$$

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

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

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

$$17.37194834, 5.936149480 \quad (1.46)$$

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. (De tilde is de **elementwise operator**.)

Let op dat u eventueel op numerieke getallen overgaat als de coëfficiënten niet gemakkelijk te bepalen zijn met het commando `coeffs`.

Een alternatief is eerst de polynoom schrijven zonder haakjes (met `expand`) anders lukt 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.

Voorbeeld met reeksontwikkeling

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 $x = 0$

$-8*x^2+1+(32/3)*x^4$



Preview

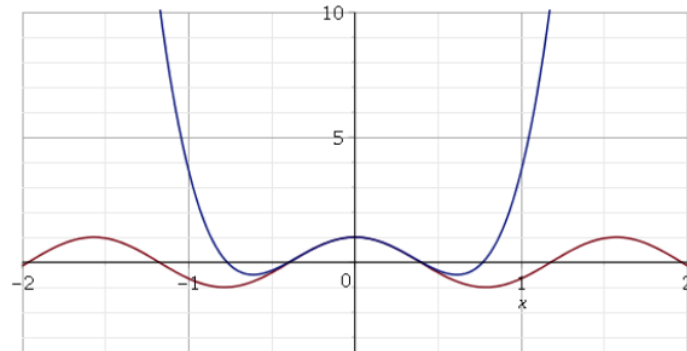


Figure 1.85: Een polynoom met decimalen matchen

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 1.59 (page 39) waar u die *Plotting Code* moet invullen.

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

```
$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) )");
$plot=plotmaple("plot([$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 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`.

In de *Grading Code* programmeert u 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 *verzamelings*

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 67) voor meer informatie over verzamelingen.

Operanden en coëfficiënten van een uitdrukking

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));
```

$$f := 3.x - 4.500000000x^3 + 2.025000000x^5 \quad (1.47)$$

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

$$2.025000000x^5, -4.500000000x^3, 3.x \quad (1.48)$$

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

$$2.025000000x^5 \quad (1.49)$$

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

$$-4.500000000 \quad (1.50)$$

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

$$3 \quad (1.51)$$

```
> coeffs(-101.5*0.8^n+100);
```

$$100, -101.5 \quad (1.52)$$

```
> nops(-101.5*0.8^n+100);
```

$$2 \quad (1.53)$$

```
> coeff(-101.5*0.8^n+100,0.8^n);
```

$$-101.5 \quad (1.54)$$

```
> map(x->round(100*x),{coeffs(-101.5*0.8^n+100)});
```

Of met de elementwise operator (`~`):

$$\{-10150, 10000\} \quad (1.55)$$

```
> round~(100*~{coeffs(-101.5*0.8^n+100)});
```

$$\{-10150, 10000\} \quad (1.56)$$

Een voorbeeld van het gebruik van operanden in uitdrukkingen is ook te vinden in figuur *Figure 1.86 (page 73)*.

Nog meer over operanden staat vermeld in paragraaf *Wortelvormen (page 86)*.

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);
```

Nog een grading code voor een **lineaire functie**:

```
evalb(abs(coeff($RESPONSE, x)-coeff($p, x))<0.1) and
evalb(abs(($RESPONSE-coeff($RESPONSE, x)*x)-($p-coeff($p, x)*x))<0.1);
```

Werken met decimale getallen

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.



Preview

Given the following difference equation

$$Y_{n+1} - 0.8 Y_n = 14$$

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.

$A \cdot 8^n + 70$

The general, $Y_n =$   (use the symbol A for the constant)

2. Determine the particular solution given $Y_1 = 12$.

$-72.5 \cdot 8^n + 70$

The particular solution =  

Figure 1.86: Uitdrukking met decimale coëfficiënten

Het algorithm is het volgende:

```
$a=decimal(1, range(0.5, 0.9, 0.1));
$b=range(10, 20, 2);
$c=range(2, 20);
$d=range(12, 20);
$displayeq1=maple("MathML[ExportPresentation](Y[n+1]-$a*Y[n]=
$b)");
$r=$a;
$YPi=maple("$b/(1-$a)");
$YPI=decimal(2, $YPi);
$Ygeneral=maple("A*($r)^n+$YPI");
$Aa=maple("($d-$YPI)/$a");
$A=decimal(2, $Aa);
$Ypart=maple("$A*($r)^n+$YPI");
$displaygeneral=maple("MathML[ExportPresentation](Y[n]=
$Ygeneral)");
$displayparticular=maple("MathML[ExportPresentation](Y[n]=
$Ypart)");
$y3=maple("round(eval($Ypart, n=3))");
$y20=maple("round(eval($Ypart, n=20))");
$display3=maple("MathML[ExportPresentation](Y[3]=$y3)");
$display20=maple("MathML[ExportPresentation](Y[20]=$y20)");
```

In het **eerste invulveld** moet het antwoord $A \cdot 0.8^n + 70$ ingevuld worden.

Grading Code: `evalb(coeff($RESPONSE, $r^n)=A) and
evalb(abs(op(sort($RESPONSE))[2] - ($YPI)) < 0.02);`

Het betekent dat de coëfficiënt van 0.8^n van het antwoord van de student gelijk moet zijn aan A. En vervolgens moet het verschil tussen de tweede operand van het studenten antwoord en de constante \$YPI minder zijn dan 0.02.

Een **alternatieve Grading Code** is met behulp van substitutie:

`evalb(abs(subs({A=1, n=1}, $RESPONSE) - subs({A=1, n=1},
$Ygeneral)) < 0.01);`

Substitueer A=1 en n=1 in het antwoord van de student en het correcte antwoord en bereken de absolute waarde van het verschil. Dat moet dan kleiner zijn dan bijvoorbeeld 0.01.

In het **tweede invulveld** van deze vraag moet als antwoord ingevuld worden het volgende:

$-72.5 \cdot 0.8^n + 70$. Het juiste antwoord is \$Ypart

Hierin mag het getal \$Aa = -72.5 afgerond zijn op 2 decimalen en het getal \$YPI = 70 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 -72.5). Het spreekt vanzelf dat de macht het grondtal 0.8 (\$r) moet hebben; daar zit geen marge in.

Verder moet de tweede operand van de uitdrukking, nadat deze op volgorde is gesorteerd, (hier is dat het getal 70) binnen een marge van 0.02 vallen met het correcte getal \$YPI.

Omdat het gaat om het afronden van de coëfficiënten 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 \$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.

Een **alternatieve Grading Code is met behulp van substitutie**. Neem dan een handig getal.

```
evalb(abs(subs(n=1,$RESPONSE)-subs(n=1,$Ypart))<0.01);
```

1.4.12 Breuken

Bekijk eerst nog eens paragraaf *Wanneer zijn formules gelijk aan elkaar?* (page 51) 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 1.73* (page 56). 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 55) 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 1.79* (page 65)).

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.



Figure 1.87: Breuken met instelling Text entry only

De *Grading Code* is:

```
evalb(simplify($RESPONSE-($antwoord))=0) and
  evalb(type($RESPONSE, `*`) ) and
evalb(StringTools[Search]("simplify", "$RESPONSE")=0) and
evalb(StringTools[Search]("normal", "$RESPONSE")=0);
```

In dit geval is gekozen voor `evalb(simplify($RESPONSE-($antw))=0)`; het is wel robuust, maar er is in dit geval ook voor gezorgd dat het antwoord door de student ingevoerd beslist in de vorm van één breuk is door middel van `evalb(type($RESPONSE, `*`) )`. Ook is in de *Grading Code* geprogrammeerd dat de Maple commando's **normal** en **simplify** niet in het antwoord van de student mogen voorkomen. Het correcte antwoord is reeds in het *Algorithm* geprogrammeerd als: `$vraag=maple("$a/($y+$c)+$b");` en `$antw=maple("simplify($vraag)");` Uiteindelijk zal het antwoord van de student zoals afgebeeld in *Figure 1.87 (page 76)* dus niet als correct beoordeeld worden, want de vorm ervan voldoet niet aan de vereenvoudigde vorm.

TIP: Zorg ook weer dat in de *Preview* de automatische vereenvoudiging niet plaatsvindt (zie *page 62*)

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 64)*.

In het onderstaande voorbeeld is te zien wat het verschil is tussen de opties *Symbol entry only* en *Text entry only*. Bij de laatste is het mogelijk met *Preview* de automatische vereenvoudiging te zien! Met 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.

Preview

Herleid tot één breuk en vereenvoudig zo goed mogelijk (geen decimale getallen gebruiken).

$$\frac{b-2}{b+6} \frac{b+3}{b-2}$$

Het antwoord is

$$\frac{b-2}{b+6} \frac{b+3}{b-2}$$

Figure 1.88: 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 64), 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 met een Custom Preview Code. Zie daarvoor paragraaf (page 62).

TIP: Om het invoerveld voor de editor iets minder hoog te krijgen voert u bij Custom CSS in:

```
.mwEquationEditor {height:100px !important}
```

TIP: Geef in het *Answer*-veld van een dergelijk invoerveld met editor altijd de MathML gecodeerde formule als correct antwoord!

```
Dus met printf(MathML[ExportPresentation]($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. Er zijn veel redenen om deze instelling **niet te gebruiken!**

In het volgende voorbeeld (Figure 1.90 (page 78)) 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:

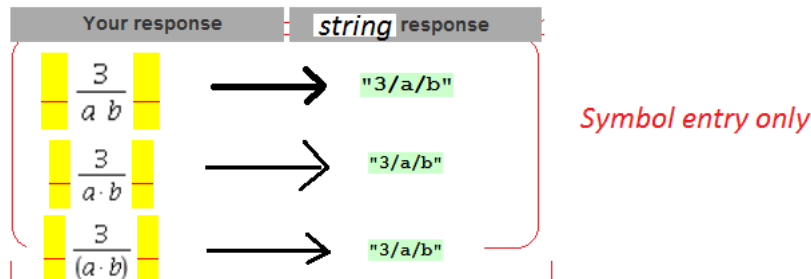


Figure 1.89: Breuken invoeren in de Editor

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!

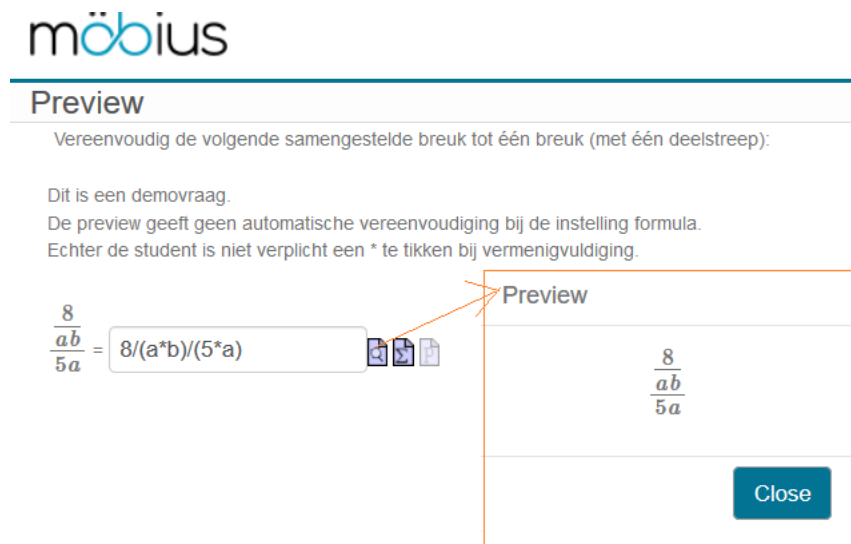


Figure 1.90: Breuken invoeren als text bij de instelling Formula

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");
$vraag=maple("$a/($L1*$L2)/($b*$L1)");
$vraagdisplay=mathml("$a/($L1*$L2)/($b*$L1)");
$santwoord=maple("simplify($vraag)");
```

```
$antwoorddisplay:=maple("printf(MathML[ExportPresentation]($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.

Het $=$ -teken achter de breuk geeft aan wat de hoofdbreukstreep is.

De variabelen $L1$ (in dit geval dus de a in de samengestelde breuk) moeten bij elkaar genomen

worden tot $\frac{8}{5a^2b}$.

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 55))

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 1.89* (page 78). U kunt eventueel zelf uitproberen hoe de invoer vertaald wordt naar een string in bovenstaande situatie bij de instelling *Formula* van *Figure 1.90* (page 78).

Bij de instelling *Formula* wordt bijvoorbeeld de invoer $8/(5a^2*b)$ vertaald naar $"(8/((5*(a^2))*b))"$.

Het is dus lastig om het aantal haakjes en het aantal sterren te controleren omdat het programma die er automatisch bij maakt als de student ze zelf niet typt.

Ook is het lastig om in het geval de student overschakelt op de editor dat dan het aantal slashes, sterren en dergelijke in de string van het antwoord ook niet meer klopt met wat de student invoert.

TIP: Bij deze instelling *Formula* is het dus heel lastig om haakjes en slashes te controleren met behulp van StringTools!!

En zeker het aantal vermenigvuldigingstekens is lastig te controleren want het systeem zet er gewoon extra sterren in waar de student die niet tikt.

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 ($teller)/($noemer): end:
printf(InertForm:-ToMathML(%));
```

Uiteindelijke conclusie: deze instelling liever niet gebruiken.

4) Met StringTools werken om het antwoord te checken en met inert form om in de preview de automatische vereenvoudiging tegen te houden.

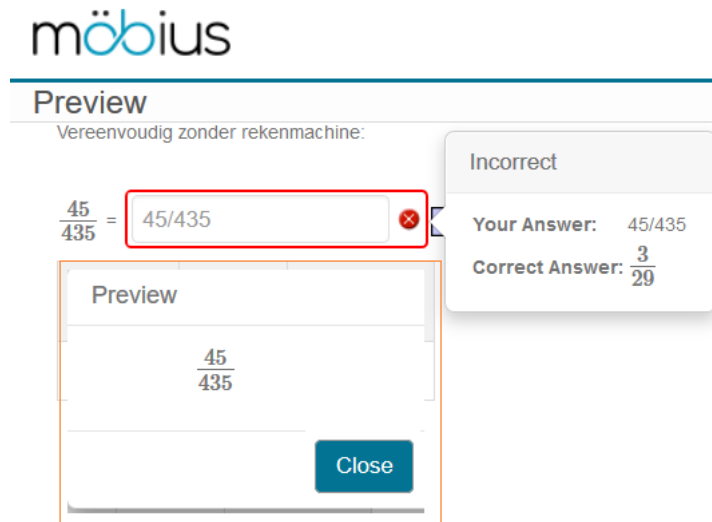


Figure 1.91: Breuken vereenvoudigen

In het **Algorithm** programmeert u het volgende:

```
$A=range(1,9);
$a=5;
$B=range(1,6);
#ithprime is het zoveelste priemgetal vanaf 2.
$b=maple("ithprime($B)");
$C=range(7,12);
$c=maple("ithprime($C)");
$D=range(1,9);
$d=3;
$abd=$a*$b*$d;
$acd=$a*$c*$d;
$vraag=mathml("($abd)/($acd)","nosimplify");
$antwoord=maple("simplify($abd/$acd)");
$stringantwoord=maple("convert($antwoord,string)");
$antwoorddisplay=maple("printf(MathML:-ExportPresentation(($antwoord)))");
```

De *Grading Code* van dit antwoordveld is:

```
stringrespons:=StringTools[Remove](IsSpace, "$RESPONSE");
evalb(stringrespons=$stringantwoord);
```

In de *Custom Preview* programmeert u het volgende:

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

5) 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 1.92: 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);
#bovenstaande betekent dat als a1 niet gelijk is aan b, dan a1, anders a1+1.
$vraagdisplay=mathml("1/($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)");
#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 $gcd. 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. U doet dat door een # vooraan de regel te plaatsen.

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.

1.4.13 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 55).

Maar eerst nemen we de invoer van de student nog eens even onder de loep bij de instelling **Maple Syntax Text entry only**.

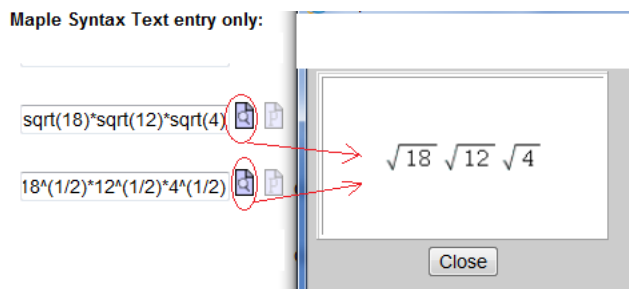


Figure 1.93: Hoe worden wortels vereenvoudigd

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 64) gaan we nu bekijken wat het oplevert als we de string van het antwoord en ook het antwoord zelf onder de loep 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 1.94: De respons en de stringrespons bij verschillende manieren van intikken van de wortels

In bovenstaande figuur ziet u in de eerste regel dat de notatie van de wortels, door de student ingevoerd met `"sqrt"`, exact overgenomen wordt in de string.

In de tweede regel in de respons van de student worden de wortels omgezet naar de macht $(1/2)$ en tegelijk ook vereenvoudigd, maar niet samengenomen.

In de derde regel: Voert de student de wortels in met behulp van $^(1/2)$, dan wordt de string van de invoer weer precies overgenomen zoals verwacht bij deze instelling van *Text entry only*,

In de vierde regel ziet u dat de respons vreemd genoeg niet wordt 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 een macht met exponent ($1/2$).

Wortel vereenvoudigen

möbius

Preview

Vereenvoudig de volgende wortelvorm

$\sqrt{96}$

Het antwoord moet in twee velden ingevoerd worden: eerst het getal en dan de wortel met `sqrt(...)`.

4 × sqrt(6)

• Number Help

numeriek veld maple-graded veld

Figure 1.95: Wortels vereenvoudigen

In bovenstaande figuur ziet u een vraag waar de vereenvoudiging van een wortel getoetst wordt. U biedt twee velden aan: De eerste is een getal en de tweede een wortel.

De *Preview* van de instelling *Maple syntax Text entry only* geeft twee dimensionaal weer wat de student heeft ingetikt.

In de *Feedback* wilt u graag dat de student ziet dat hij `sqrt(6)` moet intikken.

In de volgende figuur is in het bijbehorende *Algorithm* te zien hoe de mogelijke antwoorden zijn voorbereid.

1	<code>\$a1=switch(rint(4),2,3,5,6);</code>
2	<code>\$a2=switch(rint(2),2,3);</code>
3	<code>\$a3=switch(rint(3),2,3,5);</code>
4	<code>\$b=\$a2^2*\$a3^2;</code>
5	<code>\$c=\$a1*\$b;</code>
6	<code>\$displayvraag=mathml("(sqrt(\$c))");</code>
7	<code>\$antwoord=maple("simplify(sqrt(\$a1*\$a2^2*\$a3^2))");</code>
8	<code>#neem hiervan de eerste operand, dat is het getal voor de wortel</code>
9	<code>\$antw1=maple("op(1,\$antwoord)");</code>
10	<code>\$antw2="sqrt(\$a1)";</code>
11	<code>\$displayantwoord=maple("printf(MathML[ExportPresentation](\$antwoord))");</code>
12	

Variable	Value
a1	3
a2	2
a3	3
b	36
c	108
displayvraag	$\sqrt{108}$
antwoord	$6 \cdot 3^{1/2}$
antw1	6
antw2	$\sqrt{3}$
displayantwoord	$6 \sqrt{3}$

Figure 1.96: Wortels vereenvoudigen in Algorithm

In bovenstaande figuur ziet u het *Algorithm* waar de antwoorden worden voorbereid en gecontroleerd.

Merk op dat u bij \$antw2 geen maple-opdracht geeft om de sqrt(3) te krijgen.

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.

Met dit *Algorithm* is de feedback ook snel opgesteld:

Ontbind het getal \$c in priemfactoren $a_1 \cdot a_2^2 \cdot a_3^2$ en neem de wortel. Het antwoord is \$displayantwoord.

Het eerste antwoordveld (een numeriek type) heeft als correct antwoord het getal \$antw1.

Voor het correcte antwoord *Answer* van het tweede antwoordveld (Maple-graded) vult u in `printf("$antw2");`

Voor de *Grading Code* vult u in `evalb($antw2=$RESPONSE);`

Als de student het antwoord geeft met behulp van $3^{1/2}$ wordt het ook goed gerekend.

Wortelvorm met breuk

In dit 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 1.97: 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;
$antwoord=simplify((-($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 1.98: Algorithm voor wortels

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(($antwoord)-($RESPONSE))=0) and
evalb(StringTools[Search]("$D","$RESPONSE")=0);
```

De operanden van een uitdrukking

In het bovenstaande ziet u dat er gebruikgemaakt wordt van operanden van het antwoord.

De operanden van het correcte antwoord en het antwoord van de student kunt u ook apart checken.

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 1.99: 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));

$$10, \frac{1}{2} \quad (1.57)$$

> op(2*sqrt(10));

$$2, \sqrt{10} \quad (1.58)$$

> op(2*sqrt(6)*7*sqrt(50));

$$70, \sqrt{6}, \sqrt{2} \quad (1.59)$$

> op(3/sqrt(2));

$$\frac{3}{2}, \sqrt{2} \quad (1.60)$$

> op([2,2],3/sqrt(2));

$$\frac{1}{2} \quad (1.61)$$

In bovenstaande Maple-opdrachten wordt ook duidelijk dat de getallen steeds automatisch bij elkaar genomen worden, maar de wortels niet (tenzij er twee gelijke wortels voorkomen, die wel bij elkaar genomen worden).

In de laatste regel wordt er met een lijstje gevraagd naar de tweede operand en daarvan wordt de tweede operand genomen.

De wortel $\text{sqrt}(2)$ bestaat dus uit twee operanden want het is in feite $2^{(1/2)}$. De eerste operand hiervan is 2 en de tweede operand is $1/2$.

Wortelvereenvoudigen met checken van operanden

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. Maak 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 in een Maple-graded vraag waarbij u nog bepaalt of u een editor aanbiedt of dat u een tekstveld aanbiedt.

In de figuur hier onder is de editor aangeboden voor de student.

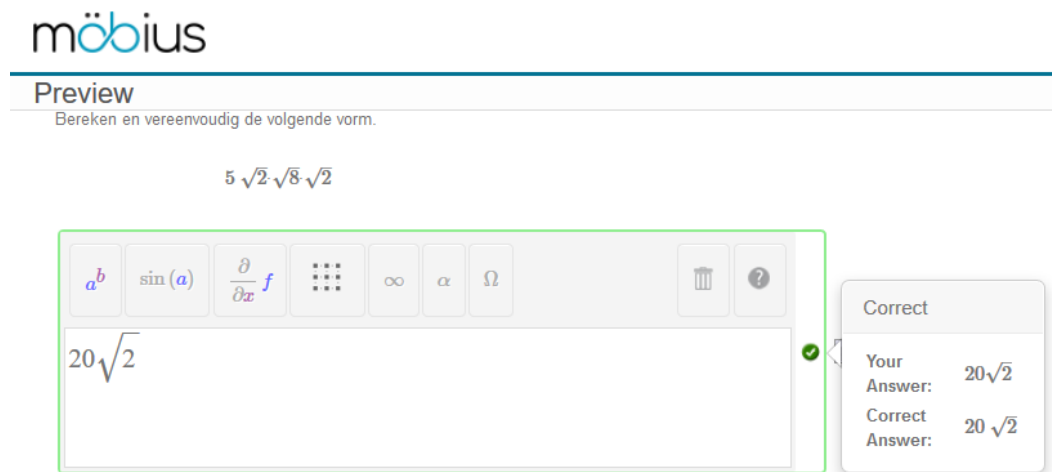


Figure 1.100: Wortels vereenvoudigen

Bij deze vraag is het lastig 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)");

#het aantal operanden van het antwoord mag niet gelijk zijn aan 1 want dan is het
#antwoord alleen een getal.
condition:ne($n,1);
condition:ne($n,1);
$displayantwoord=maple("printf(MathML[ExportPresentation]($antwoord))");
$op=maple("op([2,1],$antwoord)");
```

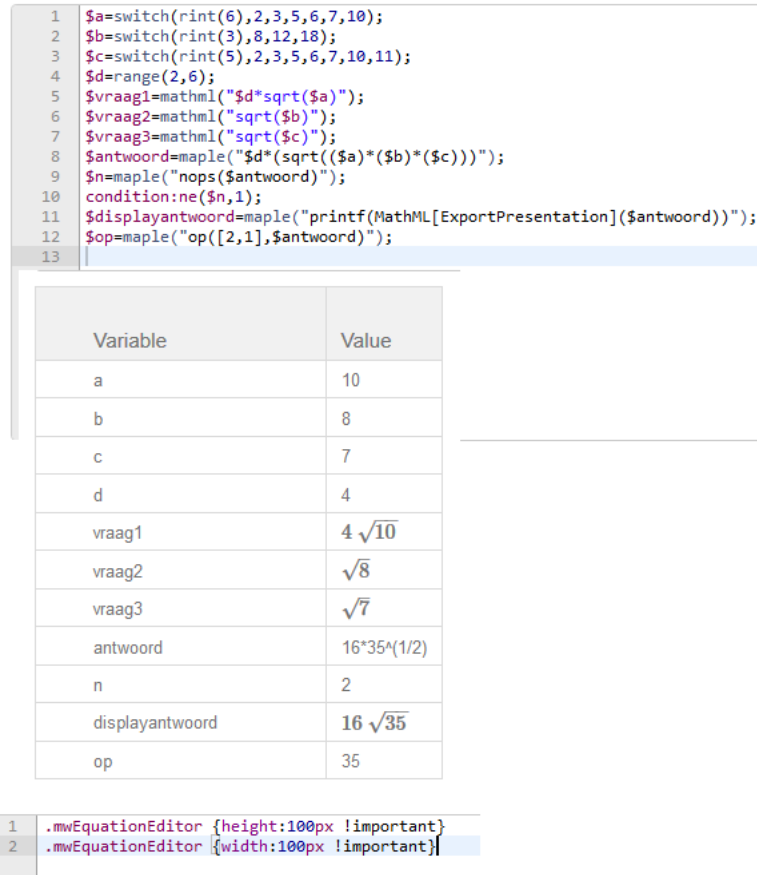


Figure 1.101: Operanden van een wortelvorm

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 u er zeker van bent dat de wortels elkaar niet uitdoven.

Met de variabele `$op` is het mogelijk om de grading code weer controleren of de wortel uit `$op` in het stringantwoord voorkomt in de vorm van `"sqrt($op)"`.

Bij de instelling met *Symbol entry only* wordt ALTIJD de wortel vertaald naar `sqrt`.
`evalb(op(1,$antwoord)=op(1,$RESPONSE)) and is(op(2,$antwoord)=op(2,$RESPONSE)) and evalb(StringTools[Search] ("sqrt($op)", "$RESPONSE")>0);`

TIP: Bij de instelling van *Text entry only* is dit laatste niet zo handig want de student kan ook de wortel invoeren met $^{1/2}$ tenzij u wilt afdwingen dat studenten met `sqrt` werken.

TIP: Om het invulveld van de editor iets kleiner te maken vult u een code in bij Custom CSS.

Bij het correcte antwoord vult u in: `printf(MathML[ExportPresentation]($antwoord));`

Wortels met StringTools

Nog een iets ingewikkelder te programmeren vereenvoudigingsvraag met wortels is de volgende.



Preview

Bereken en vereenvoudig de volgende vorm, waarbij eventuele in het antwoord voorkomende breuken als breuk (dus teller/noemer) geschreven moeten worden en geen wortels in de noemer.

$$\frac{9\sqrt{5}}{5\sqrt{9}}$$

Vul eerst het getal of de vereenvoudigde breuk in en daarna de wortel met `sqrt(...)`.

$$\frac{9\sqrt{5}}{5\sqrt{9}} = \frac{3}{5} \times \sqrt{5}$$

Figure 1.102: Wortels met breuken vereenvoudigen

De student moet eerst een vereenvoudigde breuk invoeren en vervolgens een vereenvoudigde wortel.

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);
$antw1=maple("op(1,$antwoord)");
$stringantw1=maple("convert($antw1,string)");
$antw2=maple("op(2,$antwoord)");
```

Om te controleren of de tweede operand van het antwoord een wortel is (dus van het type `^`), test u dat met `$n` en zorgt u ervoor dat `$n` niet gelijk is aan 1.

Als u 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.

Het eerste veld moet dus een breuk komen te staan die vereenvoudigd is. Dus het eerste antwoord zet u om naar een string en u vergelijkt het antwoord van de student met deze string met de volgende Grading Code:

```
stringrespons:=StringTools[Remove](IsSpace, "$RESPONSE");
evalb(StringTools[SubString](stringrespons, 1..-1)=$stringantw1);
```

Vergeet daarbij niet in de Custom Previewing Code in te voeren:

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

Het tweede veld moet een vereenvoudigde wortel staan. Dit is gemakkelijk te controleren met de Grading Code:

```
evalb(($antw2)-($RESPONSE)=0);
```

Wortels met LaTeX

In het volgende voorbeeld geeft de student een getal in een numeriek veld en een wortel in een maple-graded-veld.



Preview

Vereenvoudig de volgende optelling zo goed mogelijk.

$$4\sqrt{8} + 5\sqrt{98} - 4\sqrt{2}$$

Vul eerst het getal in en dan de wortel met behulp van sqrt(...).

×

[Number Help](#)

Figure 1.103: Wortels optellen

Met het volgende algorithm:

```
$a=switch(rint(2),2,3);
$B=switch(rint(3),4,9,16);
$b=$a*$B;
$C=switch(rint(3),25,36,49);
$c=$a*$C;
$D=switch(rint(2),1,100,$a);
$d=$a*$D;
$p=range(2,5);
$q=range(2,5);
$r=range(2,5);
$pm1=switch(rint(2),"+","-");
$pm2=switch(rint(2),"+","-");
$vraagdisplay="\( $p\sqrt{$b} $pm1 $q \sqrt{$c} $pm2 $r \sqrt{$d}
\)";
$antwoord=maple("simplify($p*sqrt($b) $pm1 $q*sqrt($c) $pm2
$r*sqrt($d))");
$antw1=maple("op(1,$antwoord)");
$antw2=maple("op(2,$antwoord)");
```

Hierin is ervoor gezorgd dat het antwoord altijd bestaat uit twee operanden: een geheel getal en een wortel.

Het eerste antwoord is voor het numerieke veld en het tweede antwoord is voor het maple-graded veld.

De vraag \$vraagdisplay voor op het scherm in de tekst is nu gemaakt met behulp van LaTeX (zie de paragraaf over LaTeX (page 27)).

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 135).

1.4.14 Vergelijkingen

Bij het matchen van vergelijkingen komt het er niet simpel op neer dat u de vergelijkingen van elkaar afrekt 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!!! Gebruik dan het commando `convert(vergelijking, rational)` om de decimalen kwijt te raken.

Een vergelijking invoeren

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 1.104: 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 brengt 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 (greatest common divisor) 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.

Robuuster 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 herleid 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 goed om in de *Custom Previewing Code* te programmeren:

```
if evalb(StringTools[CountCharacterOccurrences]
("$RESPONSE","=")=0) then printf("voer een vergelijking in
(met een -=teken).") else printf(MathML[ExportPresentation]
($RESPONSE)) end if;
```

Vergelijking met decimalen

Als u met decimalen werkt, rond dan de vergelijking van het antwoord af met een aantal significante cijfers (met `evalf[5]` bijvoorbeeld). Vervolgens kunt u de oplossingen van beide vergelijkingen matchen.



Figure 1.105: Vergelijkingen met decimalen

In de *Grading Code* kunt u programmeren:

```
evalb(evalf[3](solve($verg,x))=evalf[3](solve($RESPONSE,x))) or
evalb(evalf[3](solve($verg,x))=evalf[3](solve($RESPONSE,x)))
and type($RESPONSE,equation);
```

Zie ook in paragraaf *Verzamelingen* (page 67) voor nog meer tips bij het matchen van verzamelingen, zoals het geval waarin er decimalen mee gemoeid zijn.

Kwadratische vergelijking

In het volgende voorbeeld zien we dat er een vergelijking gevraagd wordt van een vervormde cirkel.

möbius

Preview

Geef de vergelijking in x en y van de ellips met de volgende toppen:
 $[3, 0]$ en $[-3, 0]$ en $[0, 4]$ en $[0, -4]$

Je kunt op de P klikken achter het invoerveld om de grafiek van de kromme te zien die je hebt ingevoerd.
 (Het moet wel de vergelijking van een ellips zijn, anders zie je geen grafiek.)
 Klik ook altijd even op het vergrootglasje om je invoer te controleren.

$1/9*x^2+1/16*y^2=1$

$1/9*x^2+1/16*y^2 = 1$

Preview

Preview

$$\frac{1}{9}x^2 + \frac{1}{16}y^2 = 1$$

Close

Close

Figure 1.106: Vergelijking van een ellips

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($antw,y)}={solve($RESPONSE,y)}) and
type($RESPONSE,equation);
```

Het betekent dat van de correcte vergelijking \$antw 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 67) voor nog meer tips bij het matchen van verzamelingen, zoals het geval waarin er decimalen mee gemoeid zijn.

TIP: Zie ook in paragraaf *Vergelijkingen* (page 139) om met het vraagtype *Formula* vergelijkingen te overhoren.

TIP: Zie ook in paragraaf *Verzamelingen* (page 67) 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 99) 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)");
```

1.4.15 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;
```

$$verg1 := 7a - 11b = 18$$

$$verg2 := 3a + 4b = 2 \quad (1.62)$$

```
> solve({verg1, verg2}, {a, b});
```

$$\left\{ a = \frac{94}{61}, b = -\frac{40}{61} \right\} \quad (1.63)$$

```
> solve([verg1, verg2], [a, b]);
```

$$\left[\left[a = \frac{94}{61}, b = -\frac{40}{61} \right] \right] \quad (1.64)$$

```
> verg3:=y=x^2+x-2; verg4:=y=4*x+2;
```

$$verg3 := y = x^2 + x - 2$$

$$verg4 := y = 4x + 2 \quad (1.65)$$

> `solve({verg3, verg4}, {x, y});`

$$\{x = -1, y = -2\}, \{x = 4, y = 18\} \quad (1.66)$$

> `solve([verg3, verg4], [x, y]);`

$$[[x = -1, y = -2], [x = 4, y = 18]] \quad (1.67)$$

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*.

Stelsel van twee lineaire vergelijkingen. De oplossing is één getallenpaar

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 1.107: Stelsel vergelijkingen

De bijbehorende *Algorithms* zijn hieronder te zien:

<pre> \$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])"); </pre>	<pre> \$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])"); </pre>																																		
<table> <thead> <tr> <th>Variable</th><th>Value</th></tr> </thead> <tbody> <tr><td>a</td><td>11</td></tr> <tr><td>v1</td><td>$7a - 11b = 18$</td></tr> <tr><td>v2</td><td>$3a + 4b = 2$</td></tr> <tr><td>displayv1</td><td>$7a - 11b = 18$</td></tr> <tr><td>displayv2</td><td>$3a + 4b = 2$</td></tr> <tr><td>opl</td><td>$\{a = 94/61, b = -40/61\}$</td></tr> <tr><td>displayopl1</td><td>$a = \frac{94}{61}$</td></tr> <tr><td>displayopl2</td><td>$b = -\frac{40}{61}$</td></tr> </tbody> </table>	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	$b = -\frac{40}{61}$	<table> <thead> <tr> <th>Variable</th><th>Value</th></tr> </thead> <tbody> <tr><td>a</td><td>12</td></tr> <tr><td>v1</td><td>$7a - 12b = 18$</td></tr> <tr><td>v2</td><td>$3a + 4b = 2$</td></tr> <tr><td>displayv1</td><td>$7a - 12b = 18$</td></tr> <tr><td>displayv2</td><td>$3a + 4b = 2$</td></tr> <tr><td>opl</td><td>$[[a = 3/2, b = -5/8]]$</td></tr> <tr><td>displayopl</td><td>$\left[a = \frac{3}{2}, b = -\frac{5}{8} \right]$</td></tr> </tbody> </table>	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	$\left[a = \frac{3}{2}, b = -\frac{5}{8} \right]$
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	$b = -\frac{40}{61}$																																		
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 1.108: Stelsel vergelijkingen met verzameling en met lijst

Voor de verzameling:

De bijbehorende *Grading Code*: `evalb(($opl)={$RESPONSE});`

Bij *Answer* vult u het correcte antwoord in: `$opl[1], $opl[2];`

U ziet dat de oplossing `$opl` een verzameling is en deze matcht u 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:

De bijbehorende *Grading Code*: `evalb( ($opl[1])=$RESPONSE);`

Bij *Answer* vult u het correcte antwoord in: `$opl[1];`

U vergelijkt 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 vierkante 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.

De afzonderlijke variabelen bevragen

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-10*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	7*a-10*b = 18
v2	3*a+4*b = 2
displayv1	7 a - 10 b = 18
displayv2	3 a + 4 b = 2
opl	[[a = 3/2, b = -5/8]]
opla	3/2
oplb	-5/8

Figure 1.109: 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.

Stelsel niet-lineaire vergelijkingen

In het volgende voorbeeld komen er meer oplossingen van een stelsel vergelijkingen. Er wordt gevraagd naar de coördinaten van de snijpunten van twee grafieken.

möbius

Preview

Bepaal de snijpunten van de grafieken van de volgende twee functies:

$$f(x) = x^2 + x - 2$$

$$g(x) = x + 2$$

Geef het antwoord in de vorm bijvoorbeeld:

$[x=20, y=30]$, $[x=3, y=5]$

zonder af te ronden met de rekenmachine.

$[x = 2, y = 4], [x = -2, y = 0]$



Figure 1.110: 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 1.111: Stelsel vergelijkingen met meer oplossingen Algorithm

Hierin ziet u dat de oplossing \$opl in de vorm van een lijst gegeven wordt. Het is een lijst bestaande uit twee lijsten.

Binnen maple kunt u beschikken over de afzonderlijke lijsten met \$opl[1] en \$opl[2].

Met de *Grading Code* van het invulveld is het een kwestie van verzamelingen bestaande uit twee lijstjes matchen, zodat de volgorde van de lijstjes er niet toe doet:

```
evalb({$opl[1], $opl[2]}={$RESPONSE});
```

Voor het correcte antwoord vult u in: `$opl[1], $opl[2];`

Er is voor gezorgd dat het mooi uitkomt, zodat de student niet in de verleiding komt om numerieke benaderingen toe te passen.

Numeriek oplossen van een stelsel niet-lineaire vergelijkingen

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 1.112: 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:

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

```
$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");  
$displayf:=maple(" printf(MathML[ExportPresentation](f(x)=rhs($f))) ");  
$displayg:=maple(" printf(MathML[ExportPresentation](g(x)=rhs($g))) ");  
$opl:=maple("solve([$f1,$g],[x,y])");  
$opl1:=maple("rhs~($opl[1])");  
$opl2:=maple("rhs~($opl[2])");
```

Variable	Value
a	5
b	2
f	$y = x^2 + 5x + 3$
f1	$y = x^2 + 5x + 3.0$
g	$y = -2x^2 + 2x + 8$
displayf	$f(x) = x^2 + 5x + 3$
displayg	$g(x) = -2x^2 + 2x + 8$
opl	$[[x = .8844373105, y = 8.204415909], [x = -1.884437310, y = -2.871082575]]$
opl1	$[.8844373105, 8.204415909]$
opl2	$[-1.884437310, -2.871082575]$

Figure 1.113: Algorithm van een vraag over het berekenen van snijpunten van grafieken

In bovenstaand *Algorithm* ziet u 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 krijgt u de oplossing ook in de vorm van decimale getallen, wat u hier ook wilt.

Omdat er in de opgave naar de coördinaten van de snijpunten gevraagd wordt bereidt u 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 kunt u dan geen lijstjes met decimale getallen met elkaar vergelijken en daarom vermenigvuldigt u de elementen van die lijstjes met 100 en rondt 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 (de coördinaten (x,y) 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.

1.4.16 Ongeordende en geordende lijsten

Bij vergelijkingen met meer dan één oplossing kunt een geordende of ongeordende lijst overhoren 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 67) genoemd is.

Ongeordende lijst met verzameling

In het volgende voorbeeld gaat het over het exact oplossen van een kwadratische vergelijking.



Preview

Bepaal de oplossingen van de kwadratische vergelijking:

$$(5z - 9)^2 = 64$$

Geef de antwoorden gescheiden door een komma en vul alleen getallen in zonder af te ronden met de rekenmachine.

17/5, 1/5

✓
📄
📄
(gedeeltelijke beoordeling)

of

17/5, 1/5

✓
📄
📄
goed of fout

Figure 1.114: title of the figure

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])");
```

De *Grading Code* voor één antwoordveld is dan als volgt:

```
aantal:=nops({$RESPONSE} intersect {$sopl}): 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. Zo kunt u ook de vraag gedeeltelijk goedrekenen.

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({$RESPONSE} intersect {$sopl}): evalf(aantal/2) ;
```

TIP: Als het gaat om **decimale getallen**, converteer dan de verzamelingen eerst naar verzamelingen met gehele getallen, want anders lukt het matchen niet.

De elementen van de verzameling bijvoorbeeld met 100 vermenigvuldigen en daarna afronden:

```
aantal:=nops(map(x->round(100*x),{$RESPONSE}) intersect map(x->round(100*x),{$sopl})):
if StringTools[Search]("solve", "RESPONSE")>0 then 0 else
  evalf(aantal/2) end;
Of als u geen gedeeltelijke beoordeling doet:
evalb(map(x->round(100*x),{$antw1a,$antw2a}) = map(x->round(100*x),{$RESPONSE}));
```

Zoals behandeld in paragraaf *Verzamelingen* (page 67).

TIP: voor gedeeltelijk goedrekenen met verschillende gewichten voor bijvoorbeeld drie antwoorden heeft u de volgende Grading Code:

```
if evalb(StringTools[Search]("solve", "$RESPONSE")=0) then
`if`(member($antw1,{$RESPONSE}),0.3,0)
```

```

+
`if`(member($antw2, {$RESPONSE}), 0.3, 0)
+
`if`(member($antw3, {$RESPONSE}), 0.4, 0)
else 0 end if;
evalb({$opl} = {$RESPONSE});

```

TIP: In dit geval zijn er twee antwoorden voor de oplossing van de vergelijking. Deze antwoorden kunnen 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.

Geordende lijst met vector

Geordende lijsten matchen gaat alleen als alle elementen exact aan elkaar gelijk zijn.

Een lijst is altijd geordend als er rechte haken omheen staan.

In het volgende voorbeeld zijn de lijsten A en B niet precies exact aan elkaar gelijk want het eerste element is een decimaal getal en het tweede een geheel getal.

Wilt u dus **lijsten matchen** zorg dan dat alle elementen rationaal zijn óf ga over op vectoren.

```
> A:=[3.0, 5.23, 1/3]; B:=[3, 5.23, 1/3];
```

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

$$B := \left[3, 5.23, \frac{1}{3} \right] \quad (1.68)$$

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

$$\text{false} \quad (1.69)$$

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

$$\left[3, \frac{523}{100}, \frac{1}{3} \right] \quad (1.70)$$

Hier zijn alle elementen van de lijst A geconverteerd naar rationale getallen.

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

$$\text{true} \quad (1.71)$$

```
> LinearAlgebra[Equal](Vector(A), Vector(B));
```

$$\text{true} \quad (1.72)$$

In het volgende voorbeeld moet de student in het invulveld een geordende lijst typen. Deze geordende lijst is in feite een vector.

Het is handig om die dan ook als zodanig te behandelen.

möbius

Preview

Gegeven de twee vectoren:

$$\vec{a} = \begin{pmatrix} -7.1 \\ 2.8 \end{pmatrix} \text{ en } \vec{b} = \begin{pmatrix} 4.3 \\ -5.5 \end{pmatrix}$$

Bereken het verschil $\vec{a} - \vec{b}$ van deze vectoren.

HINT: Voer bijvoorbeeld de vector $\begin{pmatrix} -7.1 \\ 2.8 \end{pmatrix}$ in als [-7.1,2.8] en klik op Preview (vergrootglasje).

Als je op de P klikt achter het invulveld, dan zie je een grafiek van de vectoren $\vec{a}$ en $-\vec{b}$.

De grijze pijl van jouw antwoord moet met deze vectoren de diagonaal vormen van het parallellogram.

$\vec{a} - \vec{b} =$  

Figure 1.115: Vector als geordende lijst

Het Algorithm van deze vraag is als volgt:

```
$a1=switch(rint(2),decimal(1,rand(-10,-1)),decimal(1,rand(1,10)));
$a2=switch(rint(2),decimal(1,rand(-10,-1)),decimal(1,rand(1,10)));
$b1=switch(rint(2),decimal(1,rand(-10,-1)),decimal(1,rand(1,10)));
$b2=switch(rint(2),decimal(1,rand(-10,-1)),decimal(1,rand(1,10)));
$dets=maple("abs($a1*($b2)-($b1)*($a2))");
condition:gt($dets,3);
$A=maple("Vector([$a1,$a2])");
$Adisplay=maple("printf(MathML[ExportPresentation]($A))");
$B=maple("Vector([$b1,$b2])");
$Bdisplay=maple("printf(MathML[ExportPresentation]($B))");
$AminusB=maple("$A-($B)");
$AminusBdisplay=maple("printf(MathML[ExportPresentation]($AminusB))");
$P=plotmaple("plots[display]({plottools[arrow]([0,0],
$A,0.08,0.3,0.2,color=red),plottools[arrow]([0,0],-
($B),0.08,0.3,0.2,color=blue),plottools[arrow]([0,0],
$AminusB,0.07,0.2,0.1,color=gray),plottools[line]
(convert($AminusB,list),convert($A,list),linestyle=dash),plottools[line]
(convert($AminusB,list),convert(-
($B),list),linestyle=dash)},scaling=constrained)");
```

Voor de *Grading Code* maakt u in dit geval van de lijsten een vector en u matcht de vectoren met elkaar. U zorgt er meteen ook voor dat er maar één lijst wordt getypt door de student door de string van het antwoord te controleren op aantal vierkante haken.

```
LinearAlgebra[Equal]($AminusB,Vector($RESPONSE)) and
evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","["<2);
```

Als er echt afrondingen op twee decimalen moeten plaatsvinden, kunt u over gaan op het afronden van correcte antwoord en studenten antwoord en deze twee vectoren met elkaar matchen.

```
LinearAlgebra[Equal]
(map(round,100*($AminusB)),map(round,100*(Vector($RESPONSE)))) and
evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","["<2);
```

Besteed ook enige aandacht aan de feedback met het geven van het correcte antwoord en eventueel een opmerking met behulp van de Custom Preview-code.

TIP: Zie verder in een paragraaf over geordende en ongeordende lijsten met een ander vraagtype (page 138).

TIP: Zie verder in een paragraaf over vectoren (page 125).

1.4.17 Ongelijkheden

Ongelijkheden testen vereist enige kennis van het computeralgebrasysteem Maple omdat u toch daaraan veel zult uitbesteden.

Ten eerste heeft Maple de onhebbelijke gewoonte om altijd alles met het $<$ -teken te presenteren. Hoewel $x > -2.82$ wel als hetzelfde wordt gezien als $-2.82 < x$, zal Maple altijd de laatste vorm met het $<$ -teken presenteren, zodat er eenduidigheid bestaat wat betreft linkerlid en rechterlid van de ongelijkheid. Eventueel moet de opgave aangepast worden wat presentatie betreft omdat Maple de ongelijkheid eerst zo veel mogelijk zal vereenvoudigen alvorens de mathml-code aan te maken. De MathML-codering zal waarschijnlijk niet altijd de ongelijkheid opleveren die u wilt aanbieden. U kunt daarbij gebruikmaken van voorbeeld mathml en dan ook het linkerlid en het rechterlid van de ongelijkheid apart aanbieden. Of met LaTeX werken

De ongelijkheid kan opgelost worden met behulp van Maple, maar dan moet dat wel op een speciale manier. Numeriek oplossen kan sowieso al niet, maar dan kan er altijd nog wel het commando evalf gegeven worden na solve.

Het solve commando moet altijd gepaard gaan met de onbekende in de vorm van een verzameling, dus `solve(ongelijkheid, {x})`. U krijgt dan namelijk de antwoorden in de vorm van ongelijkheden: de ene keer een rijtje verzamelingen en de andere keer een enkele verzameling. Iets om op te letten als u verzamelingen wilt matchen in de *Grading Code*.

```
> restart; ong1:=4*x+5>-4;
```

$$ong1 := 0 < 4x + 9 \quad (1.73)$$

```
> lhs(ong1); rhs(ong1);
```

$$0$$

$$4x + 9 \quad (1.74)$$

```
> solve(ong1, {x});
```

$$\left\{ -\frac{9}{4} < x \right\} \quad (1.75)$$

```
> ong2:=x^2>8; ong3:=x^2<8;
```

$$ong2 := 8 < x^2$$

$$ong3 := x^2 < 8 \quad (1.76)$$

```
> solve(ong2, {x});
```

$$\{x < -2\sqrt{2}\}, \{2\sqrt{2} < x\} \quad (1.77)$$

```
> evalf[3](solve(ong3, {x}));
```

$$\{-2.82 < x, x < 2.82\} \quad (1.78)$$

Er zijn verschillende mogelijkheden om de oplossing van een ongelijkheid op juistheid te controleren.

Hieronder geven we verschillende voorbeelden om te variëren in aanpak.

Lineaire ongelijkheid

De student heeft hier een apart invulveld voor het $<$ of $>$ -teken en een numeriek veld voor het invullen van het getal (eventueel met een marge). Dit heeft het voordeel dat afrondingen goed afgehandeld kunnen worden.

Preview

Gegeven is de ongelijkheid

$$\frac{3x-7}{3} > \frac{x-8}{6}$$

Bepaal de oplossing van de ongelijkheid en rond zonodig af op 2 decimalen.

x

Click for List

>

<

Figure 1.116: Lineaire ongelijkheid

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");
$ong1=maple("(3*x-$a)/3 $teken (x-8)/$b");
$opl1=maple("solve($ong1,{x})");
$Getal=maple("$rechtslinks($opl1[1])");
$opl=maple("solve((3*x-$a)/3 = (x-8)/$b,x)");
```

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 in de tekst van de vraag 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 hier 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)"); (Er is slechts één antwoord bij deze lineaire vergelijking.)

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.

Ongelijkheid met intervallen

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.

Om open antwoordvelden te maken met intervallen wordt verwezen naar Handleiding Items maken Deel C.

möbius

Preview



Bepaal de coördinaten van de snijpunten van de grafieken van de volgende twee functies:

$$f(x) = x^2 + 5x + 3$$

$$g(x) = -x^2 + 2x + 8$$

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] = [-2.50, -3.25], [1, 9]

Correct response: [-2.50, -3.25], [1., 9.]

Voor welke waarden van x is $f(x)$ groter dan $g(x)$?

- ☐ [1., ∞)
- ☐ (-∞, -2.50]
- ☐ (-2.50, 1.)
- ☐ (-∞, -2.50) ∪ (1., ∞)
- ☐ [-2.50, 1.]
- ☐ (-∞, -2.50] ∪ [1., ∞)

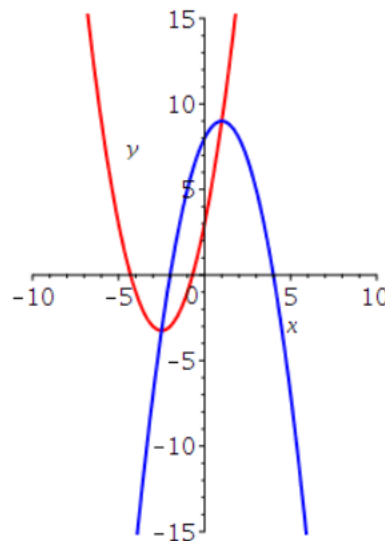


Figure 1.117: Ongelijkheid met intervallen

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 94).

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, 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");
$displayf=maple(" printf(MathML[ExportPresentation](f(x)=rhs($f)))
");
$displayg=maple(" printf(MathML[ExportPresentation](g(x)=rhs($g)))
");
$opl=maple("solve([$f1,$g],[x,y])");
$opl1=maple("rhs~($opl[1])");
$opl2=maple("rhs~($opl[2])");
$oplossing=maple("solve(rhs($f1)=rhs($g),x)");
$oplmin=maple("evalf[3](min([$oplossing]))");
$oplmax=maple("evalf[3](max([$oplossing]))");
$displayopl1=maple(" printf(MathML[ExportPresentation]($opl[1]))
");
$displayopl2=maple(" printf(MathML[ExportPresentation]($opl[2]))
");
$plot1=plotmaple("plot([rhs($f),rhs($g)],x=-10..10,y=-15..15,color=[red,blue],thickn
width=250'");
$plot=plotmaple("plot([rhs($f),rhs($g)],x=-10..10,y=-15..15,color=[red,blue],thickne
width=250'");
$optie1=maple("printf(MathML[ExportPresentation](x>$oplmin))");
$optie2=maple("printf(MathML[ExportPresentation](x<$oplmin))");
$optie3=maple("printf(MathML[ExportPresentation](x>$oplmax))");
$optie4=maple("printf(MathML[ExportPresentation](x<$oplmax))");
$test1="\([$oplmin,$oplmax]\)";
$test2="\((- \infty, $oplmin]\)";
$test3="\([$oplmax, \infty)\)";
$test4="\(($oplmin, $oplmax)\)";
$test5="\((- \infty, $oplmin] \cup [$oplmax, \infty)\)";
$test6="\((- \infty, $oplmin) \cup ($oplmax, \infty)\)";

```

De variabelen \$test 1 t/m 6 kunnen aangeroepen worden bij het maken van de multiple-choce-vraag echter deze LaTeX-code (tekst) kan ook direct ingevuld worden bij de alternatieven van het multiple-choice-invulveld.

TIP: Zie voor het maken van de dynamische figuur in paragraaf *Dynamische figuren* (page 142).

Om open antwoordvelden te maken met intervallen wordt verwezen naar Handleiding Items maken Deel C.

1.4.18 Symbolen en Griekse letters

Bij de *Maple graded*-vragen is het mogelijk om ook formules met Griekse letters en enkele andere symbolen te overhoren.

Neem dan voor de instelling *Maple-syntax Symbol entry only* zodat de student een editor aangeboden krijgt.

Zonder editor kan het ook, maar dan moet de student wat meer instructie krijgen hoe de symbolen en Griekse letters ingevoerd dienen te worden.

TIP: Voor de verschillende instelling moet u rekening houden met verschillende Grading Codes.

Griekse letters

Alle kleine letters staan in een pallet onder de knop α . In de paletten met het ∞ -teken en met de Griekse hoofdletter Ω zijn ook nog enkele mogelijkheden.



Preview

test!

1) Vul in $\frac{1}{2} d(\alpha + \beta + \epsilon)^2$

Hier is geprogrammeerd dat eventueel de spatie achter de d weggelaten mag worden.

De griekse letters kunnen met de editor gemaakt worden.

In principe altijd spaties of een * tikken tussen letters als het om vermenigvuldiging gaat.

Anders worden lettercombinaties als één variabele gezien.

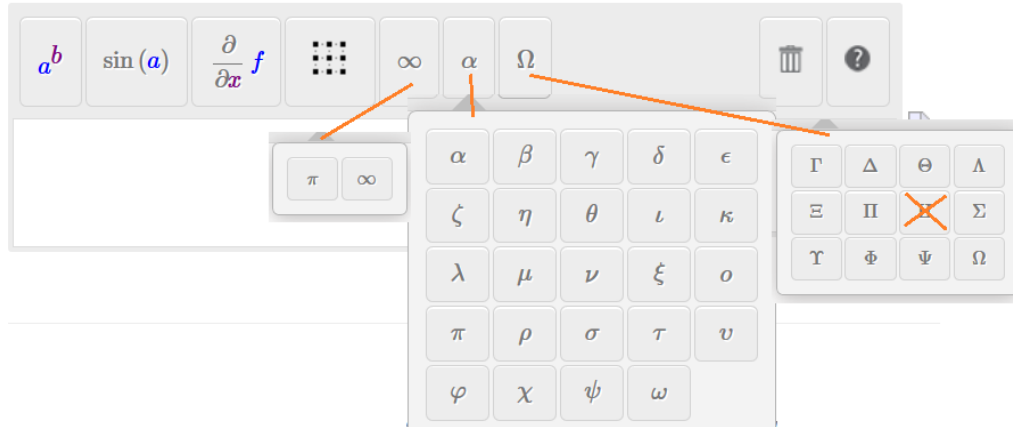


Figure 1.118: Griekse letters invoeren

TIP: Let ook op het prullenmandje rechts, waarmee de student even opnieuw kan beginnen als de formule niet lukt.

Ga steeds uit van de standaard Griekse letters zoals in Maple gedefinieerd. Dat wil zeggen de kleine letters met alpha, beta, gamma, delta etc.

De hoofdletters met Alpha, Beta, Gamma etc..

Enkele problematische letters en symbolen zijn de volgende. Hou daar dus rekening mee in de Grading Code en in de voorbereiding van de formules en de terugkoppeling van het correcte antwoord.

- Het symbool ∞ is bij maple bekend als infinity.
- De letter π bij maple bekend als Pi en als pi is in de beide palletten altijd bedoeld als de letter die bij maple bekend staat onder de naam Pi (3.14....).
- De letter ϵ die bij maple bekend staat onder de naam epsilon staat hier in het pallet onder de naam varepsilon (ϵ).
- De letter ζ die bij maple bekend staat onder de naam zeta staat hier in het pallet onder de naam Zeta.
- De hoofdletter Π staat bij maple bekend onder de naam PI en ook in de paletten onder dezelfde naam.
- De hoofdletter Γ (bekend onder de naam Gamma) heeft ook een speciale behandeling nodig. In de paletten staat hij bekend onder de naam GAMMA.
- De hoofdletter Υ is bij maple bekend onder de naam Upsilon, maar in de paletten is het een hoofdletter Y als die van het toetsenbord.
De grading lukt niet met de editor (is al gemeld bij support).

- Het symbool naast de hoofdletter PI doet het niet (support is op de hoogte).

Het antwoord en ook de vraag van bovenstaand voorbeeld worden voorbereid in het *Algorithm* zoals in dit voorbeeld hieronder te zien is.

```
$antw1:=maple("d(alpha+beta+epsilon)^2/2");
$antw1a:=maple("d*(alpha+beta+epsilon)^2/2");
$displayantw1:=maple("printf(MathML[ExportPresentation]($antw1))");
```

Er zijn voor deze situatie twee goede antwoorden geprogrammeerd vanwege wel of geen bedoelde spatie achter de d.

De *Grading Code* kan dan zijn:

```
evalb(($antw1)-($RESPONSE) = 0) or evalb(($antw1a)-($RESPONSE) = 0);
```

Het beste is eigenlijk 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. U zult zien dat deze Grading Code in dat geval niet voldoet, want door de editor wordt de letter ϵ vertaald naar varepsilon.

Om de *Grading Code* óók goed te laten werken in het geval u kiest voor de *Symbol entry only* neemt u de volgende codering:

```
evalb(subs(varepsilon=epsilon,($antw1)-($RESPONSE)) = 0)
or
evalb(subs(varepsilon=epsilon,($antw1a)-($RESPONSE)) = 0);
```

TIP: Eventueel kunnen de instellingen met deze laatste *Grading Code* 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 wel op dat als in de formule een ζ of een π gehanteerd worden dat deze letters met begin-hoofdletter vertaald worden: Zeta en Pi. Maple kent namelijk aan deze letters bepaalde functies of waarden toe. Dus hier ook met substitutie werken.

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. Het ziet er dan uit als

$\Delta \Delta x$. Het ziet er dan uit als $\hat{\Delta} x$.

De

De combinatie Deltax wordt namelijk niet gezien als een Griekse letter met daarachter een x. Dit is combinatie Deltax wordt namelijk niet gezien als een Griekse letter met daarachter een x. Dit is ook iets wat de studenten moeten weten en trainen.

1.4.19 Differentialen en differentiaalvergelijkingen

Voor het overhoren van differentialen en integralen is het vaak handig om de *Editor* ter beschikking te stellen met *Symbol entry only* bij de instellingen met *Maple-syntax*. Echter niet altijd, zoals we straks zullen zien (*Figure 1.120 (page 110)* waar uitgelegd wordt dat symbolische differentialen niet met de Editor overhoord kunnen worden).

Kennis van enige syntax voor zowel de 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 daarmee goedgerekend wordt. Dat geldt niet alleen bij de instellingen *Text entry only* maar ook voor het gebruik van de Editor (*Symbol entry*

only). 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*.

Een differentiaalvergelijking invoeren

Als de student een differentiaalvergelijking moet invoeren, stel dan de *Editor* beschikbaar met de instellingen *Maple syntax* met *Symbol entry only*. De student kan dan beschikken over de operator

$$\frac{d}{dx}$$

möbius

Preview

Geef de formule van de differentiaalvergelijking in v (de snelheid) afhankelijk van de tijd t die beschreven wordt door:

De kracht die op een voorwerp met massa m werkt, is evenredig met de snelheid.
Noem hier de evenredigheidsconstante k .

DV is

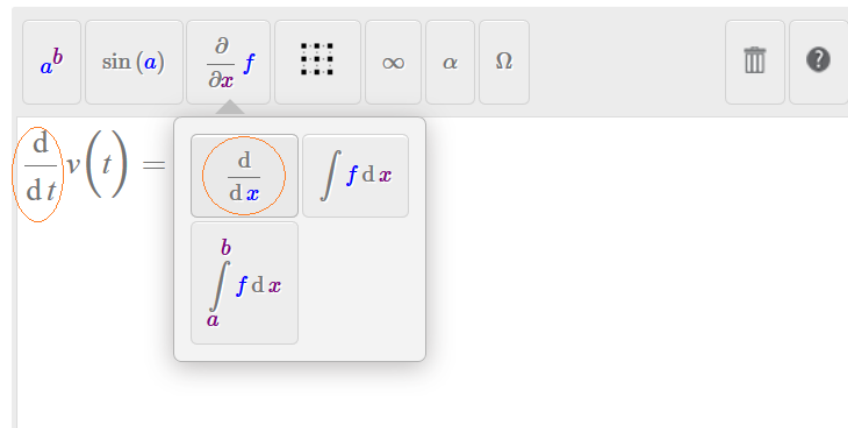


Figure 1.119: De student tikt een differentiaalvergelijking in

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))");
```

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).

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 $\text{diff}(v,t)$.

Met de instelling *Text entry only* zal het precies hetzelfde zijn want $\text{diff}(y,x)=0$.

TIP: Bij de instelling *Text entry only* is er nog een verschil te onderscheiden tussen diff en de inerte vorm Diff .

Bij de inerte vorm Diff wordt niet gerekend met 0 als $\text{Diff}(v,t)$ is ingevoerd. Deze inerte vorm is bij de editor niet mogelijk.

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.

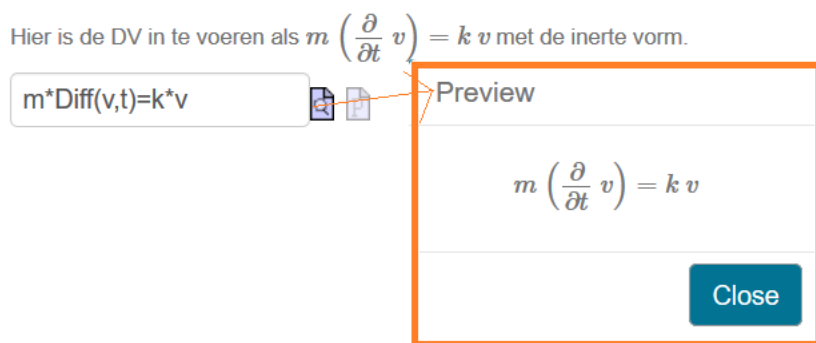


Figure 1.120: Differentiaal symbolisch met text invoer

Voor de *Grading Code* kunt u weer hetzelfde script gebruiken:

```
resp:=subs(v=v(t), $RESPONSE):
evalb(dsolve(resp, v(t), explicit)=dsolve($answ1, v(t), explicit));
```

Als de student tikt $\text{diff}(v,t)$ 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.

De eerste afgeleide kan ook met subscript geschreven worden door de student als hij daarvoor geïnstrueerd is.

(Accenten gaan slecht samen met *Grading Code*.)

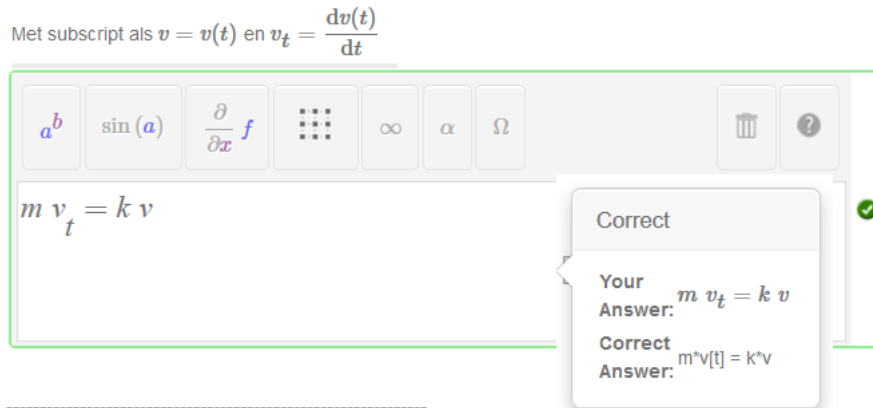


Figure 1.121: Afgeleide met subscript

In dit geval wordt de *Grading Code* voor beide instellingen (*Text en Symbol*):

```
resp:=subs({v=v(t), v[t]=diff(v(t), t)}, $RESPONSE):
evalb(dsolve(resp, v(t), explicit)=dsolve($answ1, v(t), explicit));
```

TIP: Zie ook in paragraaf *Symbol Mode* (page 4) voor de bediening van de *Editor* waar studenten nog wel degelijk wat instructie voor moeten krijgen.

De oplossing van een DV graden

Voor de Grading Code van de oplossing van een differentiaalvergelijking kunt u de oplossing van de student in de gegeven DV invullen en eisen dat het klopt zoals in het volgende voorbeeld.

möbius

Preview

Solve the following differential equation using any method.
 You can use methods 1 and 2 from the reading material or any other methods that

NOTE: In all answers, use C to denote an arbitrary constant that will be in your final answer.
 Express your final answer in a form that has just one arbitrary constant C .

What is the solution $y(x)$ to the following differential equation?

$$\frac{dy}{dx} - yx^2 + y(x^3 + a) = 0$$

Solution is $y(x) =$

Figure 1.122: Het graden van de oplossing van een DV

Het *algorithm* is als volgt:

```
$DEb:=maple("diff(y(x),x)-y(x)*x^2+y(x)*(x^3+a)=0");
$oplbdisplay1:=maple("printf(MathML[ExportPresentation](y(x)=exp(x^3/3-x^4/4-a*x+C)))");
$oplbdisplay2:=maple("printf(MathML[ExportPresentation](y(x)=C*exp(x^3/3-x^4/4-a*x)))");
$DEbdisplay="\(\frac{\text{d}y}{\text{d}x}-yx^2+y(x^3+a)=0\)";
$antw:=maple("exp(x^3/3-x^4/4-a*x+C)");
```

Variable	Value
DEb	$\text{diff}(y(x),x)-y(x)*x^2+y(x)*(x^3+a) = 0$
oplbdisplay1	$y(x) = e^{\frac{1}{3}x^3 - \frac{1}{4}x^4 - ax} C$
oplbdisplay2	$y(x) = C e^{\frac{1}{3}x^3 - \frac{1}{4}x^4 - ax}$
DEbdisplay	$\frac{dy}{dx} - yx^2 + y(x^3 + a) = 0$
antw	$\exp(1/3*x^3-1/4*x^4-a*x+C)$

Figure 1.123: De oplossing van de DV

Ten behoeve van de feedback zijn twee oplossingen gedefinieerd in mathml-formaat.

De differentiaalvergelijking die in de opgave verschijnt is gemaakt met LaTeX.

De oorspronkelijke differentiaalvergelijking \$DEb is in maple-syntax gedefinieerd en meteen op 0 herleid om mee te rekenen.

Daarbij moet dan beslist opgegeven worden dat y een functie is van x met y(x). Deze DV kunt u dan in de Grading Code gebruiken om het antwoord van de student in te vullen.

Als *Grading Code* neemt u de volgende code:

```
evalb(simplify(eval(lhs($DEb), y(x)=$RESPONSE))=0)
and
evalb(StringTools[CountCharacterOccurrences]("$RESPONSE", "C")=1);
```

Het betekent dat u het linkerlid van de oorspronkelijke DV neemt: lhs(\$DEb) en dat evalueert u met ingevuld dat y(x) gelijk is aan de respons van de student.

Daar moet dan 0 uitkomen als het goed is. Ook moet er een C in het algemene antwoord voorkomen. Daar wordt op gecontroleerd met behulp van StringTools.

Via de *Custom Preview* kunt u ook nog een waarschuwing geven dat de student de integratieconstante C niet vergeet:

```
if evalb(StringTools[CountCharacterOccurrences]("$RESPONSE", "C")=0) then
  printf(" Use a capital C as an arbitrary constant in the solution. ") else
  printf(MathML[ExportPresentation]($RESPONSE)) end if;
```

Hogere afgeleiden

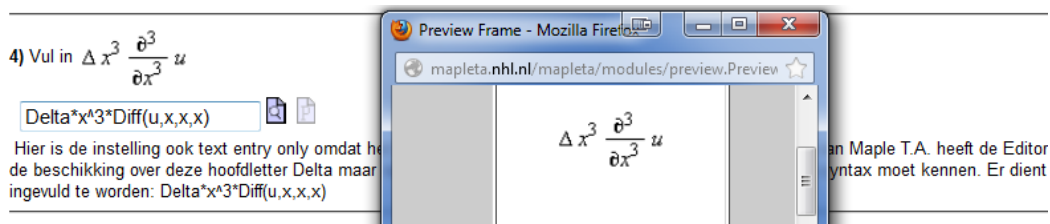


Figure 1.124: Hogere afgeleiden met het tekstveld

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 enigzins getraind is in het hanteren van de Editor of op de hoogte is van de syntax.

9) Vul in $\frac{\partial^3}{\partial x^3} u(x, y)$

The screenshot shows the Maple Equation Editor interface. The title bar says "Equation Editor". The input field contains the expression $\frac{d}{dx} \left(\frac{d}{dx} \left(\frac{d}{dx} u(x, y) \right) \right)$. The variables x and y are highlighted in blue.

Figure 1.125: 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^2/12*u[yyyy]");
```

In Maple is subscript dus met vierkante haken.

6a) Vul in $(1 - M^2) \alpha x u_{xxx} - \frac{1}{12} \alpha y^2 u_{yyyy}$

The screenshot shows the Maple Equation Editor interface. The title bar says "Equation Editor" with a "Help" link. The input field contains the expression $(1 - M) \alpha x u_{xxx} - \frac{1}{12} \alpha y^2 u_{yyyy}$. The variables x and y are highlighted in blue.



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 1.126: Subscript in de Editor

De grading code gaat dan niet met het matchen van de oplossingen van de DV, maar de formule kan wel op de normale manier gematcht worden als volgt:

```
evalb(($antw6)=$RESPONSE);
```

1.4.20 Integralen

1.4.20.1 Onbepaalde integraal

Bij het herleiden van de onbepaalde integraal vraagt u in feite de primitieve van de integrand aan de student.

De integratieconstante zet u er vast zelf achter zodat de student dat niet hoeft te doen

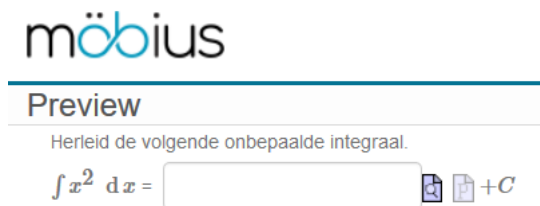


Figure 1.127: Primitiveren, de onbepaalde integraal

In het *Algorithm* hebt u bijvoorbeeld geprogrammeerd:

```
$integraal=maple("Int(x^2,x)");
$intdisplay=maple("printf(MathML[ExportPresentation]
($integraal))");
$antw=maple("value($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((diff($antw,x)-diff($RESPONSE,x))=0)` and `evalb(StringTools[Search]("int","$RESPONSE")=0);` Met deze Grading Code zorgt u er ook voor dat de student niet het maple-commando gebruikt.

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 moet 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 en als de student toch nog een constante eraan toevoegt, maakt dat niet uit.

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).

1.4.20.2 Bepaalde integraal en verschillende schrijfwijzen

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);`

Het gaat er hier even om hoe de integraal netjes op het scherm komt.



Preview

Bereken de volgende bepaalde integraal met pen en papier.

$$\int_1^2 \left(3 a x^5 + \frac{1}{5 x^3} \right) dx$$

Je kunt het antwoord intikken zoals je dat ook op de rekenmachine doet, dus met schuine streep.

Dan heb je geen extra rekenmachine nodig voor deze opgave.

$$\int_1^2 3 a x^5 + \frac{1}{5 x^3} dx =$$




Figure 1.128: De bepaalde integraal berekenen

In het *Algorithm* programmeert u het volgende met daaronder de figuur met de waarden van de variabelen:

```
$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))
");
$displayvraagtest=maple("printf(MathML[ExportPresentation]
($vraag))");
$antwoord=maple("simplify(value($vraag))");
$displayantwoord=maple("printf(MathML[ExportPresentation]
($antwoord))");
$displayvraag0="\(\ \int_{1}^{2} \$a\,a\,x^{\$b}+\frac{1}{\$b\,x^{\$a}}
\ \mathrm{d}x\)"
";
$displayvraag1="\(\ \displaystyle{\int_{1}^{2} \$a\,a\,x^{\$b}+
\frac{1}{\$b\,x^{\$a}} \ \mathrm{d}x}\)"
";
$displayvraag2="\(\ \displaystyle{\int\limits_{1}^{2} \$a\,a
\,x^{\$b}+\frac{1}{\$b\,x^{\$a}} \ \mathrm{d}x}\)"
";
```

Variable	Value
a	7
b	3
integrand	$7a^*x^3+1/3/x^7$
vraag	$\text{Int}(7a^*x^3+1/3/x^7, x = 1 \dots 2)$
displayvraag	$\int_1^2 \left(7 a x^3 + \frac{1}{3 x^7} \right) dx$
displayvraagtest	$\int_1^2 \left(7 a x^3 + \frac{1}{3 x^7} \right) dx$
antwoord	$7/128+105/4*a$
displayantwoord	$\frac{7}{128} + \frac{105}{4} a$
displayvraag0	$\int_1^2 7 a x^3 + \frac{1}{3 x^7} dx$
displayvraag1	$\int_1^2 7 a x^3 + \frac{1}{3 x^7} dx$
displayvraag2	$\int_1^2 7 a x^3 + \frac{1}{3 x^7} dx$

Figure 1.129: Verschillende schrijfwijzen van de bepaalde integraal

TIP: De integraal zoals die op het scherm verschijnt in de vraagtekst, kan op verschillende manieren gegenereerd worden.

Het eenvoudigst is om dat met maple te doen. Let eens op de spaties bij de quotes in het algoritme bij de variabele \$displayvraag.

Als u de quotes stijf tegen het commando plaatst is het integraalteken kleiner en de grenzen staan niet mooi boven en onder het integraalteken.

Dus let op de spaties bij de quotes.

Een nadeel van het genereren van de mathml-code door maple is dat er haken komen te staan rond de integrand en dan is niet altijd nodig of wenselijk.

Verder is het nadeel ook dat er een spatie tussen d en x staat waar die er niet moet staan in feite want de d is een operator.

TIP: Mooier is het om de bepaalde integraal op het scherm te krijgen met behulp van LaTeX. U kunt dan kiezen tussen wel of geen haken om de integrand.

Ook kunt u ervoor zorgen dat er geen spatie komt tussen de d en de x. Verder is de grootte van het integraalteken nog te sturen en de grenzen er mooi boven en onder te krijgen. Er zijn met \$displayvraag0 t/m 2 een aantal mogelijkheden aangedragen.

Een nadeel is dat als de variabelen de waarde 1 hebben, dat u dan ook een 1 ziet staan in de formule. Beter is het om dan de 1 als variabele niet mee te nemen in de range voor de randomisering van de integrand. Ook bij negatieve variabelen wordt het lastig om te werken met LaTeX omdat er dan wellicht +- achter elkaar te zien is.

Variable	Value
a	9
b	1
integrand	$9*a*x+1/x^9$
vraag	$\text{Int}(9*a*x+1/x^9, x = 1 \dots 2)$
displayvraag	$\int_1^2 \left(9 a x + \frac{1}{x^9} \right) dx$
displayvraagtest	$\int_1^2 \left(9 a x + \frac{1}{x^9} \right) dx$
antwoord	$255/2048+27/2*a$
displayantwoord	$\frac{255}{2048} + \frac{27}{2} a$
displayvraag0	$\int_1^2 9 a x^1 + \frac{1}{1 x^9} dx$
displayvraag1	$\int_1^2 9 a x^1 + \frac{1}{1 x^9} dx$
displayvraag2	$\int_1^2 9 a x^1 + \frac{1}{1 x^9} dx$

Figure 1.130: LaTeX en de integrand met variabele 1

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.

1.4.20.3 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.

Preview

Bereken de dubbelintegraal van de functie $f(x, y) = x e^{(xy)}$ over het gebied G .

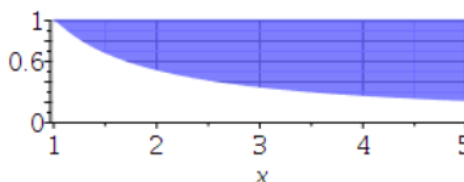
Het gebied G is het gebied in het x, y -vlak tussen de kromme $y = \frac{1}{x}$ en de lijnen $y = 1$ en $x = 5$.

TIP: teken voor jezelf het integratiegebied G .

$$\iint_G f(x, y) \, dA = \text{[input field]} \quad \times \text{ (exact antwoord)}$$

Je antwoord was niet correct.

Het integratiegebied is te zien in de figuur



Bereken dus $\int_1^5 \int_{\frac{1}{x}}^1 x e^{(xy)} \, dy \, dx = \text{[input field]}$

Figure 1.131: Dubbelintegraal berekenen

De eerste integraal is met LaTeX voorbereid in het *Algorithm*:

```
$formule="\(\displaystyle{\iint\limits_{G}f(x,y)\, \mathrm{d}A})\);
```

Het juiste antwoord is ook in het Algorithm voorbereid met variabele functie en variabele grenzen:

```
$Int=maple("Int(Int($f,y=$l1..$u1),x=$l2..$u2)");
$antw=maple("value($Int)");
```

Voor de *Grading Code* hebt u dan alleen nodig:

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

Of misschien met vereenvoudiging:

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

TIP: Voor de feedback-in-stappen kunt u tussenantwoorden ontwerpen door eenvoudig de binnenste integraal te veranderen in `int` met een kleine letter i.

```
$Inthalf=maple("Int(int($f,y=$l1..$u1),x=$l2..$u2)");
$Inthalfdisplay=maple(" printf(MathML[ExportPresentation]
($Inthalf)) ");
```

1.4.20.4 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.

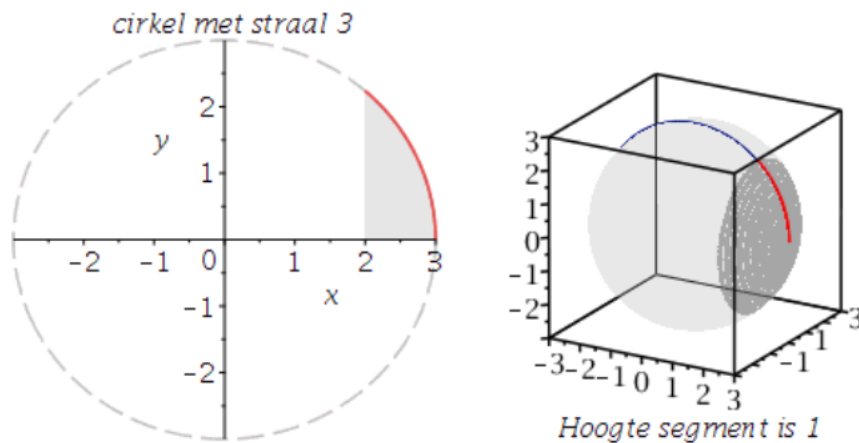
De integraal invoeren in de editor

möbius

Preview

Wentel een deel van de cirkel met straal $R = 3$ zoals in de figuur om de x -as.

J krijgt dan een bolsegment en de hoogte van het bolsegment is $h = 1$.



Stel de integraal op voor het volume het bolsegment zonder het uit te rekenen.

Het volume van het segment van de bol is de integraal

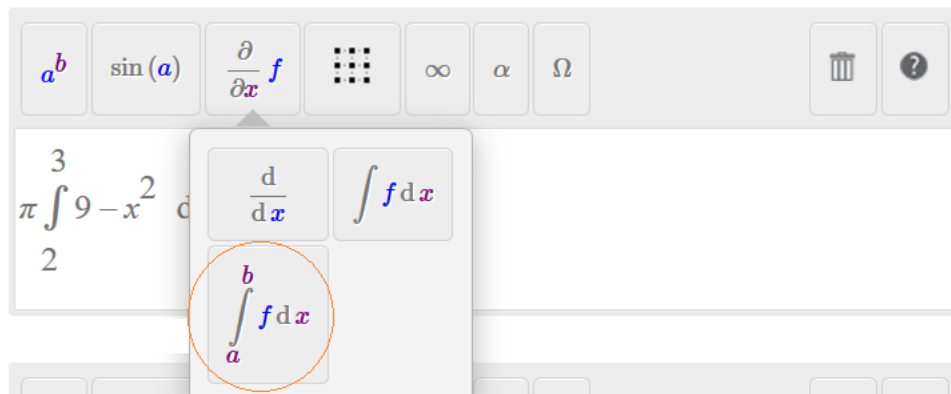


Figure 1.132: Student voert een integraal in

Het bijbehorende algoritme is

```
$kleur=switch(rint(8),"red","yellow","khaki","gray","green","magenta","cyan","turquoise");
$R=range(3,5);
$r=range(1,2);
$h=$R-$r;
$f=maple("sqrt($R^2-x^2)");
$fdisplay="\ (y=\sqrt{$R^2-x^2})\ ";
$volume=maple("simplify(Student[Calculus1][VolumeOfRevolution]($f,x=$r..
$R,output=integral))");
$voledisplay=maple(" printf(MathML[ExportPresentation]($volume)) ");
$antw=maple("value($volume)");
$antw1=maple("subs(Pi=pi,$antw)");
```

```

$antwdisplay=maple("printf(MathML[ExportPresentation]($antw))");
$vb=maple("printf(MathML[ExportPresentation](Int(f,x=a..b)) )");
$plot=plotmaple("plots[display]([plots[implicitplot](x^2+y^2=$R^2,x=-$R..$R,y=-
$R..$R,color=grey,linestyle=dash,title=`cirkel met straal $R`),
plot(sqrt($R^2-x^2),x=$R..$R,color=red),
plot(sqrt($R^2-x^2),x=$R..$R,color=$kleur,filled=true,transparency=0.6)]),
plotoptions='height=250, width=250'");
$plotbolkapje=plotmaple("plots[display]([Student[Calculus1]
[VolumeOfRevolution](sqrt($R^2-x^2),x=-$R..
$R,output=plot,volumeoptions=[color=grey,transparency=0.9],caption=`Hoogte segment
is $h`,axes=boxed),
Student[Calculus1][VolumeOfRevolution](sqrt($R^2-
x^2),x=$R..$R,output=plot,volumeoptions=[color=
$kleur,transparency=0.5],functionoptions=[thickness=3,color=red],title=`bol met
straal $R`,caption=``,axes=boxed)],orientation=[-60,70]),
plotoptions='height=250, width=250'");
$circeldisplay=maple("printf(MathML[ExportPresentation](x^2+y^2=R^2) )");
$ydisplay=maple("printf(MathML[ExportPresentation](y=sqrt(R^2-x^2)) )");
$dvddisplay="\(\pi\,y^2\,\text{d}x\)");

```

Als de waarde van de correcte integraal in het *Algorithm* geprogrammeerd is als \$antw, kunt u gebruikmaken van de volgende *Grading Code*:

```

evalb(simplify($antw-(value($RESPONSE)))=0) and
evalb(StringTools[Search]("nt","$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 in de string vertaald naar het maple-commando "int". In de string van de respons staat dan dus "int".

De letter π wordt door de editor altijd juist vertaald in Pi.

De integraal invoeren in een textveld

Dat hier niet naar "int" gezocht wordt in de string van de response maar naar "nt" heeft ermee te maken dat deze zelfde Grading Code dan ook bruikbaar is in de situatie van *Text entry only*. De student kan daar namelijk Int of int tikken. Beide zal worden goedgekeurd met dezelfde Grading Code.

De letter π zou de student ook als pi of als Pi kunnen tikken met de volgende Grading Code:

```

evalb(simplify(subs(pi=Pi,$antw-(value($RESPONSE))))=0) and
evalb(StringTools[Search]("nt","$RESPONSE")>0);

```

met een textveld en het onderzoeken van de respons

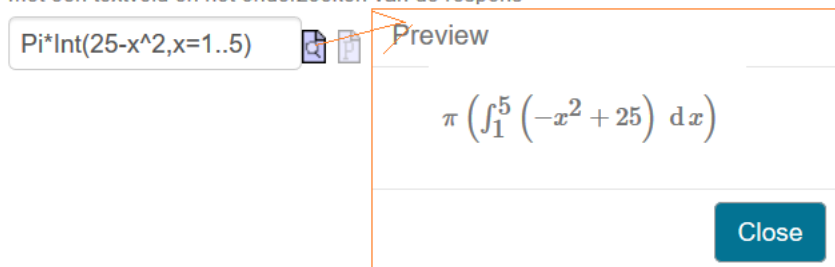


Figure 1.133: Integraal invoeren in textveld

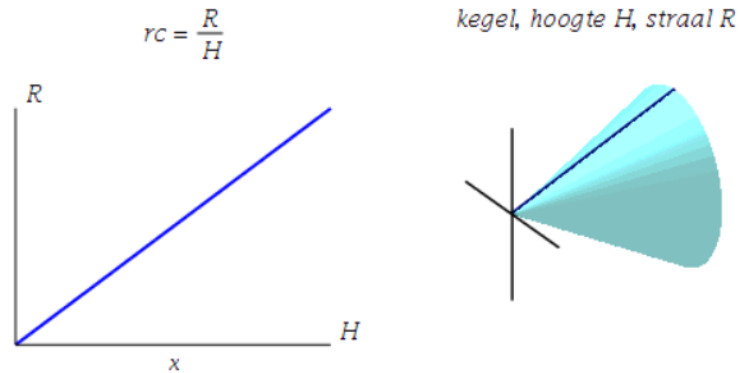
Invoeren van een integraal met parameters

Voor het geval de integrand en/of de grenzen parameters bevatten is het niet altijd gezegd dat u de ingevoerde integraal van de student kunt matchen met de correcte integraal door middel van het uitrekenen van de waarde van de integraal.

In het volgende voorbeeld ziet u hoe te handelen.

Het lijnstuk met richtingscoëfficiënt $rc = \frac{R}{H}$ wentel je om de x -as.

Je krijgt dan een kegel met hoogte H en straal van het grondvlak R zoals in de figuur.



Geef de integraal om het volume te berekenen van deze kegel zonder deze uit te rekenen.

Het volume van de kegel is de integraal

Figure 1.134: Inhoud kegel met integreren

Het algoritme is als volgt:

```
$kleur=switch(rint(8),"red","yellow","khaki","gray","green","magenta","cyan","turquoise");
$R=range(2,5);
$H=$R+range(1,5);
$f=maple("$R/$H*x");
$F=maple("R/H*x");
$volume=maple("Student[Calculus1][VolumeOfRevolution]
($F,x=0..H,output=integral)");
$voledisplay=maple(" printf(MathML[ExportPresentation]($volume)) ");
$antw=maple("value($volume)");
$antw1=maple("subs(Pi=pi,$antw)");
$antwdisplay=maple("printf(MathML[ExportPresentation]($antw))");
$vb=maple("printf(MathML[ExportPresentation](Int(f,x=a..b)) )");
$plot=plotmaple("plots[display]({plot($f,x=0..
$H,color=blue,thickness=2,tickmarks=[0,0],title=typeset(rc=R/
H),scaling=constrained),plots[textplot]([[$H,0,H],[0,
$R,R]],align={right,above}}),plotoptions='height=250, width=250'");
$plotkegel=plotmaple("Student[Calculus1][VolumeOfRevolution]
($f,x=0..$H,output=plot,tickmarks=[0,0,0],volumeoptions=[color=
$kleur,transparency=0.5],functionoptions=[thickness=3],title='kegel, hoogte H,
straal R',caption = ``),plotoptions='height=250, width=250'");
```

```
$dVdisplay="\(\pi\,y^2\,\text{d}x\)";
```

De *Grading Code* is nu als volgt, omdat het programma moet weten dat de parameters positief zijn om de integraal te kunnen berekenen en de vereenvoudiging te kunnen doen zodat er 0 uitkomt. (Iets dergelijks kunt u ook gebruiken voor bijvoorbeeld wortels en logaritmen).

```
assume(R>0): assume(H>0): evalb(simplify($antw-
(value($RESPONSE))=0) and evalb(StringTools[Search]
("nt", "$RESPONSE")>0);
```

De analyse van de respons

In de volgende figuren ziet u de invoer van de student links, vervolgens in de middelste kolom de respons en in de rechter kolom de instelling symbol of text en of er gekeken wordt naar de respons of naar de string van de respons van de student in verschillende situaties

Figure 1.135 shows the analysis of a student's answer in Symbol mode. The student input is an integral from 1 to 10 of $x^2 dx$. The response is `"int(x^2,x = 1 .. 10)"`. The mode is *Symbol entry only*.

Figure 1.135: 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 die door maple meteen berekend wordt 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 (met int) en dat is ook te controleren met behulp van StringTools en verder wordt de *waarde* van de integraal weer gebruikt om mee te matchen in de *Grading Code*.

Figure 1.136 shows the analysis of a student's answer in Text mode. The student input is `int(x^2,x=1..10)`. The response is `"int(x^2,x=1..10)"`. The mode is *Text entry only*. The second row shows the response as `333`. The third row shows the response as `Int(x^2,x=1..10)`.

Figure 1.136: De analyse van het antwoord van de student in de vorm van een integraal in Text mode

In deze figuur ziet u 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. Bij het tweede invulveld wordt de pure response (\$RESPONSE) direct vertaald naar de *waarde* van de integraal door maple berekend.

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 tekstveld, de integraal zelf weergegeven en niet de waarde ervan.

TIP: Als u van uw studenten eist dat ze "Int" met een hoofdletter hanteren in hun invoer in het tekstveld, 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));
```

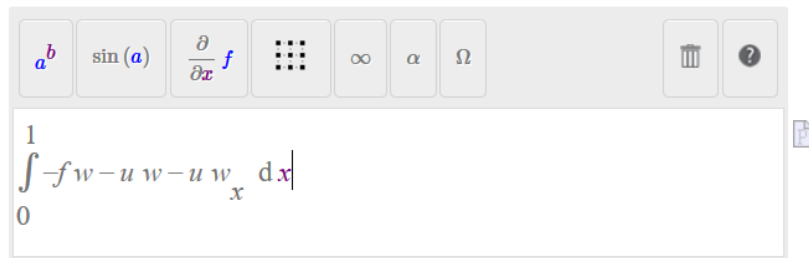
$$x^2 \quad (1.79)$$

```
> op(2,Int(x^2,x=0..10));
```

$$x = 0..10 \quad (1.80)$$

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 $\int_0^1 (-f w - u w - u w_x) dx$



Hier is de grading wat lastig omdat de integraal symbolisch is. Immers zoals hij hier staat, kan hij niet uitgerekend worden en dan kan de uitkomst ervan ook niet gematcht worden met het antwoord van de student. Daarom worden hier in de grading afzonderlijk de integrand en de grenzen van de variabele gematcht. De editor kan hier wel gebruikt worden.

Figure 1.137: Invoeren van een integraal symbolisch

Het juiste antwoord programmeert u in het *Algorithm* met

```
$antw7:=maple("Int(-w[x]*u-w*u-w*f,x=0..1)");
$displayantw7:=maple(" printf(MathML[ExportPresentation]($antw7))
");
```

Operanden van een integraal

Hier 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*f,x=0..1);
```

$$-wf - 2wu \quad (1.81)$$

```
> int(-w[x]*u-w*f,x=0..1);
```

$$\int_0^1 (-wf - wu - w_x u) dx \quad (1.82)$$

```
> op(int(-w[x]*u-w*f,x=0..1));
```

$$-wf - wu - w_x u, x = 0..1 \quad (1.83)$$

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).

Er zijn dus twee mogelijkheden voor het matchen van integralen. U vergelijkt de waarde van de integralen als deze uitgerekend kunnen worden óf u vergelijkt de operanden van de integralen.

In het voorbeeld (*Figure 1.137 (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.

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 (met vierkante haken), 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]("nt", "$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.

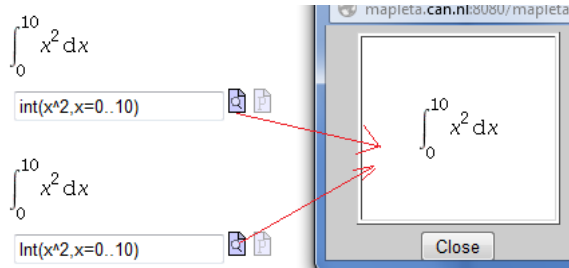


Figure 1.138: Verschil in invoer met Int en int en geen verschil in de Preview

1.4.21 Matrices en Vectoren

1.4.21.1 Inleiding

Bij Matrices en Vectoren is de *Grading Code* niet voor de hand liggend. U moet hiervoor iets meer weten. Eenvoudig gelijkstellen gaat hier niet op.

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 afzonderlijke elementen van een matrix (of vector) bepaalde waarden hebben zoals met de volgende *Grading Code*:

```
is ($RESPONSE[2,1]=0) and is($RESPONSE[3,1]=0);
```

Hier wordt gecheckt 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 geïnstrueerd worden met name bij het kiezen van de juiste dimensies van de Matrix. In het veld van de Equation Editor is een menu te zien waarin de student bijvoorbeeld een vierkante 2×2-matrix kiest. Er komt dan een matrix te staan waarin de getallen aangepast moeten worden. Zie ook onderstaande figuur.

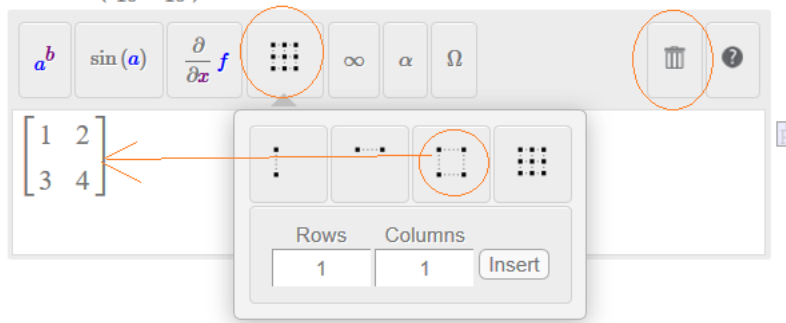


Figure 1.139: Het invoeren van een Matrix door de student in de Editor

TIP: Willen studenten veel verbeteren in de ingevoerde formule, laat ze dan met het prullenmandje in het menu alles opschonen en even opnieuw beginnen.

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 of een rijvector 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! Evenzo geldt dat voor een rijvector.

TIP: Omdat u bij vragen met matrices en vectoren in feite altijd de Editor zult gebruiken, is het handig om het correcte antwoord (bij Answer) altijd met mathml te programmeren, zodat de student in de feedback bij het correcte antwoord ook altijd de tweedimensionale presentatie ziet: `printf(MathML[ExportPresentation]($antw));`

1.4.21.2 De matrix en de vector definiëren

In het algoritme kunt u een matrix en een vector alvast gaan definiëren en ermee rekenen.

```
$index1=range(0,3);
$index2=range(0,3);
$a=range(0,9);
$b=switch($index1,p,q,r,s);
$c=range(0,9);
$d=range(0,9);
$e=switch($index2,p,q,r,s);
$f=range(0,9);
$g=range(0,9);
$h=range(0,9);
$A=maple("Matrix([[ $a, $b ], [ $c, $d ]])");
$B=maple("Matrix([[ $e, $f ], [ $g, $h ]])");
$Adisplay=maple("printf(MathML:-ExportPresentation($A))");
$Bdisplay=maple("printf(MathML:-ExportPresentation($B))");
$antw=maple("$A . $B");
$antwoord = maple("printf(MathML:-ExportPresentation($antw))");
```

Variable	Value
index1	3
index2	1
a	2
b	s
c	4
d	7
e	q
f	0
g	3
h	1
A	Matrix(2,2,{{(1, 1) = 2, (1, 2) = s, (2, 1) = 4, (2, 2) = 7},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
B	Matrix(2,2,{{(1, 1) = q, (2, 1) = 3, (2, 2) = 1},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
Adisplay	$\begin{pmatrix} 2 & s \\ 4 & 7 \end{pmatrix}$
Bdisplay	$\begin{pmatrix} q & 0 \\ 3 & 1 \end{pmatrix}$
antw	Matrix(2,2,{{(1, 1) = 2*q+3*s, (1, 2) = s, (2, 1) = 4*q+21, (2, 2) = 7},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
antwoord	$\begin{pmatrix} 2q + 3s & s \\ 4q + 21 & 7 \end{pmatrix}$

Figure 1.140: De matrix in het algorithm

In bovenstaand Algorithm ziet u hoe de matrix A opgeslagen wordt. Hiermee kunt u binnen maple gewoon rekenen met de bekende commando's voor matrices. Voor de leesbaarheid op het scherm schrijft u de matrix weg met mathml-code die u genereert met Maple.

Een vector in het Algorithm is ook gemakkelijk te definiëren.

```

$a=range(2,5);
$b=range(2,5);
$a1=range(-10,10);
$a2=range(-10,10);
$b1=range(-10,10);
$b2=range(-10,10);
$detsabs=maple("abs($a1*($b2)-($b1)*($a2))");
condition:gt($detsabs,0);
$A=maple("Vector([$a1,$a2])");
$Adisplay=maple("printf(MathML[ExportPresentation]($A))");
$B=maple("<$b1,$b2>");
$Bdisplay=maple("printf(MathML[ExportPresentation]($B))");
$AplusB=maple("$a*$A+$b*$B");
$AplusBdisplay=maple("printf(MathML[ExportPresentation]
($AplusB))");

```

Variable	Value
a	3
b	4
a1	2
a2	9
b1	-5
b2	-10
detabs	25
A	Vector[column](2,{1 = 2, 2 = 9},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
Adisplay	$\begin{pmatrix} 2 \\ 9 \end{pmatrix}$
B	Vector[column](2,{1 = -5, 2 = -10},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
Bdisplay	$\begin{pmatrix} -5 \\ -10 \end{pmatrix}$
AplusB	Vector[column](2,{1 = -14, 2 = -13},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
AplusBdisplay	$\begin{pmatrix} -14 \\ -13 \end{pmatrix}$

Figure 1.141: De definitie van een vector in het algoritme

In bovenstaand algoritme ziet u twee verschillende manieren om een (kolom)vector te definiëren: \$A en \$B. Het komt op precies hetzelfde neer en de manier van \$B is minder tikwerk.

1.4.21.3 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.

U doet dat door in de rubriek Answer van het Maple-graded antwoordveld \$REPSONSE in te voeren (of "\$RESPONSE") als u als feedback wilt zien wat u ingevoerd hebt.

Maple-graded:

Weighting	1
Answer	\$RESPONSE
	(referenced when grading as \$ANSWER)
Grading Code:	evalb((\$antw)-(\$RESPONSE)=0);

Figure 1.142: De respons als correcte antwoord bekijken

In de Preview van de vraag vult u iets in en u klikt op Grade waarna u ziet wat de vertaling is van wat u ingevoerd hebt.

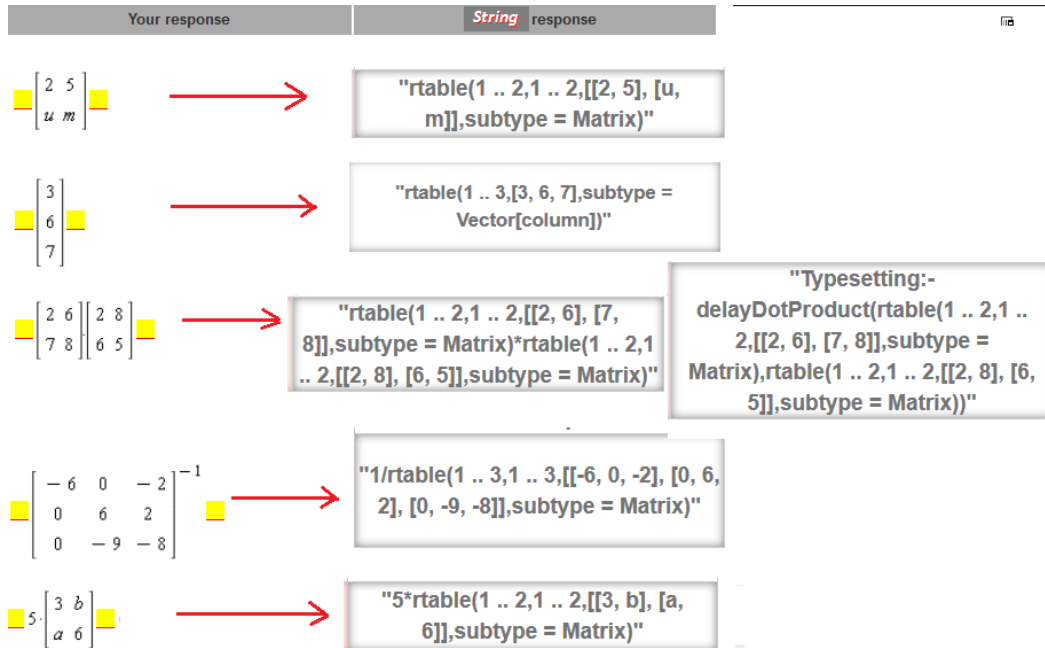


Figure 1.143: Invoer van matrices vertaald naar de string

In de figuur hierboven is te zien hoe de invoer naar een string wordt vertaald.

- Hoe een 2×2 -matrix wordt vertaald.
- Daaronder een matrix met één kolom ingevoerd wordt vertaald naar een `Vector[column]`.
- Als de student twee matrices invoert met een ster ertussen dan wordt dat niet gezien als een matrixvermenigvuldiging. De vertaling van het studentantwoord is eenvoudig iets met `"rtable() * rtable()"`. Het zal niet goedgekeurd worden.
Met een (gewone) punt ertussen, dan worden deze twee matrices opgevat als het inwendig product (`DotProduct`) van de twee matrices.
U kunt in dat geval bijvoorbeeld met `StringTools` controleren of `DotProduct` ook voorkomt in de string van de respons.
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 met `LinearAlgebra`: -
`Equal( simplify( $RESPONSE ), simplify($antw))`
`and evalb(0=StringTools[Search]("DotProduct", "$RESPONSE"));`
- Bij de inverse heeft de student de matrix ingetikt tot de macht -1 . Dit zou goedgerekend worden als u niet met `StringTools` uw maatregelen neemt met de volgende *Grading Code*:
`LinearAlgebra:-Equal( simplify( $RESPONSE ), simplify($antw))`
`and evalb(0=StringTools[Search]("/rtable", "$RESPONSE"));`
- Als de student de factor 5 vóór de matrix moet halen, dan kunt u met deze vermenigvuldiging controleren of er een ster in de string van de respons voorkomt. De student tikt een ster of een spatie of niets. Het systeem vertaalt dat altijd met een ster in de *Symbol mode*.
`LinearAlgebra:-Equal( simplify( $RESPONSE ), simplify($antw))`
`and evalb(0=StringTools[Search]("*", "$RESPONSE"));`

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
$\begin{bmatrix} 2 & 6 \\ 5 & 4 \end{bmatrix} \begin{bmatrix} 2 & 7 \\ 4 & 3 \end{bmatrix}$	Matrix(2,2,{(1, 1) = 28, (1, 2) = 32, (2, 1) = 26, (2, 2) = 47},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
$\begin{bmatrix} 3 & 8 \\ 7 & 9 \end{bmatrix}^{-1}$	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 = [])
$5 \cdot \begin{bmatrix} a & k \\ 7 & 8 \end{bmatrix}$	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 = [])

Figure 1.144: Invoer van matrices

In bovenstaande figuur is het volgende te zien

- De student voert een matrixvermenigvuldiging in (met behulp van de gewone punt tussen de twee matrices, dus niet met ster want dan komt in de feedback een error.) 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 `"/rtable...."` niet voorkomt met de *Grading Code*:

```
LinearAlgebra[Equal](simplify($RESPONSE),$antw) and
evalb(StringTools[Search]("/
rtable", "$RESPONSE")=0);
```

- De scalaire vermenigvuldiging die de student invoert met behulp van een ster (de student ziet een middot in de Editor) wordt door het systeem vertaald naar de resulterende matrix waar alle elementen met deze scalair 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.

1.4.21.4 Voorbeeld

In het volgende voorbeeld moet er een 3×4 -matrix ingevoerd worden door de student.

Preview

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

$$\begin{aligned} 15x - 5y + 3z &= 4 \\ -14x - 6z - 9y &= 3 \\ 2x - 6y + 2z &= 9 \end{aligned}$$

Geef de aangevulde 3 bij 4 matrix van dit lineaire systeem ten opzichte van de onbekenden $[x, y, z]$. Eventueel rijen verwisselen of rijen vereenvoudigen mag.

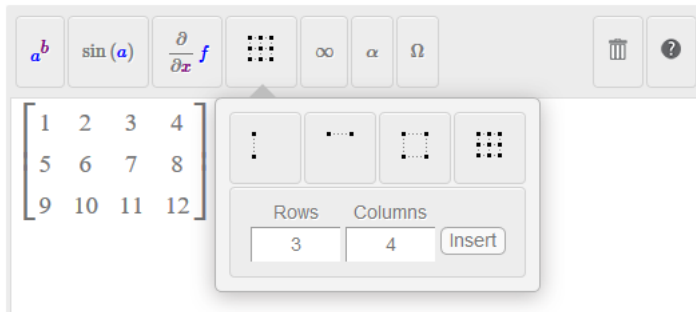


Figure 1.145: Matrix invoeren

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 *Algorithm* wordt eerst een aantal coëfficiënten voorbereid ten behoeve van de coëfficiënten van de lineaire 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. In de figuur hier onder ziet u hoe deze in code wordt opgeslagen in het *Algorithm*. Dit is dus niet iets om naar de student te communiceren.

Daarom bereid u het correcte antwoord bij Answer ook voor met de code `printf(MathML:-ExportPresentation($M));`

De variabelen \$displayantw en \$displaytest zijn ter controle waarbij de variabele \$test gebruikt wordt in de *Grading Code*.

e1	$-4y + 13x + 14z = -2$
E1	$-4y + 13x + 14z = -2$
e2	$-8y + 7z - 8x = -2$
E2	$-8y + 7z - 8x = -2$
e3	$y + 5x - 6z = -4$
E3	$y + 5x - 6z = -4$
M	Matrix(3,4,{{(1, 1) = 13, (1, 2) = -4, (1, 3) = 14, (1, 4) = -2, (2, 1) = -8, (2, 2) = -8, (2, 3) = 7, (2, 4) = -2, (3, 1) = 5, (3, 2) = 1, (3, 3) = -6, (3, 4) = -4},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
displayantw	$\begin{pmatrix} 13 & -4 & 14 & -2 \\ -8 & -8 & 7 & -2 \\ 5 & 1 & -6 & -4 \end{pmatrix}$
test	Matrix(3,4,{{(1, 1) = 1, (1, 4) = -398/1033, (2, 2) = 1, (2, 4) = 1134/1033, (3, 3) = 1, (3, 4) = 546/1033},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
displaytest	$\begin{pmatrix} 1 & 0 & 0 & -\frac{398}{1033} \\ 0 & 1 & 0 & \frac{1134}{1033} \\ 0 & 0 & 1 & \frac{546}{1033} \end{pmatrix}$

Figure 1.146: Echelon matrix

In de *Grading Code* programmeert u als volgt:

```

LinearAlgebra[Equal]($test, LinearAlgebra[ReducedRowEchelonForm]
($RESPONSE));

```

Deze gereduceerde vorm van de aangevulde matrix is eenduidig en 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.

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 in de Echelonvorm gegoten en vervolgens gematcht met het afgeronde antwoord van het correcte antwoordmodel en ook in de echelonvorm gegoten.

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.

1.4.21.5 Random matrix

Om te beginnen programmeert u een random Matrix. Dat kan ook door alle coëfficiënten stuk voor stuk als range te definiëren, maar de volgende methode 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)) ");
```

1.4.21.6 Vectoren

Vectoren met elkaar matchen doet u op dezelfde manier als bij matrices. Zorg er wel voor dat het correcte antwoord wel een vector is.

De Grading Code die daarvoor nodig is: `LinearAlgebra[Equal]($RESPONSE,$antw);`

- Om een **richtingsvector** te overhoren is het niet handig om kentallen van de vector van de student \$RESPONSE en van het correcte antwoord \$antw met elkaar te matchen. Een richtingsvector kan echter langer of korter zijn en moet dan ook goed gerekend worden. Gaat het om een tweedimensionale richtingsvector dan kunt u bijvoorbeeld in de grading code het volgende nemen:
`evalb(LinearAlgebra[Determinant](Matrix([$RESPONSE,$antw]))=0)
and evalb(signum($antw[1])=signum($RESPONSE[1]));`
 In bovenstaande *Grading Code* kijkt u naar de 2×2-matrix gevormd door de beide te matchen vectoren. Is het stelsel afhankelijk, dan is de determinant van de matrix gelijk aan 0. Als ook nog belangrijk is in welke richting zoals bij gradient en dergelijke, kunt u het teken checken van het eerste kental dat bij beide vectoren overeen moet komen en dat doet u met signum. (Of van andere kentallen als die niet 0 moeten zijn.)
- Een richtingsvector in hogere dimensies is lastiger te matchen. U maakt dan weer een matrix van de beide vectoren en u eist dat de rang gelijk is aan 1. Dat is alles:
`evalb(LinearAlgebra[Rank](Matrix([$RESPONSE,$n]))=1);`
 Eventueel de richting met signum van een van de kentallen.

1.5 Vraagtype Mathematical Formula

1.5.1 Inleiding

Het vraagtype *Mathematical Formula* biedt een aantal mogelijkheden om extra dingen met formules te doen.

Bij *Maple-graded* vragen hebt u een Grading Code nodig. Bij het vraagtype *Mathematical Formula* volstaat het correcte antwoord.

Voor gewone formules is dit vraagtype vaak ontoereikend

TIP: Het is met dit vraagtype 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.

- Een mogelijkheid voor een formule is $(-17.78 \pm 0.01) + (0.56 \pm 0.01) * x$. 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). Het nadeel is dat u geen formules met lettercombinaties kunt gebruiken en studenten mogen de sterren voor vermenigvuldigen weglaten.
- Een getal overhoren waarbij het goede antwoord \$antw in het *Algorithm* voorbereid is tikt u bij de instellingen voor het correcte antwoord van het antwoordveld: \$antw $\pm$ 0.1 . Het betekent dus een tolerantie van 0.1 in het antwoord.
- Het aardige is dat voor de tolerantie met het vraagteken eventueel ook een *algoritmische variabele* gebruikt kan worden!

← Insert Response Area

Question Feedback

Mathematical Formula:

Weighting 1

Sub-type: Formula - e.g. $e^x \sin(x^2)$

Answer \$antw $\pm$ 0.1

Figure 1.147: title of the figure

TIP: Let op dat *duizendtalseparatoren* beslist niet geaccepteerd worden in dit vraagtype. Wilt u deze dan toch gebruiken, neem dan uw toevlucht tot het vraagtype *Numeric*.

Hieronder is te zien welke *Sub-typen* er mogelijk zijn voor de vraagstelling binnen dit vraagtype.

Insert Response Area

Question Feedback

Mathematical Formula:

Weighting 1

Sub-type: Formula - e.g. $e^x \sin(x^2)$

Answer: Formula - e.g. $e^x \sin(x^2)$

- Formula without logs and trig - e.g. $x^2/(x^2+1)$
- Formula with physical units - e.g. $-9.8 (1/2)t^2 \text{ m/s}^2$
- Formula that matches responses to within +C
- Formula without simplification
- Equation - e.g. $x+3y=1$
- Unordered list of formulas - e.g. $1 + \sqrt{5}$; $1 - \sqrt{5}$
- Ordered lists of formulas - e.g. $2x^3, 4x$
- Vector of formulas - e.g. $(2t, t+1, 3t^2)$
- Chemical Equation - e.g. $2H_2 + O_2 \rightarrow 2H_2O$

Figure 1.148: Subtypes van het vraagtype Formula

In de volgende paragrafen worden de mogelijkheden besproken.

1.5.2 Formula - e.g. $e^x \sin(x^2)$

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.

U vult het juiste antwoord in en er komt verder geen Grading Code aan te pas. Maple wordt hier niet ingeschakeld bij de grading of bij het correcte antwoord.

Bij het getal π maakt het niet uit of de student Pi of pi intikt. Beide keren wordt het als de bedoelde constante 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 Σ -knop. Dat wil zeggen kiezen tussen tekst-invoer of met behulp van een Editor.

De *Preview*-functie bij tekst invoer 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. Het advies is om dit vraagtype met deze instelling niet te gebruiken.

1.5.3 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, $1/2\pi$ wordt niet goed gerekend.

Is bijvoorbeeld het antwoord $1/2\pi$, dan wordt bij de instelling *Formula without simplification* $5\pi/10$ wel goedgekend maar $\pi/2$ niet

Bij de instelling met *Formula* zou dit allemaal wel goedgekend worden. Het is raadzaam dit soort dingen even uit te proberen.

Bij de instelling met *Formula* kan bijvoorbeeld goedgekend worden: $\pi/2 = \pi/2 = 5\pi/10 = 5\pi/10$ enz. Hoofdletter en kleine letter pi of π maakt niet uit.

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

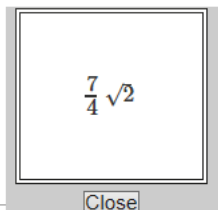
Jump To: [Question](#) | [Information Fields](#)

Question:

Evaluate:

$$\lim_{x \rightarrow \infty} \left(\frac{7x+6}{\sqrt{8x^2+4x+7}} \right) = \frac{7}{4}\sqrt{2}$$

Enter the answer in the simplified form and no decimal numbers.



Evaluate:

$$\lim_{x \rightarrow \infty} \left(\frac{7x+6}{\sqrt{8x^2+4x+7}} \right) = \frac{7}{4}\sqrt{2}$$

Enter the answer in the simplified form and no decimal numbers.

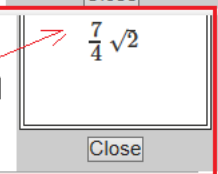


Figure 1.149: Limieten met wortelvorm

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: $7/4\sqrt{2}$.

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 dus zonder maple want die maakt er weer een macht van met exponent $1/2$.

`$ans="$d/$a1*sqrt($a1)";`

Dan maakt u het antwoordveld aan met vraagtype *Formula* en subtype *Formula without simplification* als volgt.

Edit Response Area

Question

Feedback

Mathematical Formula:

Weighting

Sub-type:

Answer

Figure 1.150: 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 1.149 (page 136)* 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.

Evaluate:

$$\lim_{x \rightarrow \infty} \left(\frac{8x+4}{\sqrt{8x^2+8x+5}} \right) =$$

$22^{\frac{1}{2}}$

Close

Enter the answer in the simplified form and no decimal numbers.

Figure 1.151: Preview met wortels

TIP: Eventueel overheort u het antwoord in bijvoorbeeld twee delen met een apart antwoordveld voor het getal en nog een apart antwoordveld voor de wortel.

Zie ook in de paragraaf over wortels en Grading Code bij het Maple-graded vraagtype (*page 82*))

1.5.4 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.

1.5.5 Geordende en ongeordende Lijsten en Vectoren met Formula

Met het vraagtype *Mathematical Formula* is het mogelijk om **geordende lijsten** van getallen of formules te overhoren met het *Sub-type* instelling *Ordered list of formulas*. U geeft dan gewoon het rijtje geordende objecten bij het juiste antwoord (Answer). Wel of geen haakjes (vierkant of niet) eromheen maakt niet uit.

Precies hetzelfde werkt het subtype *Vector of formulas*, maar ook het subtype *Unordered list of formulas*.

Deze *Sub-typen* zijn in feite allemaal hetzelfde met twee dingen van belang.

- Volgorde wel van belang: gebruik komma's als scheidingsteken.
- Volgorde niet van belang: gebruik puntkomma's als scheidingsteken.
- Haakjes (rechte of gewone haakjes) er wel of niet omheen maakt niet uit. (Communiceer dit met de student.)

TIP: De student kan bij deze instellingen NIET kiezen voor de Editor.

Edit Response Area

Question Feedback

Mathematical Formula:

Weighting 1

Sub-type: Ordered lists of formulas - e.g. $2x^3$, $4x$

Answer

- Formula - e.g. $e^x \sin(x^2)$
- Formula without logs and trig - e.g. $x^2/(x^2+1)$
- Formula with physical units - e.g. $-9.8 (1/2)t^2 \text{ m/s}^2$
- Formula that matches responses to within +C
- Formula without simplification
- Equation - e.g. $x+3y=1$
- Unordered list of formulas - e.g. $1 + \sqrt{5}$; $1 - \sqrt{5}$
- Ordered lists of formulas - e.g. $2x^3$, $4x$**
- Vector of formulas - e.g. $(2t, t+1, 3t^2)$
- Chemical Equation - e.g. $2\text{H}_2 + \text{O}_2 \rightarrow 2\text{H}_2\text{O}$

Figure 1.152: Ongeordende en geordende lijsten bij vraagtype Mathematical Formula

Een ONgeordende lijst kan overhoord worden met ook alle drie de *Sub-typen*. Echter dan zijn de verschillende getallen of formules gescheiden door **puntkomma's**.

TIP: Dan is er ook nog de mogelijkheid om vectoren (een rijtje getallen of formules met haakjes eromheen en gescheiden door komma's) te overhoren.

Geef in willekeurige volgorde de lijst met vectoren (3,4) en (5,7) en scheid deze vectoren door middel van een puntkomma.

Mathematical Formula

vector of formulas

Edit Response Area

Question

Feedback

Mathematical Formula:

Weighting

Sub-type:

Answer

Figure 1.153: Question Designer Lijsten en Vectoren

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.

Dat kan wel bij het Maple-graded vraagtype door te werken met verzamelingen (*page 67*)

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.

TIP: U kunt ook nog tolerantie aanbrengen in het antwoord met het vraagteken zoals (3.1 ? 0.1, 5.7 ? 0.1).

1.5.6 Eenvoudige vergelijkingen

Voor het gebruik van het sub-type *Equation* van het vraagtype *Mathematical Formula* is even wat extra informatie nodig.

Gebruik dit vraagtype alleen voor eenvoudige vergelijkingen. De student kan de vergelijking intikken in welke vorm dan ook. Het nadeel hiervan is dat de student geen sterren voor vermenigvuldiging hoeft in te tikken (maar het mag wel). De syntax is dus niet streng. Er is geen Grading Code mogelijk dus u moet het hiermee doen.

Als u het correcte antwoord invult bij Answer, 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: $x=(5-2*y)/3$ of $y=5/2-3/2*x$, 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 37)).

Het 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 een Σ -knopje voor de student beschikbaar om te wisselen tussen Editor

en tekstveld. Het is dezelfde low-level editor als bij de instelling *Formula* bij de *Maple-graded vraagtype*.

Edit Response Area

Question

Feedback

Mathematical Formula:

Weighting

Sub-type:

Answer

```

1 $verg="2*y+3*x=5";
2 $vergdisplay=maple("printf(MathML[ExportPresentation]($verg))");
3 $antw=maple("isolate($verg,y)");
4

```

Variable	Value
verg	$2*y+3*x=5$
vergdisplay	$2y + 3x = 5$
antw	$y = 5/2-3/2*x$

Figure 1.154: Vergelijkingen met het vraagtype Formula

In bovenstaande figuur moet in de *Answer code* de vergelijking in expliciete vorm ingevoerd zijn. In het *Algorithm* is dit antwoord in de juiste vorm voorbereid, dus mocht ook in de vorm $x=(5-2*y)/3$.

De student mag dan elke gelijkwaardige vorm invullen om een goede beoordeling te krijgen.

1.5.7 Chemische formules met Formula

Met de instelling *Chemical Equation* e.g $2H_2+O_2 \rightarrow 2H_2O$ kan het correcte antwoord ingevoerd worden.

De student kan bij deze instelling kiezen voor tekst invoer of met behulp van de Editor. De student zet de editor aan met behulp van de Σ -knop onder het veld als het veld zichtbaar is als tekstveld.

In de editor wordt een bijpassend menu aangeboden met knoppen voor sub- en superscript en pijltjes.

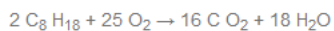
Het pijltje kan ook met het toetsenbord gemaakt worden met $\rightarrow$ en superscript en subscript met $\wedge$ en underscore $_$.

De Editor is niet zo uitgebreid, maar voor scheikundeformules biedt het goede tools met de pijltjes en de subscripten.



Preview

Tik de volgende reactievergelijking over



Tik ook 1 als je 1 ziet staan.

Figure 1.155: Chemische formules

Ook hier is het belangrijk dat de student oefent met het intikken van formules in deze editor. Maak een paar vragen om formules na te tikken.

Het juiste antwoord tikt u in als volgt:

2 C_8 H_18 +25 O_2->16 C O_2+18 H_2 O

Edit Response Area

Question

Feedback

Mathematical Formula:

Weighting

Sub-type:

Answer

Figure 1.156: Correcte antwoord van een chemische formule

1.6 Vraagtype Numeric

Zie Handleiding Möbius Items Maken Deel A.

TIP: De enige instelling bij het vraagtype Numeric van belang kan zijn voor formules is de instelling *Accept arithmetic*.

Edit Response Area

Question Feedback

Numeric

Weighting: 1

Numeric Part: \$ans

Units Part:

Numeric Format: ☐ Accept 1000s separator ☐ Accept scientific notation ☐ Accept \$ sign ☒ Accept arithmetic

Required with: Absolute accuracy

-1234.0; 23 - 1257

Figure 1.157: title of the figure

De student kan dan de berekening invoeren met haakjes dakjes sterren en dergelijke zonder de rekenmachine te hanteren. Zet de *accuracy* dan op *Absolute*.

Omdat de grading buiten Maple om gaat, is hier de betekenis van log de logaritme met grondtal 10 en ln de logaritme met grondtal e.

Zie ook paragraaf over log en exp en pi (page 58)).

TIP: Het getal e kan wel als antwoord dienen in het numerieke vraagtype, maar het kan niet als antwoord ingevuld worden ook al staat *Accept arithmetic* aangevinkt.

1.7 Dynamische figuren

Dit onderdeel staat nu in *Handleiding Möbius 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.
- `add(x, x in L)` Hiermee tel je alle elementen op die in lijst L zitten.
- `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).
- `evalf[3]` Evalueert een getal naar een decimaal getal (met floating point) met drie significante cijfers.

- `factor` Factoriseert een uitdrukking.
- `frac(breuk)` Geeft het gedeelte van de breuk dat overblijft nadat de helen er afgehaald zijn.
- `Float(round((6 * $var1/($var1 + 4))*10^1), -1)` Geeft het resultaat van de berekening met één decimaal.
- `MapleTA:-Builtin:-decimal(3,5*1.7^4)` Geeft het resultaat van de berekening met drie decimalen.
- `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" = ggk = 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 gevalueerde 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]($matrix)` Om de determinant van een matrix te berekenen.
- `LinearAlgebra[GenerateMatrix]([$e1, $e2, $e3], [x, y, z], augmented=true)` berekent van de drie vergelijkingen \$e1 en \$e2 en \$e3 in de variabelen x, y en z de aangevulde matrix. (3 rijen en 4 kolommen).
- `LinearAlgebra[Rank]($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 ($\sim$) 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 ($\sim$) 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.
- `parse ("...")` maakt van een string weer een gewone formule zonder quotes. (strippen van quotes).
- `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 \$RESPONSE 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 $i k^2$ 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]);
```

$[3, x^2]$ (1.84)

Betekent dat je een lijst krijgt met de tweede operand van het eerste kental en de eerste operand van het tweede kental.

```
> zip((x, y) -> x+y, [1, 2, 3, 7], [4, 4, 3, 1]);
```

$[5, 6, 6, 8]$ (1.85)

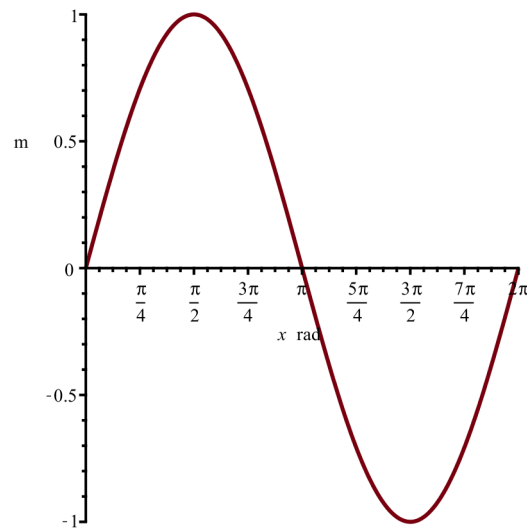
De lijsten zijn even lang. Volgens voorschrift van de functie worden alle elementen opgeteld.

```
> zip((x, y) -> x+y, [1, 2, 3, 7], [4, 4], 0);
```

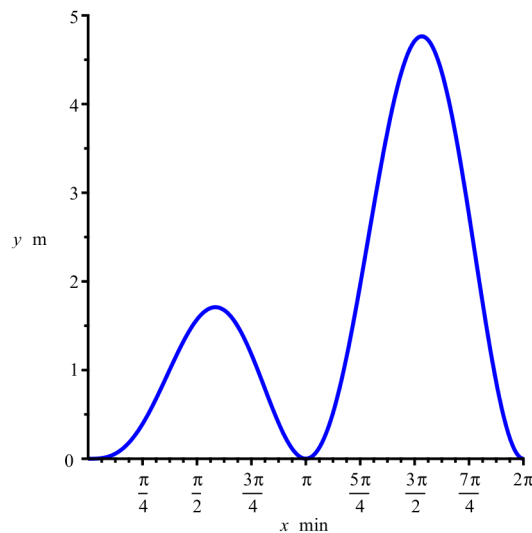
$[5, 6, 3, 7]$ (1.86)

Hier zijn de lijsten niet even lang. De betekenis is dat de elementen opgeteld moeten worden en als de tweede lijst niet genoeg elementen heeft, tel er dan 0 bij op.

```
> 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);
```



```
> v1:=3*x^2-6*y=0;
```

$$v1 := 3x^2 - 6y = 0 \quad (1.87)$$

```
> v2:=3*y^2-6*x=0;
```

$$v2 := 3y^2 - 6x = 0 \quad (1.88)$$

```
> [solve({v1,v2},{x,y})];
```

```
[{x=0,y=0},{x=2,y=2},{x=-2*RootOf(_Z^2+_Z+1)-2,y=2*RootOf(_Z^2+_Z+1)}] (1.89)
```

```
> use RealDomain in simplify([solve({v1,v2},{x,y})]) end use;
```

$$[x=2,y=2], [x=0,y=0] \quad (1.90)$$

Als er complexe oplossingen van reële vergelijkingen komen, zijn die uit te sluiten door in de reële getallen te werken.

Index

