

# SMath Studio mit Maxima

**Einführung und Referenz für Version 0.99.7561**

Prof. Dr.-Ing. Martin Kraska

[kraska@th-brandenburg.de](mailto:kraska@th-brandenburg.de)

16. September 2020

# Inhaltsverzeichnis

|   |           |
|---|-----------|
| <b>1. Einleitung</b>                                      | <b>8</b>  |
| 1.1. Funktionsüberblick                                   | 12        |
| 1.2. SMath im Internet                                    | 13        |
| 1.3. SMath unter Linux                                    | 16        |
| 1.4. Alternativen   | 16        |
| 1.5. Literatur  | 18        |
| <b>2. Installation</b>                                    | <b>21</b> |
| 2.1. Standardinstallation (Windows)                       | 21        |
| 2.2. Portable THB-Distribution                            | 23        |
| 2.3. Linux  | 23        |
| <b>3. Erste Schritte</b>                                  | <b>25</b> |
| 3.1. Speichern und Rückgängig                             | 25        |
| 3.2. SMath als Taschenrechner                             | 25        |
| 3.3. Namen  | 27        |
| 3.4. Ergebnisdarstellung und Runden                       | 29        |
| 3.5. Kopieren und Verschieben                             | 30        |
| 3.6. Textbereiche   | 31        |
| 3.7. Bilder einfügen <sup>1</sup>                         | 32        |
| 3.8. Vektoren und Matrizen                                | 34        |
| 3.9. Diagramme  | 36        |
| 3.10. Etwas Programmierung                                | 39        |
| 3.11. Tips und Tricks                                     | 40        |
| <b>4. Programmbedienung</b>                               | <b>42</b> |
| 4.1. Benutzeroberfläche                                   | 42        |
| 4.1.1. Hauptmenü  | 43        |
| 4.1.2. Werkzeugleiste                                     | 44        |
| 4.1.3. Seitenleiste                                       | 44        |
| 4.1.4. Statuszeile  | 45        |
| 4.2. Dynamische Hilfe                                     | 45        |
| 4.3. Interaktives Handbuch                                | 46        |
| 4.4. Die Zwischenablage                                   | 48        |
| 4.4.1. Aus SMath in die Zwischenablage kopieren           | 48        |
| 4.4.2. Einfügen aus der Zwischenablage                    | 49        |
| 4.5. Verzeichnisse  | 49        |
| 4.6. Import/Export  | 52        |
| 4.6.1. Export mit „Speichern unter“                       | 52        |
| 4.6.2. Snapshot-Bereich für Bildschirmfotos (Screenshots) | 53        |
| 4.7. Seitenansicht  | 55        |

---

<sup>1</sup>Erfordert das Plug-in *Image Region*

|  |           |
|--|-----------|
| 4.8. Drucken . . . . .   | 56        |
| 4.9. Einstellungen . . . . .   | 56        |
| 4.9.1. Hintergrundbilder . . . . .   | 56        |
| 4.9.2. Dialog „Blatteinstellungen“ . . . . .                                   | 57        |
| 4.9.3. Dialog „Datei Eigenschaften“ . . . . .                                  | 58        |
| 4.9.4. Dialog „Einstellungen“ . . . . .  | 59        |
| <b>5. Das Rechenblatt</b> . . . . .  | <b>62</b> |
| 5.1. Bereiche bearbeiten . . . . .   | 62        |
| 5.2. Kopf- und Fußzeilen . . . . .   | 63        |
| 5.3. Dynamische Felder . . . . .   | 64        |
| 5.4. Formelbereich . . . . .   | 64        |
| 5.5. Textbereich . . . . .   | 65        |
| 5.5.1. Textbereiche mit Verweis (Hyperlink) . . . . .                          | 66        |
| 5.6. Erweiterter Textbereich . . . . .   | 67        |
| 5.6.1. Formeln und Hyperlinks . . . . .  | 68        |
| 5.6.2. Voreinstellungen . . . . .  | 68        |
| 5.7. Diagrammbereiche . . . . .  | 69        |
| 5.8. Image-Bereich . . . . .   | 71        |
| 5.9. Tabellenbereich . . . . .   | 73        |
| 5.10. Hyperlink-Bereich . . . . .  | 77        |
| 5.11. Formatierte Textausgabe . . . . .  | 78        |
| 5.12. Steuerelemente (Controls) . . . . .                                      | 79        |
| 5.13. Beschreibungstexte . . . . .   | 83        |
| 5.14. Trennlinien . . . . .  | 84        |
| 5.15. Mehrsprachigkeit . . . . .   | 86        |
| 5.16. Lesbare Rechenblätter . . . . .  | 87        |
| 5.17. Beispiel . . . . .   | 89        |
| <b>6. Mathematik</b> . . . . .   | <b>91</b> |
| 6.1. Formelbereich . . . . .   | 91        |
| 6.1.1. Einfügemarke in Formelbereichen . . . . .                               | 91        |
| 6.1.2. Griechische Buchstaben . . . . .  | 91        |
| 6.1.3. Maßeinheiten . . . . .  | 92        |
| 6.1.4. Internes Format . . . . .   | 93        |
| 6.1.5. Formatierung . . . . .  | 94        |
| 6.2. Grundbegriffe . . . . .   | 94        |
| 6.3. Symbolisch und numerisch . . . . .  | 96        |
| 6.3.1. Rechengenauigkeit . . . . .   | 98        |
| 6.4. Format der Ergebnisanzeige . . . . .                                      | 99        |
| 6.4.1. Darstellung von Dezimalzahlen . . . . .                                 | 99        |
| 6.4.2. Abtrennen eines Faktors (Zehnerpotenz, Konstante, Maßeinheit) . . . . . | 101       |
| 6.4.3. Prozentrechnung . . . . .   | 102       |
| 6.5. Funktionen und Operatoren . . . . .                                       | 102       |
| 6.5.1. Winkelfunktionen . . . . .  | 104       |
| 6.5.2. Logische und Vergleichsoperationen . . . . .                            | 106       |
| 6.5.3. Stückweise stetige Funktionen . . . . .                                 | 108       |
| 6.6. Rechnen mit Variablen . . . . .   | 110       |
| 6.6.1. Namen . . . . .   | 112       |
| 6.6.2. Definition symbolischer Ausdrücke . . . . .                             | 112       |

|           |  |     |
|-----------|--|-----|
| 6.6.3.    | Variablen löschen  | 113 |
| 6.6.4.    | Matrizen und Vektoren  | 114 |
| 6.6.5.    | Die Substitutionsfunktion <code>at()</code>                            | 115 |
| 6.6.6.    | Parametrische Rechnungen   | 115 |
| 6.6.7.    | Zwischenspeichern von Variablen in Dateien                             | 117 |
| 6.7.      | Komplexe Zahlen  | 118 |
| 6.8.      | Matrizen und Vektoren  | 120 |
| 6.8.1.    | Interaktive Erzeugung von Matrizen                                     | 120 |
| 6.8.2.    | Große Matrizen anzeigen  | 122 |
| 6.8.3.    | Der <code>SMathMatrixEditor</code>                                     | 122 |
| 6.8.4.    | Funktionen zur Erzeugung von Matrizen und Vektoren                     | 123 |
| 6.8.5.    | Matrixalgebra  | 125 |
| 6.8.6.    | Manipulation von Matrizen  | 129 |
| 6.8.7.    | Funktionen auf Vektor- oder Matrixelemente anwenden                    | 134 |
| 6.8.8.    | Summen und Produkte  | 136 |
| 6.8.9.    | Vektorrechnung in kartesischen Koordinaten                             | 137 |
| 6.9.      | Listen   | 140 |
| 6.9.1.    | Listenobjekte erzeugen   | 141 |
| 6.9.2.    | Rechnen mit Listen   | 143 |
| 6.9.3.    | Umwandlung Liste $\longleftrightarrow$ Vektor                          | 144 |
| 6.9.4.    | Maxima-Funktionen für Listen   | 144 |
| 6.10.     | Analysis   | 145 |
| 6.10.1.   | Differenzialrechnung   | 146 |
| 6.10.2.   | Symbolische Integration mit Maxima                                     | 148 |
| 6.10.3.   | Numerische Integration   | 150 |
| 6.11.     | Gleichungslösung   | 155 |
| 6.11.1.   | Symbolische Lösungen mit <code>Solve()</code>                          | 156 |
| 6.11.1.1. | Anwendungsbeispiel: Lineares System                                    | 157 |
| 6.11.1.2. | Anwendungsbeispiel: Nichtlineares System                               | 159 |
| 6.11.1.3. | Beispiel: Schnittpunkt von Kurven                                      | 160 |
| 6.11.2.   | Numerische Lösung mit Startwerten: <code>FindRoot()</code>             | 161 |
| 6.11.3.   | Numerische Lösung mit vorgegebenem Intervall: <code>Bisection()</code> | 162 |
| 6.12.     | Gewöhnliche DGL  | 163 |
| 6.12.1.   | Numerische Löser   | 164 |
| 6.13.     | Lorentz-Attraktor  | 169 |
| 6.13.1.   | Löser aus dem Maxima-Plug-in   | 170 |
| 6.14.     | Datenimport und -export  | 171 |
| 6.14.1.   | Funktionsübersicht   | 173 |
| 6.14.2.   | Textdateien lesen mit <code>importData()</code>                        | 173 |
| 6.14.3.   | XLSX- und ODS-Tabellen   | 174 |
| 6.15.     | Statistik und Datenanalyse   | 175 |
| 6.15.1.   | Deskriptive Statistik  | 176 |
| 6.15.2.   | Spezielle Funktionen   | 176 |
| 6.15.3.   | Verteilungen   | 177 |
| 6.15.4.   | Multilineare Regression  | 177 |
| 6.15.5.   | Nichtlineare Regression  | 179 |
| 6.15.6.   | Interpolation  | 182 |

|   |            |
|---|------------|
| <b>7. SMath Studio mit Maxima</b>                                 | <b>185</b> |
| 7.1. Installation und Konfiguration                               | 185        |
| 7.2. Funktionsübersicht   | 186        |
| 7.3. Die Maxima-Sitzung   | 188        |
| 7.4. Die Funktion <code>Maxima()</code>                           | 190        |
| 7.5. Logging  | 191        |
| 7.5.1. <code>MaximaLog()</code>                                   | 191        |
| 7.5.2. Log Window   | 191        |
| 7.6. Configuration  | 191        |
| 7.6.1. Example For Custom Definitions in <i>maxima.xml</i>        | 192        |
| <b>8. Grafik</b>  | <b>194</b> |
| 8.1. Diagramme mit <i>Maxima</i>                                  | 195        |
| 8.1.1. Funktionen <code>Draw2D()</code> und <code>Draw3D()</code> | 196        |
| 8.1.2. Achsen und Gitter  | 198        |
| 8.1.3. Beschriftungen   | 198        |
| 8.1.4. Maxima-Diagrammbereiche                                    | 199        |
| 8.1.5. Grafikobjekte  | 201        |
| 8.1.6. Optionen   | 217        |
| 8.1.7. Nützliche Befehle für <code>user_preamble</code>           | 222        |
| 8.2. 2D-Diagramme (Standard)                                      | 223        |
| 8.2.1. Funktionen   | 224        |
| 8.2.2. Polygonzüge  | 225        |
| 8.2.3. Text   | 225        |
| 8.2.4. Farben   | 226        |
| 8.2.5. Achsengrenzen ändern                                       | 227        |
| 8.2.6. Animationen steuern  | 228        |
| 8.3. X-Y Plot   | 229        |
| 8.3.1. Explizite Funktionen                                       | 230        |
| 8.3.2. Implizite Funktionen                                       | 231        |
| 8.3.3. Text   | 233        |
| 8.3.4. Geometrische Grundformen                                   | 233        |
| 8.3.5. Mausfunktionen im X-Y Plot                                 | 235        |
| 8.3.6. Einstellungen  | 235        |
| 8.4. Der Modeller-Bereich   | 238        |
| 8.4.1. Erste Schritte mit dem Modeller                            | 238        |
| 8.4.2. Objektdefinition (Templates)                               | 240        |
| 8.4.3. Predefined Layout (vordefinierte Anordnung)                | 241        |
| 8.4.4. Dependent Layout (dynamische Anordnung)                    | 241        |
| 8.4.5. Ausgabe  | 242        |
| 8.4.6. Beispiel: Modelleditor für Statikberechnungen              | 242        |
| 8.4.7. Weiterführende Ressourcen                                  | 244        |
| <b>9. Maßeinheiten</b>  | <b>245</b> |
| 9.1. Maßeinheit in Ergebnissen                                    | 247        |
| 9.2. Eigene Maßeinheiten  | 248        |
| 9.3. Vektoren und Matrizen mit Maßeinheiten                       | 248        |
| 9.4. Temperatureinheiten  | 249        |
| 9.4.1. Kurzfassung  | 249        |
| 9.4.2. Details  | 249        |

|  |            |
|--|------------|
| 9.5. Winkel und Umdrehungen . . . . .                | 251        |
| 9.6. Winkel in Grad, Minuten, Sekunden . . . . .     | 252        |
| 9.7. Plotten mit Maßeinheiten . . . . .              | 253        |
| 9.8. Maßeinheits-Funktionen . . . . .                | 254        |
| 9.9. Vordefinierte Konstanten . . . . .              | 255        |
| <b>10. Programmierung</b>                            | <b>257</b> |
| 10.1. Benutzerfunktionen . . . . .                   | 257        |
| 10.1.1. Einfache Funktionen . . . . .                | 258        |
| 10.1.2. Symbolische Variable . . . . .               | 261        |
| 10.1.3. Prozeduren . . . . .                         | 262        |
| 10.2. Ablaufsteuerung . . . . .                      | 266        |
| 10.2.1. Verzweigung . . . . .                        | 266        |
| 10.2.2. Schleifen . . . . .                          | 266        |
| 10.2.3. Angabe der Schrittweite . . . . .            | 268        |
| 10.2.4. Abfangen von Fehlern . . . . .               | 268        |
| 10.3. Fehlersuche . . . . .                          | 269        |
| 10.4. Funktionen für Zeichenketten . . . . .         | 271        |
| 10.5. Code-Bausteine . . . . .                       | 275        |
| 10.5.1. Erzeugung eines Bausteins . . . . .          | 275        |
| 10.5.2. Verwendung von Bausteinen . . . . .          | 278        |
| <b>11. Der SMath Viewer</b>                          | <b>280</b> |
| 11.1. Einführungsbeispiel . . . . .                  | 280        |
| 11.2. Gestaltungsmöglichkeiten . . . . .             | 282        |
| 11.3. Anwendungshinweise . . . . .                   | 284        |
| 11.4. Lizenz . . . . .                               | 285        |
| 11.5. Veröffentlichung . . . . .                     | 285        |
| <b>12. Erweiterungen</b>                             | <b>287</b> |
| 12.1. Erweiterungsmanager . . . . .                  | 287        |
| 12.2. Erweiterungsarten . . . . .                    | 289        |
| 12.2.1. Handbücher (Handbook) . . . . .              | 289        |
| 12.2.2. Beispiele (Examples) . . . . .               | 289        |
| 12.2.3. Interaktive Handbücher (Book) . . . . .      | 290        |
| 12.2.4. Plug-ins . . . . .                           | 291        |
| 12.2.4.1. Upload of Plug-ins . . . . .               | 292        |
| 12.2.5. Anwendungen (Applications) . . . . .         | 292        |
| 12.2.6. Code-Bausteine (Snippets) . . . . .          | 293        |
| 12.3. Das Registrierungsportal . . . . .             | 293        |
| 12.3.1. Versionierung von SMath-Dokumenten . . . . . | 294        |
| 12.4. Quellcode auf dem SVN-Server . . . . .         | 295        |
| <b>A. Glossar</b>                                    | <b>296</b> |
| <b>B. Operatoren</b>                                 | <b>297</b> |
| <b>C. Tastenbelegung</b>                             | <b>301</b> |
| <b>D. Vordefinierte Maßeinheiten und Konstanten</b>  | <b>304</b> |

---

|  |            |
|--|------------|
| <b>E. Menüstruktur</b>                     | <b>315</b> |
| <b>F. Portables SMath mit Maxima</b>       | <b>322</b> |
| F.1. Portables SMath                       | 322        |
| F.2. Konfiguration                         | 324        |
| F.3. Eigene Maßeinheiten                   | 324        |
| F.4. Auslieferung                          | 326        |
| F.5. Handlungsbedarf                       | 326        |
| <b>G. Dachboden</b>                        | <b>329</b> |
| G.1. Schriftart in Textbereichen ändern    | 329        |
| G.2. solve() und roots()                   | 330        |
| G.3. Ebene Polygone                        | 333        |
| G.3.1. Polygon-Objekte erzeugen            | 334        |
| G.3.2. Polygon-Objekte darstellen          | 335        |
| G.3.3. Polygon-Operationen                 | 335        |
| G.3.4. Polygon-Objekte lesen und schreiben | 338        |
| G.4. 3D-Diagramme (Standard)               | 338        |
| G.4.0.1. 3D-Diagrammbereich erzeugen       | 339        |
| G.4.0.2. Darstellbare Objekte              | 339        |
| G.4.0.3. Navigation im 3D-Diagrammfenster  | 340        |
| G.5. Diagramme mit plot2D/3D               | 341        |
| G.5.1. Optionen                            | 341        |
| G.6. Wie dieses Handbuch entsteht          | 343        |
| <b>Literaturverzeichnis</b>                | <b>343</b> |
| <b>Index</b>                               | <b>345</b> |

# 1. Einleitung

SMath Studio ist ein Programm zur Durchführung und Dokumentation technischer Berechnungen. Ein SMath-Dokument besteht aus frei auf dem Papier angeordneten Formeln, Texten, Diagrammen und Bildern. Formeln kann man aus taschenrechnerartigen Paletten zusammenklicken oder als erfahrener Anwender ganz ohne Maus, nur mit der Tastatur erstellen. Großartig ist die umfassende Unterstützung von Maßeinheiten. Die Formeln sind gleichzeitig Rechenanweisung und Dokumentation. Formeln erscheinen in natürlicher mathematischer Schreibweise (nicht wie Programmcode), so dass auch Dritte, die das Programm nicht kennen, sie verstehen können.

Dieses Handbuch soll den Anwender bei der Einarbeitung und beim Erkunden der Möglichkeiten mit SMath Studio unterstützen. Grundlage dafür sind die langjährige Erfahrung des Autors im Einsatz von SMath Studio in Forschung und Lehre an der TH Brandenburg sowie eine Vielzahl von Beiträgen im Anwenderforum auf [smath.info](http://smath.info).

Die an der Technischen Hochschule Brandenburg entwickelte Maxima-Schnittstelle erweitert SMath Studio um leistungsfähige Computer-Algebra-Funktionen. So ermöglicht sie z.B. die symbolische Integration und Gleichungslösung. Zudem erschließt sie die Grafikmöglichkeiten von Gnuplot für den SMath-Anwender.

Frühere Versionen dieses Handbuchs haben sich auf die portable Distribution „SMath mit Maxima“, wie sie vom Autor zusammengestellt wurde ([Download<sup>1</sup>](#)) bezogen. Mittlererweile ist die Konfiguration der Maxima-Schnittstelle aber stark vereinfacht, so dass man als Anwender gut mit der Standardversion von der SMath-Website zurecht kommt. Zudem ist die Maxima-Installation inzwischen auf über 130 MB (gezippt) und 600 MB (installiert) angewachsen, so dass eine standardmäßige Bündelung mit SMath Studio nicht mehr sinnvoll erscheint.

Daher ist die nun als „THB-Distribution“ bezeichnete Distribution auf ein vorkonfiguriertes portables SMath-Studio beschränkt.

Kapitel 1 ordnet SMath in die Welt der Mathematikprogramme ein und gibt Hinweise zum Leistungsumfang.

Kapitel 2 erläutert die Standardinstallation und die Installation der portablen THB-Distribution.

Kapitel 3 führt in einer Schritt-für-Schritt-Anleitung durch die Installation und die Bedienung des Programms. Man kann sie in 1-2 Stunden durcharbeiten.

Kapitel 4 beschreibt die Benutzeroberfläche mit der dynamischen Hilfe und dem interaktiven Handbuch. Zudem dokumentiert es betriebssystemnahe Funktionen wie Zwischenablage, Drucken, Import/Export und diverse Einstellungsmöglichkeiten.

Kapitel 5 bietet eine Übersicht über die Grundbausteine und Gestaltungsmöglichkeiten für Rechenblätter (Bereiche wie Text, Formeln, Diagramme und Bilder, Kopf- und Fußzeilen, Hintergrundbilder).

---

<sup>1</sup><http://smath.info/wiki/SMath%20with%20Plugins.aspx>



Kapitel 6 beschreibt die mathematischen Fähigkeiten von SMath. Hier sind auch die auf Maxima basierenden Funktionen enthalten. Installation und Bedienkonzept des Maxima-Plug-ins werden in Kapitel 7 behandelt.

Kapitel 8 dokumentiert die Diagrammfunktionen (aus dem Maxima-Plug-in, im Standard-Diagrammbereich und im sogenannten X-Y Plot-Bereich).

Kapitel 9 dokumentiert den Umgang mit Maßeinheiten. Besonders wird auf Temperatur- sowie Drehzahl- und Frequenzeinheiten eingegangen. Eine Auflistung aller in SMath vordefinierter Maßeinheiten und Konstanten findet sich in Anhang D.

Kapitel 10 behandelt die Möglichkeiten zur Definition eigener Funktionen und Algorithmen in SMath-Arbeitsblättern. Abschnitt 10.5 zeigt, wie SMath-Programme als Bausteine in anderen Arbeitsblättern verwendet werden können. Der Umgang mit Zeichenketten wird in Abschnitt 10.4 erläutert.

Seit der Version 0.96 (Mai 2013) kann SMath Studio selbstständige Anwendungen exportieren, die unter dem Namen *SMath Viewer* laufen. Kapitel 11 erläutert dies näher.

Kapitel 12 behandelt den Erweiterungsmanager und externe Plug-ins, mit denen der Funktionsumfang von SMath gegenüber der Standardinstallation erweitert werden kann.

Das letzte Kapitel enthält Anwendungsbeispiele und Hinweise zur Verwendung des Programms bei der Dokumentation technischer Berechnungen, darunter auch Mechanik-Aufgaben.

Die Anhänge enthalten Referenzmaterial wie z.B.

- Liste der Operatoren und deren Tastatureingabe
- Tastenbelegung
- Menüstruktur
- Vordefinierte Maßeinheiten und Konstanten
- „Dachboden“ mit historischen Abschnitten zu Funktionen, die keine wesentliche Rolle mehr in der Anwendung von SMath Studio spielen.

Abgeschlossen wird das Handbuch durch ein Literatur- und ein Stichwortverzeichnis.

Kritik und Anregungen sind willkommen und können an [kraska@th-brandenburg.de](mailto:kraska@th-brandenburg.de) gerichtet werden.

## Änderungen gegenüber der Handbuchversion von 2019

Weiterentwicklungen von SMath Studio:

- Starke Verbesserung des Textbereichs: Automatischer Zeilenumbruch, Einbettung anderer Bereiche, insbesondere von Formelbereichen,
- Arial als neue Standardschriftart im Textbereich,
- Erweiterte Steuerungsmöglichkeiten im X-Y Plot.

## Änderungen gegenüber der Handbuchversion von 2018

Weiterentwicklungen von SMath Studio:

- Kopf- und Fußzeilen in Rechenblättern (siehe Abschnitt 5.2),
- Zugriff auf Dokumenteigenschaften in Ausdrücken (siehe Abschnitt 5.3),
- Plug-in-Filter im Funktions-Dialog,
- Auswahl überlappender Bereiche (z.B. Annotationen in Diagrammen und Skizzen) wurde verbessert.
- Verbesserte Unterstützung hochauflösender Displays,
- Fehlermeldungen markieren nun die betroffenen Unterausdrücke,
- Flexiblere Fehlerbehandlung: (Abbruch, Pause oder Weiterrechnen).
- Das Format für Geometrieobjekte im X-Y Plot hat sich geändert.

Das Kapitel „Das Rechenblatt“ ist neu und behandelt die Gestaltungsmöglichkeiten (Bereichstypen, Einstellungen) für Rechenblätter. Das Kapitel „Anwendungen“ wurde aufgelöst und auf andere Kapitel verteilt.

## Änderungen gegenüber der Handbuchversion von 2017

SMath Studio hat sich erheblich weiterentwickelt.

- Numerische Ergebnisse werden in diesem Handbuch generell mit drei signifikanten Stellen ohne nachfolgende Nullen dargestellt. Diese Möglichkeit ist neu, frühere Handbuchversionen basierten auf der Voreinstellung mit vier Nachkommastellen ohne nachfolgende Nullen.
- Indexvektoren und implizite Schleifen mit Bereichsvariablen sind möglich.
- Die Formatierung von Formeln wurde umgestellt. Nun sind anwenderdefinierte Formelzeichen kursiv, Textindizes zudem in kleinerer Schrift und Maßeinheiten erscheinen aufrecht (und nach wie vor blau).
- Bereiche werden nun in eckigen Klammern dargestellt.
- Der X-Y-Plotbereich kennt nun auch Textmarken und geometrische Formen. Zudem wurde sein Erscheinungsbild etwas geändert und er ist so stabil geworden, dass man ihn als Standard empfehlen kann.

Zudem wurde der Abschnitt zu Grafik mit Maxima (8.1) neu geordnet.

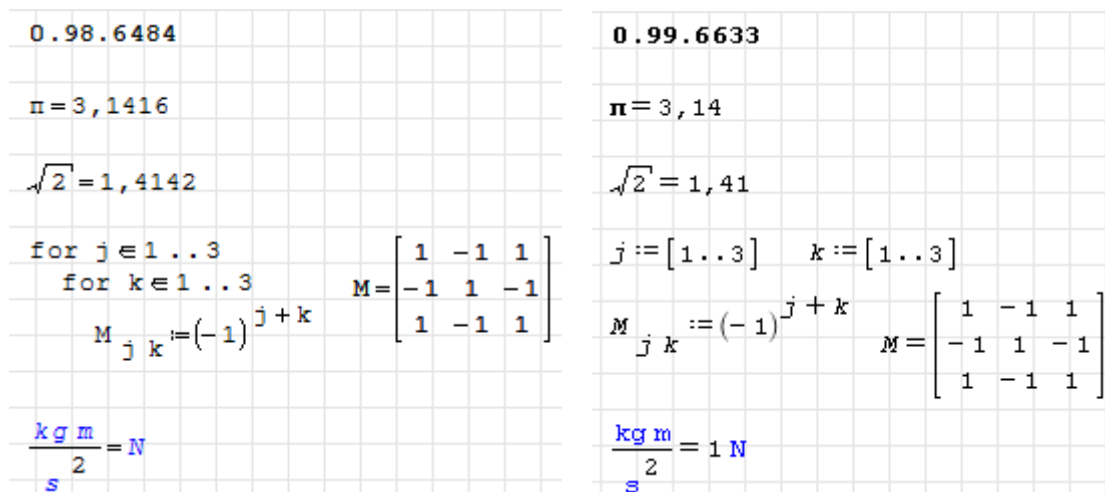


Abbildung 1.1.: Unterschiede zwischen der SMath-Version 2017 (links) und 2018 (rechts)

## Was ist SMath Studio?

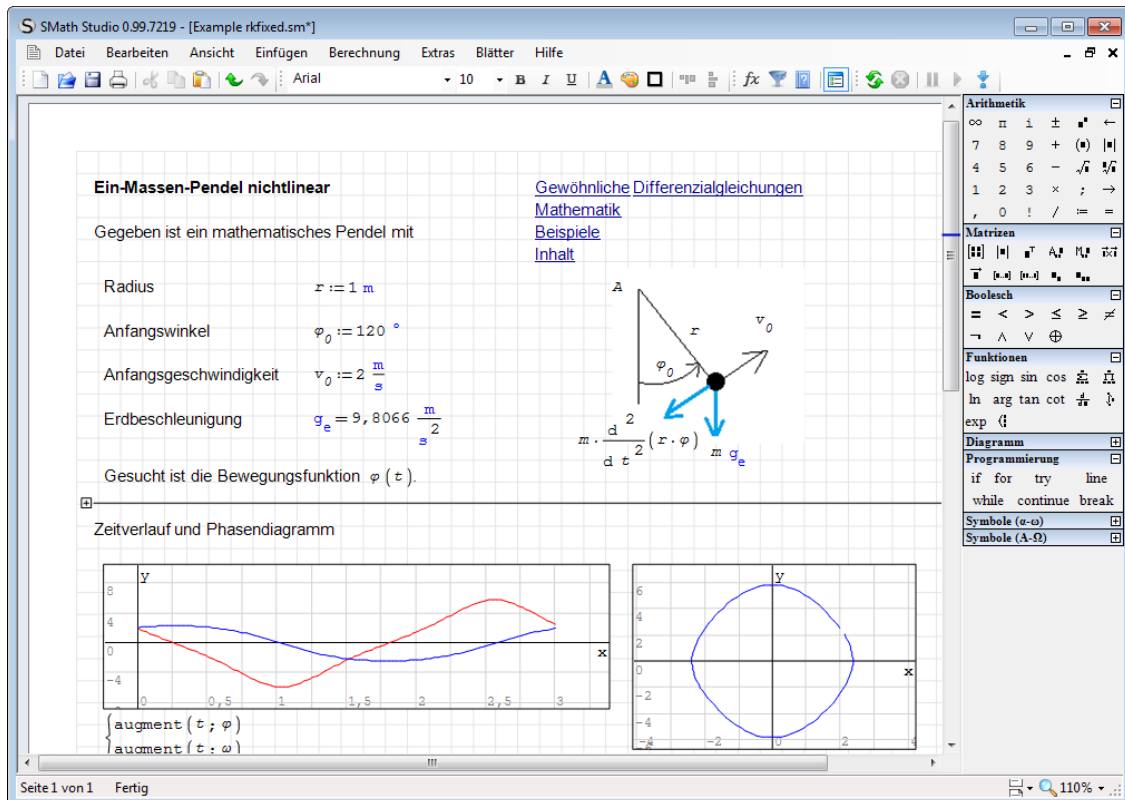


Abbildung 1.2.: Benutzeroberfläche von SMath (ein Beispiel aus dem Interaktiven Handbuch, siehe Abschnitt 4.3).

SMath Studio ist ein freies, allerdings nicht quelloffenes Programm zur Durchführung und Dokumentation technischer Berechnungen. Es stellt ein elektronisches Rechenpapier bereit, auf welchem man Formeln, Text, Diagramme und Bilder (sogenannte Bereiche) frei plazieren kann.

Nun sind Formeln in einem Programm nichts Besonderes, schon gar nicht in einem Mathematikprogramm. Das Besondere aber ist, dass diese Formeln weitgehend der gewohnten mathematischen Schreibweise mit Papier und Bleistift aus Schule und Studium folgen. Damit können auch Leser, die SMath Studio nicht kennen, die Rechnung nachvollziehen.

Das kann man bei einer Formel im Formeleditor von Textprogrammen auch, aber diese Formeln sind in der Regel nicht aktiv, sie teilen dem Rechner nicht mit, was gerechnet werden soll. Eher wird der Textschreiber den Taschenrechner zücken, die Berechnung dort durchführen und das Ergebnis ins Textprogramm eintragen. Es ist dann dem Prinzip Hoffnung überlassen, ob er mit dem Taschenrechner wirklich das gemacht hat, was die Formel behauptet.

Anders in SMath: Dort ist die Formel gleichzeitig Berechnungsvorschrift für das Programm und andererseits gut für jedermann lesbar. Was man als Formel sieht, wird auch so gerechnet. In der Textprogramm+Taschenrechner-Variante ist das nicht sicher.

In Tabellenkalkulationsprogrammen wie Excel oder LibreOffice Calc kann man auch sehr gut rechnen, aber die Formeln sieht man in der Regel nicht. Und wenn, dann ähneln sie eher Programmcode.

Programmcode ist auch das, was typische Computeralgebra-Systeme (CAS) zur Kommunikation im Frage-und-Antwort-Spiel an der Kommandozeile benutzen.

Wenn Sie Berechnungen nachvollziehbar dokumentieren und gleichzeitig rechnergestützt durchführen wollen, dann ist SMath Studio genau richtig für Sie.

SMath Studio ist nicht das erste und nicht das einzige Programm, das sich diesem Anspruch stellt. Der Klassiker ist Mathcad, der eine große Fangemeinde insbesondere im technischen Bereich hatte und hat. Dazu trug bei, dass dieses Programm wie auch SMath Studio sehr komfortabel mit Maßeinheiten umgehen kann. Maßeinheiten werden von der reinen Mathematik verpönt, helfen aber dem Praktiker ungemein, reale Probleme mit handhabbaren Zahlen zu beschreiben.

Der Mathcad-Experte wird einige Einschränkungen bei SMath bemerken:

- Weniger Formatierungsmöglichkeiten (Zahlendarstellung, keine Formatvorlagen für Textbereiche),
- Rudimentäre Dokumentation (das vorliegende Handbuch versucht hier, etwas zu helfen),
- Schwache Editierfunktionen für Matrizen, insbesondere kein Einfügen oder Löschen von Zeilen und Spalten,
- Keine Lösungsblöcke (aber leistungsfähige numerische und symbolische Löser),
- Geringerer mathematischer Funktionsumfang.

Entwickelt wird das Programm vom Sankt Petersburger Andrey Ivashov. Das Programm hat verschiedene Erweiterungsschnittstellen, über die neue Funktionen bereitgestellt werden können. Diese werden in der Regel im SMath-Forum vorgestellt und diskutiert.

## Was ist Maxima?

**Maxima** ist ein quelloffenes und freies Computer-Algebra-Programm. Es hat eine Kommandozeilen-Schnittstelle, die zugehörigen Dokumente ähneln Programmquelltexten und sind in der Regel für Außenstehende nicht verständlich. Das Programm basiert auf dem am MIT entwickelten Macsyma und wird bis heute aktiv weiterentwickelt.

In der THB-Distribution ([Download<sup>2</sup>](#)) von SMath Studio ist eine Maxima-Schnittstelle enthalten. Wenn Maxima auf Ihrem Rechner installiert ist, können Sie von SMath Studio aus darauf zugreifen.

## 1.1. Funktionsüberblick

Merkmale der Basisversion von SMath Studio:

- Numerische und symbolische Berechnungen mit reellen und komplexen Zahlen
- Symbolische Differentiation (Ableitungsbildung)
- Numerische Integration (bestimmte Integrale berechnen)
- Viele Berechnungen können mit Maßeinheiten durchgeführt werden.

---

<sup>2</sup><http://smath.info/wiki/SMath%20with%20Plugins.aspx>

- Matrizenrechnung (Addition, Multiplikation, Inversion, Determinanten), allerdings keine Eigenwerte und Eigenvektoren.
- Lösen von linearen und nichtlinearen Gleichungen und Gleichungssystemen (Nullstellensuche)
- Einfache 2D-Diagramme mit Animationsfunktion
- Programmierung mit benutzerdefinierten Funktionen, Schleifen und Anweisungsblöcken
- Dynamische Hilfe für das Einfügen von Einheiten, Variablen- und Funktionsnamen sowie Code-Bausteinen
- Eingebautes mathematisches Nachschlagewerk
- Import von ASCII-Daten

Über Plug-ins bereitgestellt werden (Auswahl):

- Computer-Algebra-Funktionen und Diagramme mit Gnuplot (*Maxima*). Im Grunde ist der gesamte Funktionsumfang von Maxima verfügbar.
- Komfortable numerische Lösungsverfahren (*Nonlinear Solvers*)
- Interaktive Bedienelemente wie Auswahllisten, Schieberegler, Schaltflächen
- Dynamische Anzeige von Bilddateien und Matrizen sowie Erstellung und Editierung von Abbildungen (*ImageRegion*)
- Formatierte Tabellen (*TableRegion*)
- Textbereiche mit komfortablen Formatierungsfunktionen (*WriterRegion*)

Prüfen Sie stets kritisch Ihre Arbeitsergebnisse. Wie jede Software enthält SMath Studio auch Fehler. Sie benutzen das Programm auf eigene Gefahr.



## 1.2. SMath im Internet

Wenn Sie nach dem Programm im Internet suchen, verwenden Sie am besten den Suchbegriff „SMath Studio“, sonst kommen zuviele irrelevante Ergebnisse. Die offizielle Seite ist [smath.com](http://smath.com), die vom Programm-Entwickler und -Besitzer Andrey Ivashov auf einem eigenen Server gehostet wird.

Es gibt Versionen für Windows, Linux, Android und iOS. Die beiden letzteren sind noch am Anfang ihrer Entwicklung und werden in diesem Handbuch nicht behandelt.



**SMath Studio** ★★★★★

Summary | License | History | Reviews | FAQ

Tiny, powerful, free mathematical program with WYSIWYG editor and complete units of measurements support. It provides numerous computing features and rich user interface translated into about 40 different languages. Application also contains integrated mathematical reference book.

Download: version 0.99.7030 - Stable (released at 2019-04-01) - recommended

|  |                            |
|--|----------------------------|
|  <b>SMath Studio Desktop</b> (2.24 MB)          | Downloads: 52038 of 727179 |
|  <b>SMath Studio Desktop for Mono</b> (1.37 MB) | Downloads: 8689 of 88328   |

GET IT ON Google Play | Download on the App Store | Get it from Microsoft

Download: version 0.99.7219 - Beta (released at 2019-10-07)

|   |                           |
|---|---------------------------|
|  <b>SMath Studio Desktop</b> (2.28 MB)         | Downloads: 1231 of 727179 |
|  <b>SMath Studio Desktop for Mono</b> (1.4 MB) | Downloads: 180 of 88328   |

Application can be easily extended based on your needs. Built-in Extensions Manager tool allows to get access to hundreds official and third-party resources of the following types: usage examples, plug-ins, SMath Viewer based applications, snippets, interface translations, interactive books, handbooks and tutorials.

Abbildung 1.3.: Webseite [smath.com](https://smath.com)

Einige nützliche Links sind im Fußteil der Hauptseite aufgeführt:

| Products                                  | Information                  | Developer tools                         | Account                  |
|---|------------------------------|---|--------------------------|
| <a href="#">SMath Studio</a>              | <a href="#">Forum</a>        | <a href="#">Sources</a>                 | <a href="#">Log in</a>   |
| <a href="#">SMath Studio in the Cloud</a> | <a href="#">Wiki</a>         | <a href="#">Release Manager</a>         | <a href="#">Register</a> |
| <a href="#">Donate</a>                    | <a href="#">Bug tracker</a>  | <a href="#">File Manager</a>            |                          |
|   | <a href="#">Translations</a> | <a href="#">Extensions registration</a> |                          |
|   | <a href="#">Statistics</a>   |   |                          |

Andrey Ivashov © 2006-2018 (на русском) | [Privacy policy](#)

Abbildung 1.4.: Webseite [smath.com](https://smath.com), Fußzeile

**Forum** Hier werden neue Entwicklungen und Beispiele vorgestellt und Fragen beantwortet. Wenn man selbst Fragen stellen oder anderen helfen will, dann muss man sich registrieren (Angabe der email reicht).

**SMath Studio in the Cloud** Online-Version, in der Sie das Programm ohne lokale Installation ausprobieren können (Screenshot in Abbildung 1.5). Es gibt Funktionen zum Herunterladen, Hochladen und statischen Verlinken („Share“) von Rechenblättern. Gegenüber dem PC-Programm gibt es allerdings Einschränkungen:

- Weniger Sprachversionen.
- Das Dezimaltrennzeichen und Argumenttrennzeichen sind nicht separat einstellbar, sondern an die Sprachversion gekoppelt. Die Einstellung „Deutsch“ setzt das Komma als Dezimaltrenner und das Semikolon als Argumenttrennzeichen.
- Keine externen Plug-ins verfügbar. Der Funktionsumfang entspricht der Basisversion ohne Erweiterungen.

- Mitunter ist der Versionsstand etwas älter.

**Wiki** Das Wiki ist eine Sammlung von Anleitungen zu einzelnen Themen, die zum Teil über den Inhalt dieser Anleitung hinausgehen. Allerdings ist nicht alles auf dem neuesten Stand.

**Sources** Hier geht es zum SVN-Server, auf dem Quelltext und zum Teil auch kompilierte Versionen externer Plug-ins liegen. Dieser Bereich ist für Nutzer interessant, die an der Entwicklung mitarbeiten wollen, siehe dazu auch Abschnitt 12.4. An den Programmcode des SMath-Kerns kommt man aber nicht heran, denn SMath ist kein quelloffenes Programm.

**Translations** Hier kommt man in den Bereich der Sprachanpassung. Nutzer können Vorschläge für Texte in der Benutzeroberfläche machen. Dieser Mechanismus macht es möglich, dass SMath in sehr vielen Sprachversionen verfügbar ist.

**Bug tracker** Seit März 2014 gibt es ein System zur Verwaltung von Fehlermeldungen und Anwenderwünschen (*bug tracking system*). Um dort Inhalte einzustellen, benötigt man ein separates Login. Die Liste von Fehlermeldungen und Anwenderwünschen ist in folgende Projekte unterteilt:

- *SMath Studio* (das Programm zur Erstellung von Rechenblättern)
- *SMath Studio in the Cloud* (die Online-Version)
- *SMath Viewer* (Ausführung kompilierter Rechenblätter als eigenständige Anwendung, Details dazu im Kapitel 11)

Häufig werden Fehlermeldungen auch im Forum gepostet, dort kann man sie aber schnell aus den Augen verlieren.

The screenshot shows the SMath Studio online interface. The main window displays a calculation sheet with the following data and formulas:

|   |   |                 |                        |
|---|---|-----------------|------------------------|
| Length  | $l := 100 \text{ mm}$   | Bending angle   | $\alpha := 90^\circ$   |
| Thickness   | $h := 1 \text{ mm}$   | Young's modulus | $E := 210 \text{ GPa}$ |
| Width   | $b := 10 \text{ mm}$  | Poisson's ratio | $\nu := 0,3$           |
| Radius of curvature                               | $\rho := \frac{l}{\alpha} = 6,366 \text{ cm}$                   |                 |                        |
| Long. strain at the surface                       | $\epsilon := \frac{h}{2 \cdot \rho} = 0,7854 \%$                |                 |                        |
| Longitudinal stress (plane stress, at the edges)  | $\sigma := E \cdot \epsilon = 1649 \text{ MPa}$                 |                 |                        |
| Longitudinal stress (plane strain, in the center) | $\sigma := \frac{E}{1 - \nu} \cdot \epsilon = 1812 \text{ MPa}$ |                 |                        |
| Moment of inertia                                 | $I := \frac{1}{12} \cdot b \cdot h^3 = 0,8333 \text{ mm}^4$     |                 |                        |
| Bending moment                                    | $M := \frac{E \cdot I}{\rho} = 2749 \text{ N mm}$               |                 |                        |
| Strain energy                                     | $\bar{W} := \frac{1}{2} \cdot M \cdot \alpha = 2159 \text{ mJ}$ |                 |                        |

A diagram on the right shows a beam of length  $l$  fixed at the left end. A bending moment  $M$  is applied at the right end, causing the beam to bend at an angle  $\alpha$ . The beam has thickness  $h$  and width  $b$ . The coordinate system  $(x, z)$  is shown with  $x$  along the beam's length and  $z$  perpendicular to it.

Abbildung 1.5.: Online-Version von SMath, Link zu diesem Rechenblatt: <https://en.smath.com/cloud/worksheet/DwQ2BnBB>

## 1.3. SMath unter Linux

Die **THB-Distribution**<sup>3</sup> ist grundsätzlich auch unter Linux lauffähig, dies wurde vom Autor unter Linux Mint 17.2 mit Mono 4.2.1 getestet. **Mono**<sup>4</sup> ist eine plattformübergreifende, quelloffene .NET-Umgebung. Die Linux-Version bringt ein Shell-Skript *SMathStudio\_Desktop\_Mono* mit, welches den Befehl

```
mono SMathStudio_Desktop.exe
```

ausführt. Dabei werden eventuelle Parameter (zu öffnende Dateien) durchgereicht.

Die Maxima-Schnittstelle und einige andere Plug-ins funktionieren unter Linux nicht.

## 1.4. Alternativen

Hier werden einige Mathematikprogramme aufgeführt, deren Funktionsumfang sich mit dem von SMath Studio überschneidet.

**Mathcad**<sup>5</sup> ist das Original, von dem die Entwicklung von SMath Studio inspiriert ist. Die Jahreslizenz kostet 93,30€ für Studenten und 625€ für kommerzielle Anwendung (Stand Oktober 2019). Es gibt auch eine kostenlose Version „Mathcad Express“, die auf die Basisfunktionen beschränkt ist.



**MathStudio**,<sup>6</sup> verfügbar für iOS und OS-X (\$20) und als kostenlose Online-Version. Dies ist eine kommandozeilenorientierte Anwendung mit großem Funktionsumfang und guten Grafikfunktionen. Sie kann als Ersatz für einen guten CAS-Taschenrechner gelten.



Vorteil:

- Für Nutzung auf Smartphones geeignet
- Umfangreiche Grafik- und Rechenfunktionen

Nachteile:

- kein 2D Dokumentformat

**TI Nspire CAS**<sup>7</sup> Taschenrechner mit Computeralgebrasystem, basierend auf dem nicht mehr erhältlichen Derive. Das System gibt es auch als Software für PCs und als iOS-App.

Vorteile:

- Besserer Funktionsumfang und Reifegrad
- Interaktive Grafik (Geometrie und Diagramme)

Nachteile:

- kein 2D Dokumentformat.

<sup>3</sup><http://smath.info/wiki/SMath%20with%20Plugins.ashx>

<sup>4</sup><http://mono-project.com>

<sup>5</sup><https://www.ptc.com/de/products/mathcad>

<sup>6</sup><http://mathstud.io>

<sup>7</sup><https://education.ti.com/de/products/calculators/graphing-calculators/ti-nspire-cx-cas>



**Maxima**<sup>8</sup> ist das beste quelloffene Computeralgebrasystem derzeit.



Vorteile:

- Quelloffen, aktive Benutzer- und Entwicklergemeinschaft
- Besserer Funktionsumfang und Reifegrad

Nachteile: kein 2D Dokumentformat.

Mit der SMath Studio Schnittstelle zu Maxima werden die Vorteile genutzt und der Nachteil ausgeglichen.

**Giac/Xcas**<sup>9</sup>: Giac ist eine quelloffene Bibliothek für CAS-Funktionen, Xcas ist eine Benutzeroberfläche dafür. Die Bibliothek ist Grundlage der CAS-Funktionen des Taschenrechners HP50 und in Geogebra. Es gibt auch eine im Browser laufende **Online-Version** .



Vorteile (nur im Vergleich zu SMath Studio ohne Maxima):

- Quelloffen, aktive Benutzer- und Entwicklergemeinschaft
- Besserer Funktionsumfang und Reifegrad

Nachteile: kein 2D Dokumentformat.

**microMathematics** (**μMath**)<sup>10</sup> Kommandozeilenorientierter Grafik-Taschenrechner als Android-App, entwickelt von Mikhail Kulesh (Bremen). Einen Überblick über Konzept und Lösungsansatz bietet ein **russischer Artikel**<sup>11</sup>



Vorteile:

- natürliche Formelschreibweise bei Ein- und Ausgabe,
- reichhaltiger Funktionsumfang,
- quelloffen (eine erweiterte kommerzielle Version gibt es auch)
- μMath-Dateien können in SMath importiert werden (siehe **Anleitung im μMath-Wiki**),

Nachteile:

- läuft nur unter Android,
- Eingeschränktes 2D-Dokumentenformat

**Miramath**<sup>12</sup> Ein Nachbau von MathCAD, basierend auf numerischen und symbolischen Python-Bibliotheken. Für den Download muss man sich registrieren. Das Programm wurde vom Autor dieser Anleitung nicht getestet. Nur unter Linux verfügbar.

Vorteile:

- 2D Dokumentformat
- Hochwertiger Formelsatz

Nachteile:

<sup>8</sup><http://maxima.sourceforge.net/de/index.html>

<sup>9</sup><http://www-fourier.ujf-grenoble.fr/~parisse/giac.html>

<sup>10</sup>[github.com/mkulesh/microMathematics](https://github.com/mkulesh/microMathematics)

<sup>11</sup>[habrahabr.ru/post/250727/](http://habrahabr.ru/post/250727/)

<sup>12</sup><http://miramath.newfrontierscompsoln.com/>

- Nicht quelloffen
- Keine Maßeinheitenbehandlung
- Entwicklungsintensität unklar (letzte Änderung auf der Website von 2011)

**iMath**<sup>13</sup> ist eine LibreOffice-Erweiterung, die Formeln aktiv auswertet. Sie befindet sich in aktiver Entwicklung und hat vom Konzept her das Potenzial zu einer starken Konkurrenz zu SMath Studio.



Vorteile:

- Direkte Integration in LibreOffice
- Maßeinheitenbehandlung
- Symbolische Funktionen

Nachteile:

- Umständliche Formeleditierung

**Microsoft Mathematics**<sup>14</sup> 4.0 gibt es als eigenständige Anwendung und als Add-In für den Formeleditor in Microsoft Word.



Vorteile:

- Nahtlose Integration in Microsoft Word (Add-in)

Nachteile:

- keine formelübergreifende Definition von Funktionen und Variablen (Add-In)
- Keine Maßeinheiten in Formeln

**RedCrab**<sup>15</sup> ist ein Mathematikprogramm, welches ähnlich wie Mathcad und SMath ein 2D Dokumentenformat bietet. Es ist Shareware (65\$, Februar 2019), nicht quelloffen und es gibt eine funktionsreduzierte freie Version. Im Unterschied zu SMath Studio und Mathcad werden Ergebnisse nicht in der Reihenfolge ihrer Anordnung auf dem Blatt sondern gemäß der Verwendungshierarchie berechnet.



Vorteile:

- 2D-Dokumentformat
- Maßeinheitenverwaltung (nicht in der freien Version)
- Handbuch, Einführungsvideos

Nachteile:

- keine symbolischen Berechnungen

## 1.5. Literatur

Es gibt viele Bücher zum Einsatz von Mathcad und dialogorientierten Computeralgebra-Systemen für technische Berechnungen.

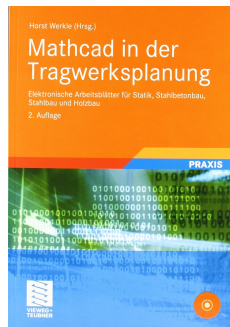
<sup>13</sup><http://ooo-imath.sourceforge.net>

<sup>14</sup><http://www.microsoft.com/de-de/download/details.aspx?id=15702>

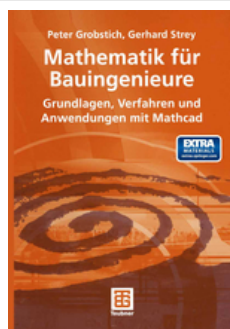
<sup>15</sup><http://redchillicrab.com/en/redcrab/>



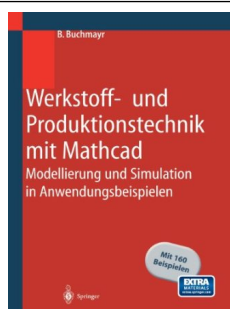
Henning/Jahr/Mrowka (2004) Entstanden an der Fachhochschule Düsseldorf zu einem Wahlpflichtkurs „Angewandte Mechanik“.



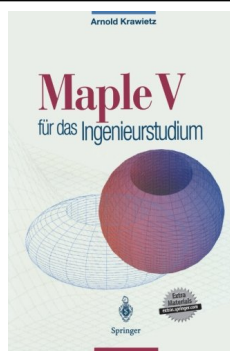
Wörke (2012) Der Herausgeber unterrichtet Baustatik an der HTWG Konstanz.



Grobstich/Strey (2004) Die Autoren waren an der Fachhochschule Neubrandenburg tätig.



Buchmayr (2002) Numerische Verfahren bis hin zu einem FEM-Programm für Scheiben und einem Werkstoffdatenbank-System, Materialdatenanpassung, Prozessmodelle



Krawietz (1997) Mathematisch sehr weitreichende Behandlung mit Beispielen aus der Mechanik. Der Autor hat an der Technischen Fachhochschule Berlin Mechanik unterrichtet.



Liengme (2015) Erstes und einziges SMath-Buch auf dem Markt. Die Beispiele aus der Physik sind als Ergänzung zum vorliegenden Handbuch gut geeignet.

## 2. Installation

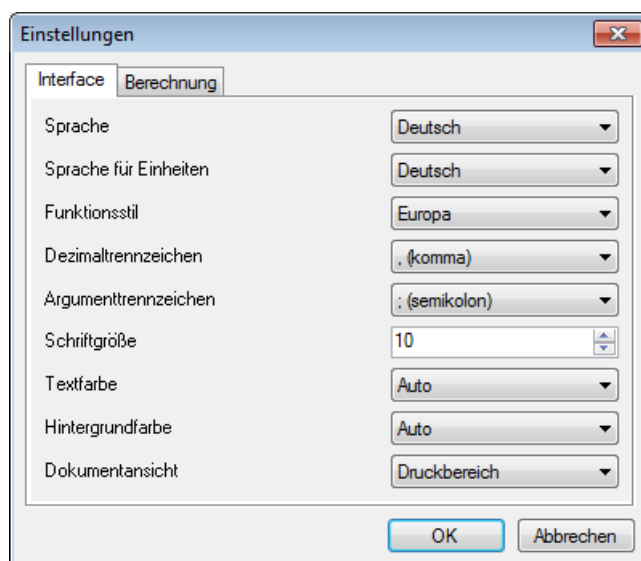
Hier wird beschrieben, wie Sie SMath Studio installieren und so konfigurieren, dass Sie das vorliegende Handbuch nachvollziehen können. Es gibt zwei Optionen, beide werden weiter unten im Detail erläutert:

**Standardinstallation** Dafür wird der offizielle Windows-Installer verwendet, der allerdings nur die Basisversion installiert. Einige Einstellungen sind dann noch anzupassen und Erweiterungen nachzuladen. Die Erweiterungen und Updates werden in einem Benutzerverzeichnis auf dem Rechner gespeichert.

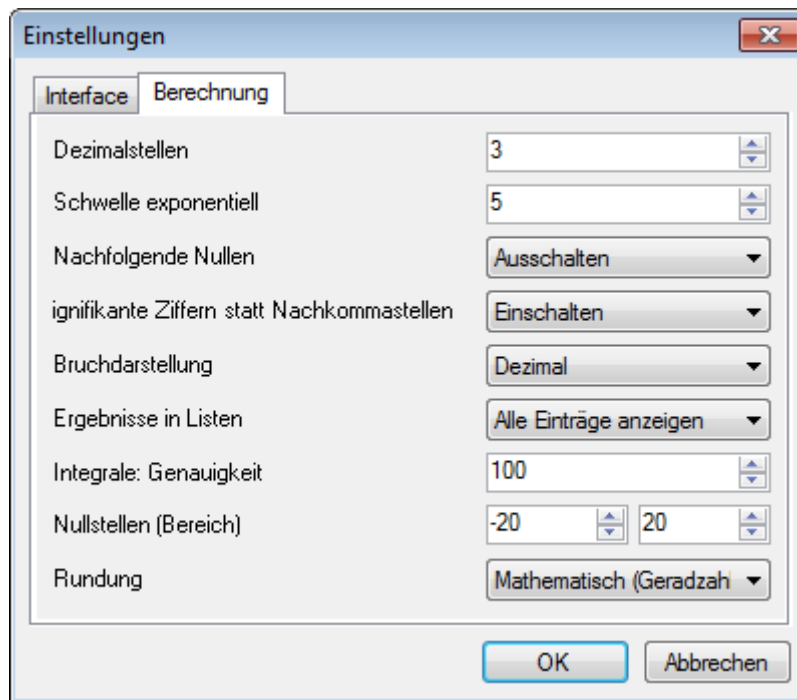
**Portable THB-Distribution** Diese Distribution kann von einem beliebigen Verzeichnis aus gestartet werden, z.B. auch auf einem USB-Stick. Einstellungen und Erweiterungen werden relativ zu diesem Verzeichnis gespeichert.

### 2.1. Standardinstallation (Windows)

1. Download des Installers für die Windows-Basisversion (MSI-Datei) von der [SMath-Homepage](#)<sup>1</sup>. Er ist kaum größer als 2 MB. Die später nachzuladenden Erweiterungen können insgesamt mehrere 100 MB einnehmen.
2. Ausführen des Installers.
3. Starten des Programms mit der Schaltfläche **S**.
4. Sprache auf Deutsch stellen: **Tools > Settings: Language: German**
5. Weitere Einstellungen anpassen: **Extras > Einstellungen:**



<sup>1</sup>[en.smath.com/view/SMathStudio/summary](http://en.smath.com/view/SMathStudio/summary)



Die Einstellungen besagen:

- Dezimalzahlen werden mit Komma geschrieben, also 0,33 statt 0.33.
- Argumente in Funktionen werden mit Semikolon getrennt, also  $f(x; y)$
- 3 Dezimalstellen sollen angezeigt werden. Diese Genauigkeit ist für technische Berechnungen meist sinnvoll.
- Ab dem Wert  $10^5$  werden Zahlen in Exponentialschreibweise dargestellt.
- Nachfolgende Nullen an Dezimalbrüchen werden weggelassen.
- Die Dezimalstellen geben die Mantissenlänge (Zahl der signifikante Ziffern) und nicht die der Nachkommastellen an.
- Numerisch berechnete Ergebnisse werden möglichst als Dezimalbruch dargestellt (bei symbolisch berechneten Ergebnissen wird immer die Bruchdarstellung verwendet).
- Rundung erfolgt mathematisch (wenn auf die letzte zu erhaltende Stelle nur noch eine 5 folgt, dann wird zur nächstliegenden Geradzah gerundet).

6. Erweiterungen nachladen:

- a) **Extras** > **Plugins** öffnet die Erweiterungsverwaltung.
- b) Rechts oben von „Lokale Speicherung“ auf „Online-Galerie“ umschalten

Nun können Sie Plug-ins, PDF-Handbücher und interaktive Handbücher installieren:

- a) Kategorie „Handbücher“: SMATH Studio Handbuch (das vorliegende Dokument)
- b) Kategorie „Interaktive Handbücher“: Interaktives SMATH Handbuch. Setzen Sie unten den Haken bei „Standardmäßig nutzen“

7. Optional: Wählen Sie Ihr Lieblings-Zeichenprogramm (z.B. Paint) als Standardprogramm für PNG-Bilddateien. Dann können Sie aus SMATH Studio heraus Bilder erzeugen und editieren. Wahlweise können Sie auch SVG-Dateien (Vektorgrafik) erzeugen und z.B. mit Inkscape editieren.

## 2.2. Portable THB-Distribution

Diesem Handbuch liegt die **THB-Distribution**<sup>2</sup> von SMath Studio mit Maxima zugrunde. Vorteile gegenüber der offiziellen SMath-Distribution sind:

- Vorinstallation der meisten externen Plug-ins,
- Vorinstallation des interaktiven Handbuchs mit ca. 500 verlinkten bilingualen (deutsch/englisch) Hilfeseiten,
- Portable Installation (z.B. auf einem USB-Stick).

Schritte für die Installation:

- Entpacken Sie die Datei in einem beliebigen Speicherort. Dann finden Sie dort die ausführbare Datei *SMathStudio\_Desktop.exe* und einige andere Dateien und Verzeichnisse. Damit können Sie schon mal loslegen.
- Optional: Erklären Sie die ausführbare Datei *SMathStudio\_Desktop.exe* zum Standardprogramm für .sm Dateien. Dann können Sie im Dateimanager SMath-Dateien mit Doppelklick öffnen.
  - Rechtsklick auf *welcome.sm* (oder irgendeine andere .sm-Datei)
  - Kontextmenü > Öffnen mit > Standardprogramm auswählen
  - und *SMathStudio\_Desktop.exe* im Installationsverzeichnis auswählen.
  - Wenn Sie SMath Studio auf einem USB-Stick installiert haben, müssen Sie diese Zuordnung auf dem jeweiligen Rechner neu ausführen.
- Optional: Wählen Sie Ihr Lieblings-Zeichenprogramm (z.B. Paint) als Standardprogramm für PNG-Bilddateien. Dann können Sie aus SMath Studio heraus Bilder erzeugen und editieren. Wahlweise können Sie auch SVG-Dateien (Vektorgrafik) erzeugen und z.B. mit Inkscape editieren.
- Stellen Sie die Sprache der Benutzeroberfläche auf deutsch (voreingestellt ist englisch):
  - Tools > Options > Interface language „Deutsch“** (Extras > Einstellungen > Sprache)

## 2.3. Linux

The following instruction was tested with Kubuntu 19.10 in 2020/01. The Maxima plug-in doesn't work.

### Installation from Debian package

- Install Mono:

```
>sudo apt-get install mono-complete
```

- Download and install SMath Studio debian package from [Sourceforge](https://sourceforge.net/projects/smath-studio/). By 01/2020 the newest version is 0.99.7202.

<sup>2</sup><https://smath.com/wiki/SMath%20with%20Plugins.ashx>

**Portable installation from [smath.com](http://smath.com):**

- Install Mono:

```
>sudo apt-get install mono-complete
```

- Download SMath Studio Desktop for Mono (either stable or beta release). You get a zipped TAR archive named *SMathStudioDesktop.x\_yy\_zzzz.Mono.tar.gz*
- Create an installation directory and unpack the archive there.
- Create an empty file named *portable.version*. This indicates that the installation directory is also the settings directory, where settings and extensions are stored. This way, you can have multiple installations in different directories side by side.

```
>touch portable.version
```

- In the installation directory, run

```
>./smathstudio_desktop_mono-
```

**Setup of the environment:** For creating and editing sketches from within image regions, you have to set the application for PNG or SVG images.

**SVG**

**Boxy SVG** Opens with an untitled new SVG file (without error message)

**Inkscape** (the free standard for vector graphics) Can't handle mixed path separators generated by image region. If files are copied elsewhere, then they can be opened but Inkscape complains about wrong resolution.

**Karbon** works, but does not save without asking for filename and permission to overwrite.

**PNG**

**Gnu Paint** (similar to MS Paint). No undo function, images get black background by default.

**KolourPaint** (fast and simple, much like MS Paint.) Version 19.04.3 works. The background of new images becomes black as soon as the first edit is done. You have to fill the background with white color first.

**PDF**

**IPE** Kann die erzeugte temporäre Datei nicht öffnen, da sie nicht mit IPE erzeugt wurde.



## 3. Erste Schritte

Dieses Kapitel wurde bereits im Unterricht an der Technischen Hochschule Brandenburg erprobt und kann in ein bis zwei Stunden durchgearbeitet werden.

### 3.1. Speichern und Rückgängig machen

SMath speichert Rechenblätter als XML-Dateien mit der Endung .sm. Legen Sie zu Beginn der Arbeit mit **Datei > Speichern unter** den Dateinamen fest und speichern Sie regelmäßig (am bequemsten mit **Strg-S**, das heißt: **Strg** und **S** gleichzeitig drücken).

Rückgängig machen per **Strg-Z** und Wiederherstellen per **Strg-Y** funktionieren in den meisten Fällen, aber maximal bis zum letzten Speicherpunkt.

Rückgängig machen **Strg-Z** ist die beste Korrektur für Tippfehler. Die Rücktaste entfernt in einigen Fällen mehr als nur das letzte Zeichen.

### 3.2. SMath als Taschenrechner

Wenn Sie einfach lostippen, entsteht an der Stelle der Einfügemarke + ein Formelbereich. In Formelbereichen können Sie Berechnungen durchführen und Funktionen und Variablen definieren. In Formelbereichen wird eine Schrift verwendet, deren Zeichen alle gleich breit sind (Schreibmaschine).

Beispiele:

| Tastatureingabe         | Ergebnis          |
|-------------------------|-------------------|
| 3+3=                    | $3 + 3 = 6$       |
| 3^3=                    | $3^3 = 27$        |
| cos(p <b>Strg-G</b> ) = | $\cos(\pi) = -1$  |
| \3=                     | $\sqrt{3} = 1,73$ |

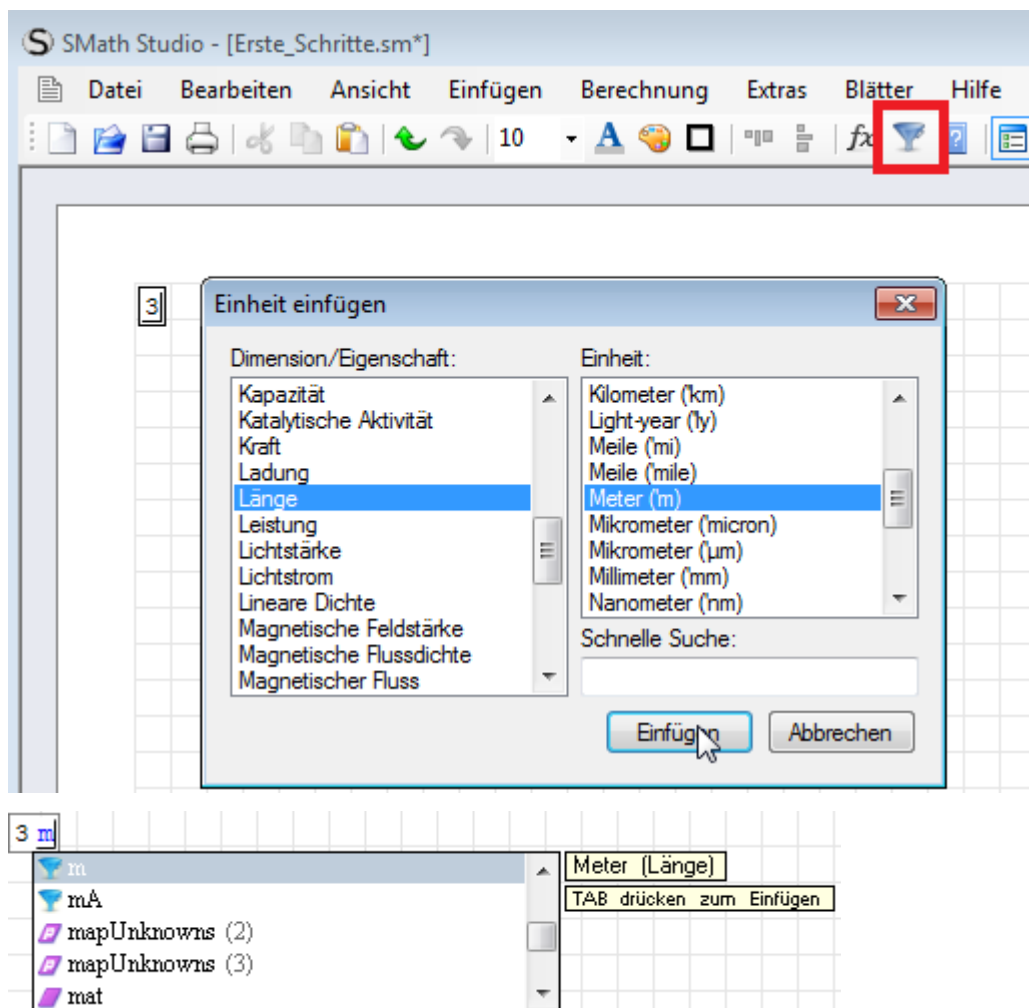
Die Zahl der angezeigten Nachkommastellen kann geändert werden, darauf gehen wir später noch ein.

In den Platzhalter ■ hinter dem Ergebnis kann eine Maßeinheit oder sonst ein Faktor, den man im Ergebnis haben will, eingetippt werden. Ansonsten verschwindet er, wenn man den Formelbereich verlässt.

|           |                     |             |           |
|-----------|---------------------|-------------|-----------|
| 3 + 3 = 6 | 3 <sup>3</sup> = 27 | cos(π) = -1 | √3 = 1,73 |
|-----------|---------------------|-------------|-----------|

SMath kann mit Maßeinheiten rechnen. Der Dialog zum Einfügen von Maßeinheiten wird über die Werkzeugleiste oder  $\text{Strg} - \text{W}$  aufgerufen.

3



Maßeinheiten sind spezielle vordefinierte Konstanten, die man in Formeln an der blauen aufrechten Schrift erkennt. Bei der Eingabe von Namen zeigt die dynamische Hilfe an, welche bekannten Namen von Variablen, Maßeinheiten und Funktionen zur Eingabe passen würden.

Sie können weitere Buchstaben eintippen oder mit den Pfeiltasten aus der Liste auswählen. Mit  $\text{↵}$  (Tabulator) wird bestätigt.




Intern steht vor dem Einheitennamen ein Hochkomma ( $'$ ), wie man auch im Auswahldialog sieht. Das können Sie auch bei der Eingabe benutzen, um Maßeinheiten zu kennzeichnen und von gleichnamigen Formelzeichen (z.B.  $m$  für Meter und  $m$  für die Masse) zu unterscheiden.

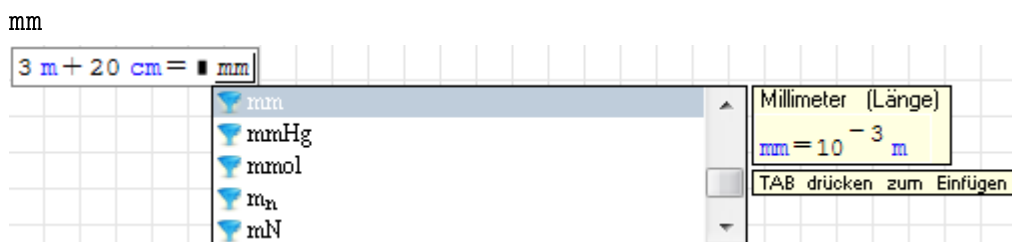
Setzen Sie die eben begonnene Formel fort:

$$+20\text{cm} \text{ ↵} =$$


$$\boxed{3 \text{ m} + 20 \text{ cm} = 3,2 \text{ m}}$$

Die interne Darstellung dieses Ausdrucks ist  $3 * 'm + 20 * 'cm = 3.2 * 'm @ \#$

Um das Ergebnis in mm anzuzeigen, stellen Sie die Eingabemarke auf den Platzhalter hinter dem Ergebnis und tippen mm gefolgt von  :



$3 \text{ m} + 20 \text{ cm} = 3200 \text{ mm}$

Sie können aber auch dabei wieder den Einheiten-Dialog benutzen. Die Berechnung erfolgt erst nach Verlassen des Formelbereichs ( drücken oder irgendwo anders im Rechenblatt klicken, dann verschwindet der Rahmen um die Formel):

Formeln und alle anderen Bereiche können frei auf dem Rechenblatt angeordnet werden. Die Auswertung erfolgt immer von links nach rechts und von oben nach unten. Dabei zählt die linke obere Ecke des Bereichs.


### 3.3. Namen

Funktionen oder Variablen können einen Namen bekommen, der mit dem Zuweisungsoperator  $:=$  definiert wird. Dieser ist einfach als Doppelpunkt einzutippen. Namen werden immer (spätestens nach Verlassen der Formel) *kursiv* geschrieben, außer bei vordefinierten Namen (z.B. von eingebauten Funktionen und Maßeinheiten).

Groß- und Kleinschreibung zählt!  $X$  und  $x$  sind verschiedene Namen!




Definitionen sind immer unterhalb und rechts ihres Bereiches bekannt (dabei zählt die linke obere Ecke als Position des Bereichs).

$X: 3\text{m}$  

$X := 3 \text{ m}$



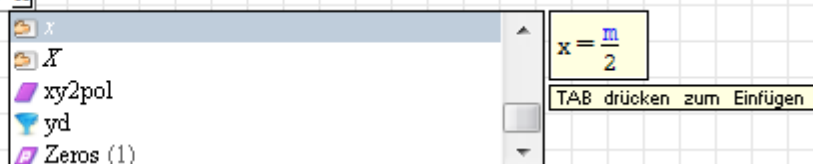
$X := 3 \text{ m}$

$x: 50\text{cm}$  

$X := 3 \text{ m}$     $x := 50 \text{ cm}$

$c: a+b$

$X := 3 \text{ m}$     $x := 50 \text{ cm}$     $c := X + x$



=  
 $X := 3 \text{ m}$     $x := 50 \text{ cm}$     $c := X + x = 3,5 \text{ m}$

Die dynamische Hilfe kennt alle bereits definierten Namen (Variablen, Funktionen, Maßeinheiten). Der Wert wird in einer symbolisch vereinfachten Form angezeigt. So kann man erkennen, was genau SMATH als Definition gespeichert hat.

Den Zahlenwert einer Variablen zeigt man mit „=“ an, entweder wie eben gleich bei der Definition oder in einer extra Formel. Natürlich kann man dabei auch wieder die Maßeinheit wählen.

$c = \rightarrow$

$c = \text{ft}$

|            |   |
|------------|---|
| ft         | Fuß (Länge)                                 |
| furlong    | $\text{ft} = 3,048 \cdot 10^{-1} \text{ m}$ |
| fwrite (2) | TAB drücken zum Einfügen                    |
| fwrite (3) |   |
| g          |   |

$\leftarrow$

$X := 3 \text{ m}$     $x := 50 \text{ cm}$     $c := X + x$     $c = 11,5 \text{ ft}$

Namen können auch mit einem tiefgestellten Namenszusatz (Textindex) versehen werden. Die Tiefstellung wird mit einem Punkt (.) eingeleitet, der als unsichtbares Zeichen Bestandteil des Namens bleibt. Die folgenden Eingaben sind jeweils in einem neuen Bereich zu machen.

F. x: 2N    $\leftarrow$     $\downarrow$

F. y: 4N    $\leftarrow$     $\downarrow$

F: \F. x^2   Leert.   +F. y^2=    $\downarrow$

$F_x := 2 \text{ N}$

$F_y := 4 \text{ N}$

$F := \sqrt{F_x^2 + F_y^2} = 4,47 \text{ N}$

Verwenden Sie möglichst aussagefähige Variablenamen entsprechend dem mathematischen Gebrauch Ihres Fachgebiets.

Griechische Buchstaben erzeugen Sie mit den Funktionspaletten auf der Seitenleiste. Alternativ können Sie nach einem lateinischen Buchstaben  $\text{Strg}-\text{G}$  drücken. Eine Übersicht über die Zuordnung finden Sie in Tabelle 6.1.

p    $\text{Strg}-\text{G}$    =    $\downarrow$

$\pi = 3,14$

**Überflüssige Klammern löschen:** Es kann passieren, dass in Formeln überflüssige Klammern stehen. Das schadet nicht der Rechnung, sieht aber mitunter nicht gut aus. Hier ein Beispiel:

$$\frac{(a+b)}{c}$$

Klammern können nur gemeinsam mit ihrem Inhalt gelöscht werden. Daher muss man vorher den Inhalt kopieren.

1. Klammerinhalt markieren (nur den Inhalt, nicht die äußere Klammer)

$$\frac{(a+b)}{c}$$

2. Ausschneiden mit **Strg**-**X**

$$\frac{(a)}{c}$$

3. Verbleibende Klammer löschen, z.B. mit **←** **←** **←**

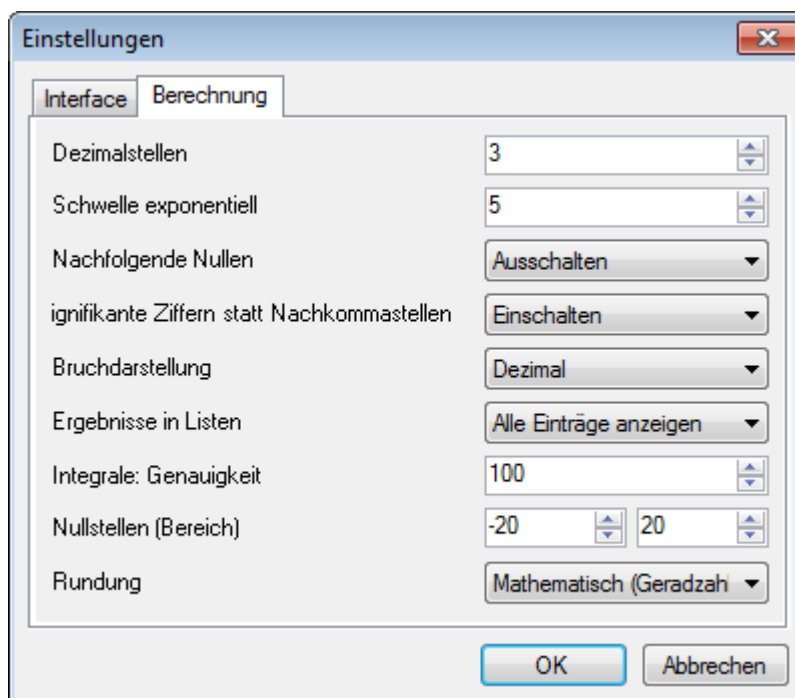
$$\frac{a}{c}$$

4. Inhalt wieder einfügen mit **Strg**-**V**

$$\frac{(a+b)}{c}$$

## 3.4. Ergebnisdarstellung und Runden

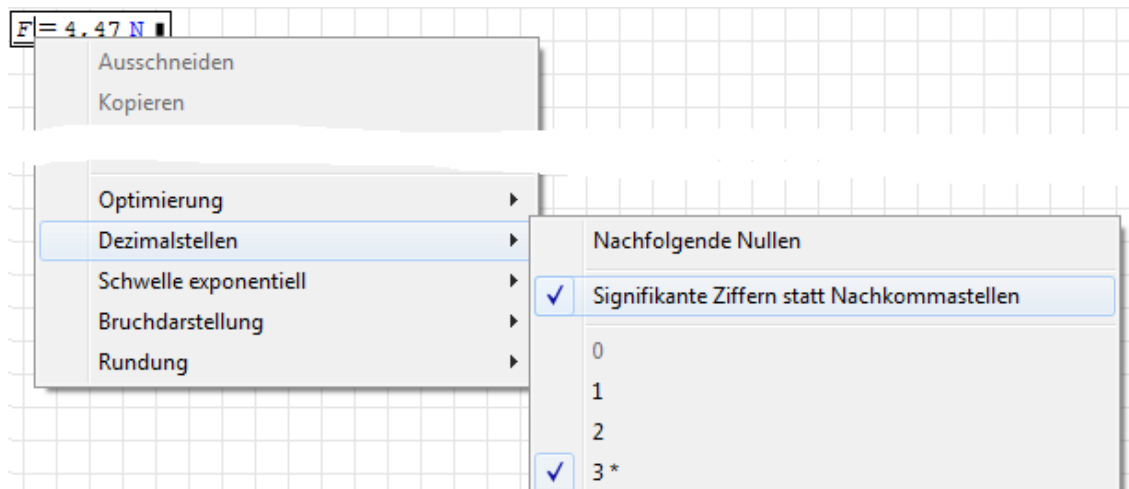
Die Ergebnisanzeige kann im Kontextmenü (rechte Maustaste) für jede Formel einzeln oder als globale Voreinstellung unter **Extras** > **Einstellungen** > **Berechnung** beeinflusst werden.



Die Einstellungen besagen:

- 3 Dezimalstellen sollen angezeigt werden. Diese Genauigkeit ist für technische Berechnungen meist sinnvoll.
- Ab dem Wert  $10^5$  werden Zahlen in Exponentialschreibweise dargestellt.
- Nachfolgende Nullen an Dezimalbrüchen werden weggelassen.
- Die Dezimalstellen geben die Mantissenlänge (Zahl der signifikante Ziffern) und nicht die der Nachkommastellen an.
- Zahlen werden möglichst als Dezimalbruch dargestellt.
- Rundung erfolgt mathematisch (wenn auf die letzte zu erhaltende Stelle nur noch eine 5 folgt, dann wird zur nächstliegenden Geradzahl gerundet).

Rechtsklick auf einen Formelbereich öffnet das Kontextmenü:



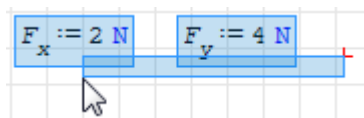
Hier wurde der Wert „Dezimalstellen“ auf 5 geändert:

$$F = 4,4721 \text{ N}$$

## 3.5. Kopieren und Verschieben

Bereiche können frei auf dem Rechenblatt angeordnet werden. Die Auswertung erfolgt immer von links nach rechts und von oben nach unten (wobei die linke obere Ecke der Bezugspunkt des Bereichs ist).

Zum Kopieren, Verschieben oder Löschen ganzer Formeln zieht man mit der Maus ein Rechteckfenster über die zu markierenden Bereiche. Sie erscheinen dann in Hellblau:



Mit **Strg**-**C** kopiert man die markierten Bereiche und kann sie mit **Strg**-**V** anderswo im Rechenblatt einfügen. So kopierte Bereiche kann man außerhalb von SMATH Studio (z.B. in Word oder Powerpoint) als Bild einfügen.

Will man Teile einer Formel kopieren oder löschen, klickt man zunächst in die Formel und zieht dann mit der Maus eine Markierung auf:

$$F := \sqrt{F_x^2 + F_y^2} = 4,47 \text{ N}$$

Mit **Strg**C kopiert man den markierten Bereich. Er kann auch außerhalb von SMath als Text eingefügt werden, so sieht man, wie er intern dargestellt wird:

F.x<sup>2</sup>+F.y<sup>2</sup>

Kopierte Teilausdrücke erzeugen beim Einfügen ins Rechenblatt einen neuen Bereich. Einfügen in eine Formel ersetzt den gerade markierten Bereich oder füllt den Platzhalter an der Cursorposition aus.

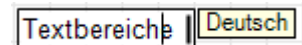
## 3.6. Textbereiche

Das Rechenblatt kann mit Kommentaren versehen werden. Dafür gibt es den (Standard-)Textbereich und einen Erweiterten Textbereich (*writer region*). Dieser entspricht schon eher dem, was man von der Absatz- und Zeichenformatierung einer Textverarbeitung erwartet (Abschnitt 5.6).



Seit der Version 0.99.7200 ist der Textbereich drastisch leistungsfähiger, man kann nun auch Formelbereiche darin einfügen. Man erzeugt ihn durch Eintippen von Gänsefüßchen "



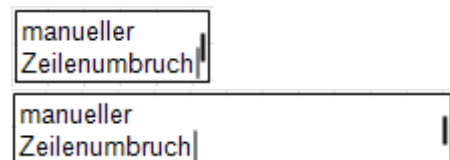
"  


Man kann auch einfach im Rechenblatt lostippen, am ersten Leerzeichen (nach den ersten Buchstaben) merkt SMath, dass man keine Formel, sondern Text schreibt. Dann erscheint die dynamische Sprachanzeige und die Schriftart springt auf Arial:

Textbereiche□  


Am vertikalen Balken ganz rechts kann man mit der Maus ziehen und die Breite des Bereichs verändern. Dabei werden Zeilen umgebrochen, eine Silbentrennung erfolgt allerdings nicht.

Ein manueller Zeilenumbruch kann mit  -  eingefügt werden. Er bleibt erhalten, auch wenn er bei der gegebenen Breite des Bereichs nicht nötig wäre:

Manueller□  -  Zeilenumbruch  


**Textbereiche können nur als Ganzes formatiert werden.** Am schnellsten erreichbar sind die Formatierungen **fett**, unterstrichen und *kursiv*, die mit **Strg**-Befehlen ein- und ausgeschaltet werden:

Normal    **Strg-b**    Strg-u    *Strg-i*

Hinweis: Wenn das nicht funktioniert, haben Sie möglicherweise einen Formelbereich erzeugt. Wenn Sie ein Wort, z.B. **Strg** tippen und dann ein Minus, dann denkt SMath, dass Sie eine Formel schreiben wollen. Dann nutzt auch ein Leerzeichen nicht mehr. Dann

müssen Sie mit Strg-Z rückgängig machen oder einen neuen Bereich anfangen, hier z.B. mit

"Strg-b

Strg-b

Der Unterschied zum Formelbereich ist aber leicht zu erkennen, denn die die Schrift in Textbereichen ist Arial und Namen in Formelbereichen sind kursiv. Außerdem ist dort das Minus länger und hat etwas mehr Platz. Im folgenden Beispiel fügen wir erst einen Textbereich ein und dann eine Formel:

"Strg-b ↵ Strg-b

|                 |               |
|-----------------|---------------|
| Strg-b          | Textbereich   |
| <i>strg - b</i> | Formelbereich |

Bereiche kann man mit □ (Werkzeuggestreife) einrahmen.

F = 4,47 N

Die Formatierungsmöglichkeiten finden sich in der Werkzeuggestreife am oberen Rand des Rechenblatts:



- Schriftart (nicht in Formelbereichen wirksam) und Schriftgröße
- Fettschrift (*bold*), Kursiv (*italic*) und Unterstrichen (*underline*)
- Schriftfarbe A und Hintergrundfarbe 🎨
- Einrahmung

Diese Optionen sollten vorsichtig benutzt werden, denn das Rechenblatt ist schnell durch wahllose Formatierung verhunzt.

Man kann auch Formeln in Textbereiche einfügen: **Einfügen** > **Formelbereich**. Die eingebetteten Formeln werden nach der Reihenfolge im Zeilenfluss ausgeführt.

Abschätzung einer Wurzel mittels linearer Näherung (Taylorreihe erster Ordnung):

$$\sqrt{19} = \sqrt{4^2 + 3} \approx 4 + \frac{1}{2} \cdot \frac{3}{4} = 4,375$$

Zum Vergleich:  $\sqrt{19} = 4,3589$

Abschätzung einer Wurzel mittels linearer Näherung (Taylorreihe erster Ordnung):

$$\sqrt{19} = \sqrt{4^2 + 3} \approx 4 + \frac{1}{2} \cdot \frac{3}{4} = 4,375$$

Zum Vergleich:  $\sqrt{19} = 4,3589$

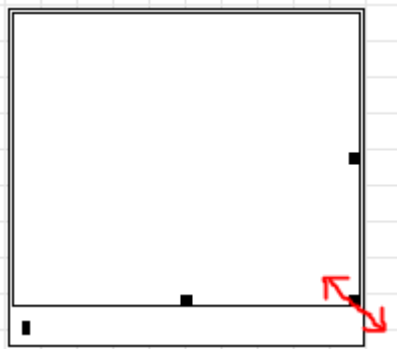
### 3.7. Bilder einfügen<sup>1</sup>

Häufig ist es sinnvoll, Bilder zur Erläuterung der Rechnung einzufügen. Bilder kann man aus der Zwischenablage einfügen oder aus SMath heraus erzeugen. Letzteres geht so:

<sup>1</sup>Erfordert das Plug-in *Image Region*

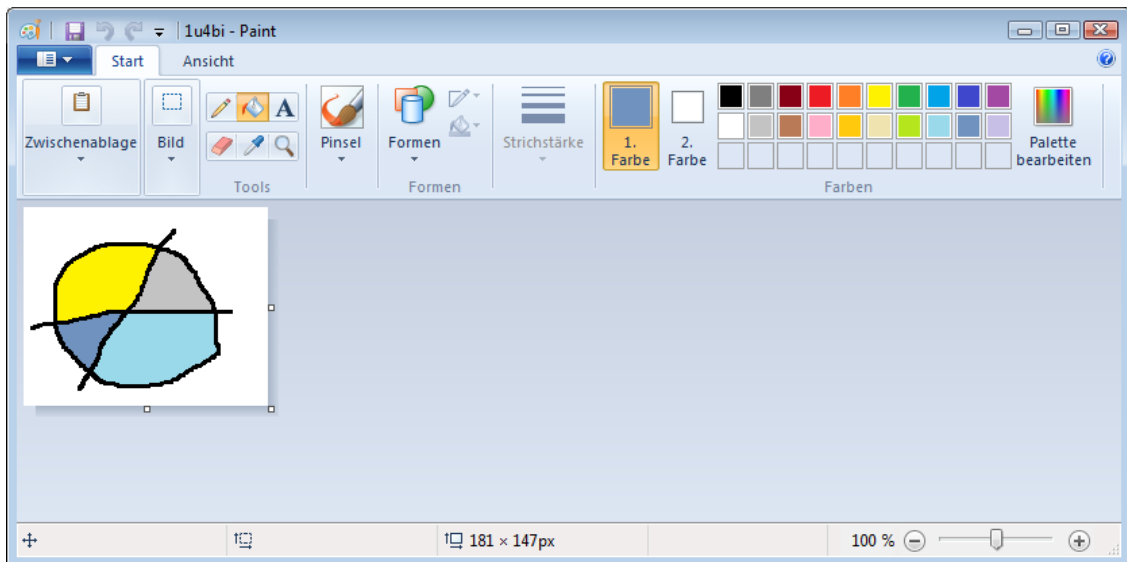


1. Sie fügen zunächst einen Image-Bereich ein: **Einfügen** > **Image** und ziehen ihn auf die ungefähre gewünschte Größe:



2. Doppelklick auf den Bereich:

Es öffnet sich das Standardprogramm für die PNG-Bearbeitung (hier ist das Paint) mit einem temporären zufälligen Dateinamen. Zeichnen Sie das Bild. Sie können dabei auch die Größe des Bildes anpassen.



Schließen Sie das Programm und bestätigen Sie die Speicherung. Damit Sie das Bild in SMath sehen, müssen Sie eventuell kurz neben den Bildbereich und dann wieder darauf klicken.

3. Sie können das Bild an den Randmarken vergrößern oder verkleinern, auch wenn es schon Inhalt hat.

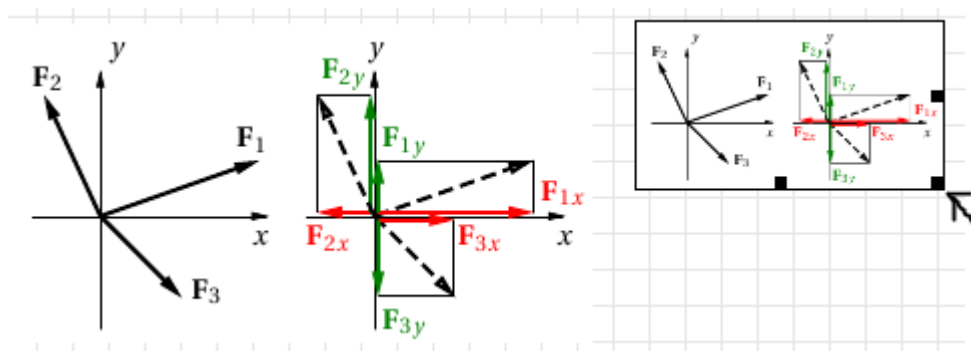
Im Kontextmenü können Sie unter anderem

- das Bild auf die tatsächliche Größe zurücksetzen („Reset to original size“),
- den Rand ausschalten („Switch border on/off“)
- den Platzhalter verbergen („Eingabedaten anzeigen“)



So erzeugte Bilder können durch Doppelklick erneut editiert werden.

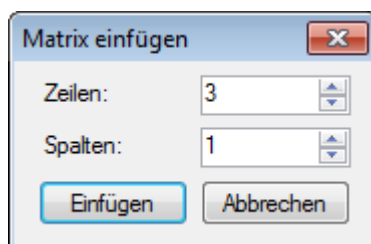
Man kann auch Bilder direkt per  $\boxed{\text{Strg}}-\boxed{\text{V}}$  aus der Zwischenablage einfügen (dabei entsteht ein sogenannter Bildbereich (*picture region*)). Solche Bilder sind für die spätere Bearbeitung innerhalb von SMath Studio nicht geeignet. Man kann aber auf diese Weise sehr einfach externe Bilder oder Screenshots einfügen.



An den Randmarken ■ kann man mit der Maus ziehen, um die Größe zu ändern.

## 3.8. Vektoren und Matrizen

SMath kennt einige Vektor- und Matrixoperationen. Der Befehl  $\boxed{\text{M}}$  aus der Matrix-Palette der Seitenleiste öffnet den Dialog

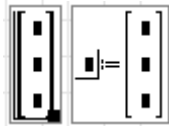


Er ist hier schon auf 3 Zeilen und 1 Spalte eingestellt. Alternativ ruft man den Dialog mit  $\boxed{\text{Strg}}-\boxed{\text{M}}$  auf.  $\boxed{\text{Einfügen}}$  liefert einen passenden Platzhalter



Stellt man die Einfügemarke vor die linke Klammer, kann man vor der Matrix etwas einfügen, z.B. den Zuweisungsoperator (Doppelpunkt)

:



Diese Platzhalter werden nun ausgefüllt. Dabei kann man mit den Pfeiltasten oder mit der Maus zwischen den Platzhaltern navigieren.

F. A

1

1

1

$$F_A := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Der mit dem Punkt . erzeugte Index ist ein Textindex.

Rechte Klammer anklicken und noch eine Einheit hinzufügen:

\*N

$$F_A := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \text{ N}$$

Bei der Anzeige kann eine Maßeinheit für die ganze Matrix gewählt werden:

$$F_A = \begin{bmatrix} 1 & \frac{\text{kg m}}{\text{s}^2} \\ 1 & \frac{\text{kg m}}{\text{s}^2} \\ 1 & \frac{\text{kg m}}{\text{s}^2} \end{bmatrix}$$

Dafür wählt man den Platzhalter an und tippt

N

$$F_A = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \text{ N}$$

Man kann Vektoren und Matrizen auch erzeugen, indem man einzelnen Elementen Werte zuweist. Elemente werden mit Elementindizes angesprochen. Elementindizes werden durch [ oder mit dem Symbol  $\blacksquare$  in der „Matrizen“-Palette erzeugt:

F. B[2]  : 3kN

$$F_B := 3 \text{ kN}$$

Elementindizes (mit `[]` erzeugt) unterscheiden sich in der Formatierung von Textindizes (mit `.` erzeugt) durch einen größeren Abstand vom Formelzeichen. Zudem ist die Schrift in Textindizes kleiner.

Verwendet man nur *einen* Elementindex, wird ein Spaltenvektor erzeugt. Die erzeugten Matrizen oder Vektoren sind gerade so groß, wie die höchsten vorkommenden Indizes.

F.B.=

$$F_B = \begin{bmatrix} 0 \\ 3000 \frac{\text{kg m}}{\text{s}^2} \end{bmatrix}$$

Die Addition von  $F_A$  und  $F_B$  scheitert, weil die Matrizen unterschiedlich groß sind:

F.A+F.B=

$$F_A + F_B = \blacksquare \blacksquare$$

Die Anzahl der Zeilen oder Spalten der ersten Matrix entspricht nicht der Anzahl der Zeilen oder Spalten der zweiten Matrix.

Definiert man noch das dritte Vektorelement, dann geht es:

F.B[3]  : 0

$$F_B := 0$$

$$F_A := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \text{ N} \quad F_B := 3 \text{ kN} \quad F_B := 0$$

$$F_B = \begin{bmatrix} 0 \\ 3000 \frac{\text{kg m}}{\text{s}^2} \\ 0 \end{bmatrix} \quad F_A + F_B = \begin{bmatrix} 1 \\ 3000 \\ 1 \end{bmatrix} \text{ N}$$

Dabei haben wir uns das Ergebnis in Newton (N) anzeigen lassen.

Das Skalarprodukt zweier Vektoren schreibt man als gewöhnliches Produkt:

F.A\*F.B=

$$F_A \cdot F_B = 3000 \frac{\text{kg W}}{\text{s}}$$

Der Betrag eines Vektors ist die Wurzel aus dem Skalarprodukt mit sich selbst:

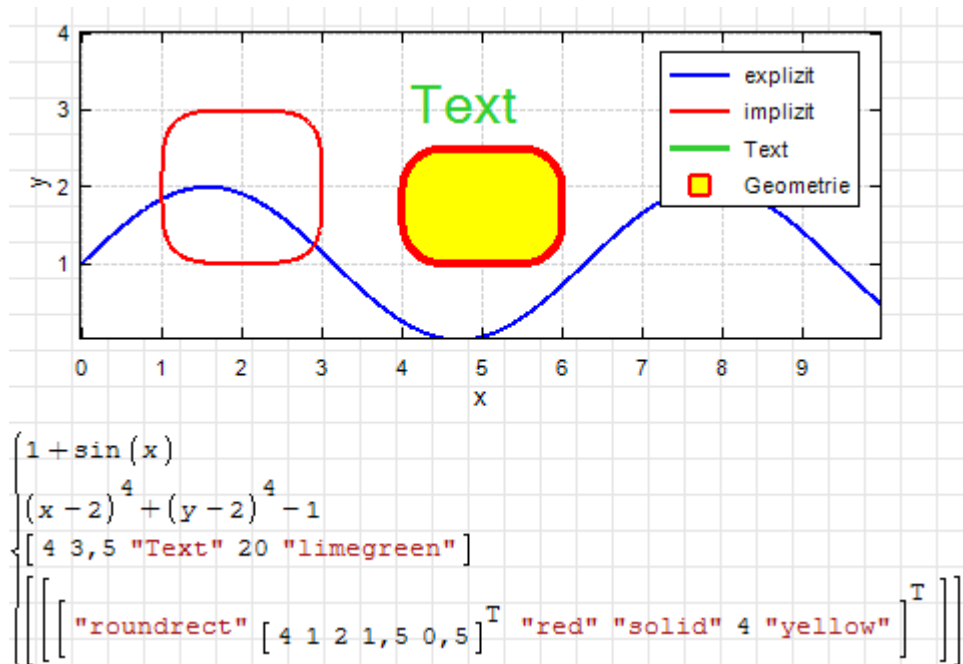
\F.A^2=

$$\sqrt{F_A^2} = 1,73 \text{ N}$$

## 3.9. Diagramme

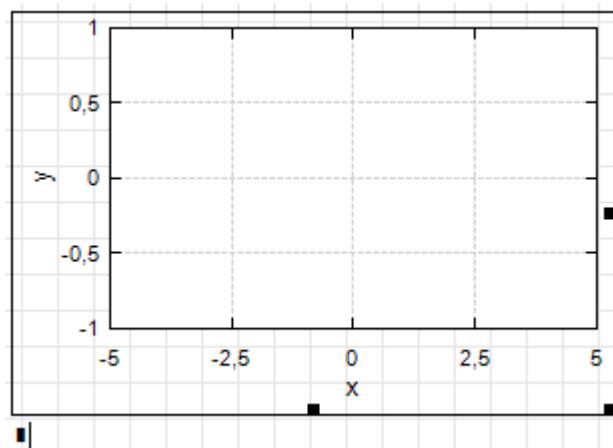
Für Diagramme empfehlen wir den XY-Plotbereich aus dem Plug-in *X-Y Plot Region*. Er kann explizite Funktionen  $f(x)$  und implizite Funktionen  $f(x; y) = 0$  sowie Textmarken

und geometrische Grundformen darstellen. Sein Erscheinungsbild kann in einem Einstellungsmenü angepasst werden.



Daneben gibt es noch den eingebauten 2D-Diagrammbereich, der etwas kompakter ist, aber dessen Erscheinungsbild nur wenig angepasst werden kann. Erweiterte Optionen bieten die Grafik-Funktionen der Maxima-Schnittstelle (Abschnitt 8.1).

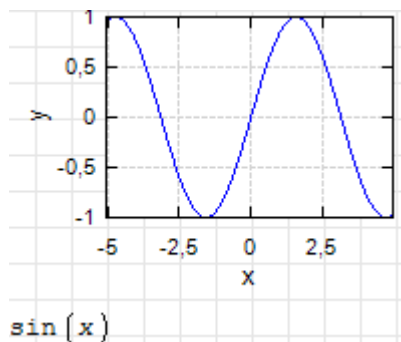
Ein neuer XY-Plotbereich wird erzeugt mit **Einfügen** > **Diagramm** > **X-Y Plot**:



In den Platzhalter unten links schreibt man den zu plottenden Ausdruck. An den Randmarken kann man mit der Maus ziehen, um die Diagrammgröße zu ändern. Der Rand verschwindet, wenn man den Bereich verlässt.

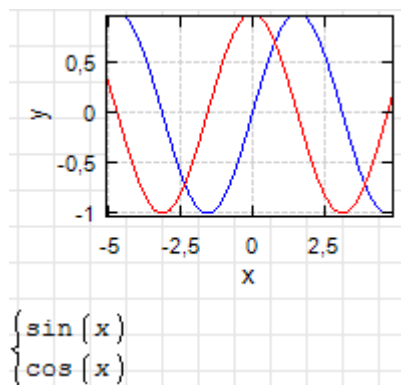
Bei Funktionsplots kann als unabhängige Variable irgendein undefinierter Name (der noch ohne Wert ist) benutzt werden. Im folgenden Bild wurde der Bereich auf minimale Größe gebracht.

`sin(x)`



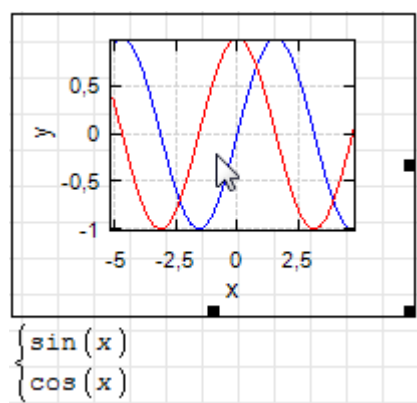
**Mehrere Funktionen** werden in einer Liste („System“) gruppiert. Das ist etwas ähnliches wie ein Vektor und hat nur links eine geschweifte Klammer. Man bekommt sie durch Eintippen von `sys`  $\left\{ \right\}$  oder mit dem Symbol  $\left\{ \right\}$  in der Funktionen-Palette der Seitenleiste. Hier ein Beispiel:

`sys`  $\left\{ \right\}$  `sin(x)`  $\left\{ \right\}$  `cos(x)`  $\left\{ \right\}$



### Bereichseinstellung in Diagrammen:

Das Diagramm muss zunächst angeklickt werden, so dass die Randmarken zu sehen sind.




- Ausschnitt verschieben: Klicken und Ziehen der Diagrammfläche
- Beide Achsen zoomen: Mausrad
- nur  $x$ -Achse zoomen:  $\left\{ \right\}$ -Mausrad
- nur  $y$ -Achse zoomen: `Strg`-Mausrad

Bei Doppelklick auf das Diagramm öffnet sich ein Fenster, in dem man Merkmale wie Achsenteilung, Wertebereiche, Beschriftungen, Farben, Linienarten und Legenden angeben kann. All das wird im Abschnitt 8.3 behandelt.



## 3.10. Etwas Programmierung

Für die Datenverarbeitung und deren grafische Aufbereitung bieten sich häufig Schleifen an. Man erzeugt sie mit „for“ aus der Palette „Programmierung“ der Seitenleiste oder mit der Tastatur:


for  i

```
for i ∈ [ ]
  |
```

Es entstehen drei Platzhalter, den ersten haben wir bereits mit dem Schleifenzähler  $i$  ausgefüllt. Der Zähler soll nacheinander Werte von 1 bis 5 annehmen. Dafür verwenden wir die Funktion `range()` auf dem zweiten Platzhalter

 rang  1



```
for i ∈ [1.. ]
  |
```

 5

```
for i ∈ [1..5]
  |
```


Die Funktion `range()` ist auch über **Seitenleiste** > **Palette „Matrizen“** erreichbar (Symbol ).

Der dritte Platzhalter ist der Schleifenkörper.

```
for i ∈ [1..5]
  |
```

Dort wollen wir nun einen Vektor erzeugen, dessen  $i$ -tes Element die Zahl  $i^2$  enthält. Der Elementindex  $i$  bei  $v$  wird mit der eckigen Klammer auf `[` erzeugt und unterscheidet sich von Textindizes durch den zusätzlichen Zwischenraum:

v[i  :i<sup>2</sup>

```
for i ∈ [1..5]
  v_i := i2
```


Das Ergebnis entspricht den Erwartungen.

```
for i ∈ [1..5]
  v_i := i^2
v = [ 1
      4
      9
      16
      25 ]
```

Derartige Schleifen kann man auch implizit hinschreiben. Man gibt den Wertebereich der Indexvariable an und schreibt dann die für jeden Indexwert auszuführenden Gleichungen hin.

```
j := [1..5]
x_j := j
v_j := x_j^2
j:rang 1 5
x[j]
v[j] : x[j]^2
```

```
x = [ 1
      2
      3
      4
      5 ]
v = [ 1
      4
      9
      16
      25 ]
```

Alternativ kann man auch die Vektorisierungsfunktion benutzen (`vectorize()`, Taste  in der Palette „Matrizen“):


```
y := x^2 = 1 m
y:vec x^3 =
```

Nach Durcharbeiten der „Ersten Schritte“ sollten Sie in der Lage sein, das Rechenblatt zur Interpolation in Abschnitt 5.17 nachzuvollziehen.

### 3.11. Tips und Tricks für unfallfreies Arbeiten

- In Namen wird Groß- und Kleinschreibung unterschieden!
- Benutzen Sie `i` nur dann als Schleifenzähler (auch in Summen- oder Produktoperatoren), wenn Sie nicht mit komplexen Zahlen rechnen wollen. Dadurch ist `i` nicht mehr als imaginäre Einheit verwendbar.
- Schreiben Sie logische Operationen (`=`, `≠`, `∧`, `∨`, `⊕`) mit ihren Operanden immer in Klammern, also  $(a \wedge b)$ . Diese Operatoren haben unerwartet niedrige Priorität, insbesondere niedriger als der Zuweisungsoperator `:=`.
- Beachten Sie den Unterschied zwischen beschreibenden Textindizes, z.B. `F.x` und Elementindizes für Matrizen und Vektoren, z.B. `v[i]`.
- Definieren Sie stückweise stetige Funktionen durch Multiplikation mit logischen Ausdrücken oder mit der `cases()` Funktion, nicht jedoch mit der `if()`-Anweisung.
- **Maßeinheiten** mit der Tastatur:
  - Eingabe eines Hochkommata und des Namens



- Eingabe des Namens und Bestätigung mit  $\boxed{\leftrightarrow}$  in der Vorschlagsliste
- Unnötige Klammern entfernen:
  1. Inneres markieren und kopieren oder ausschneiden
  2. Klammer mit Inhalt markieren, einfügen (durch Zwischenablage ersetzen)
- Interaktives Handbuch nutzen
  - Aufruf über **Werkzeugleiste** 
  - [Verweisen](#) mit  $\boxed{\text{Strg}}$ -Klick folgen
- Wichtige Operatoren und ihre Tastatureingabe (eine vollständige Liste finden Sie im Anhang B):

| Operator   | Bedeutung                      | Eingabe                |
|--|--------------------------------|------------------------|
| $\mathbf{a} \mid \mathbf{b}$   | Substitution                   | <code>at(;</code>      |
| $\left\{ \begin{array}{l} \mathbf{a} \text{ if } \mathbf{b} \\ \mathbf{c} \text{ otherwise} \end{array} \right.$ | Fallunterscheidung             | <code>cases(;;</code>  |
| $\mathbf{a} \mathbf{b}$  | Elementadressierung in Feldern | <code>[</code>         |
| $\mathbf{a} \mathbf{b} \mathbf{c}$   | mit zwei Indizes               | <code>[;</code>        |
| $\frac{d}{d} \mathbf{a}$   | Differentiation                | <code>diff(;</code>    |
| $\frac{d}{d} \mathbf{a} \mathbf{b}$  | Differentiation                | <code>diff(;;</code>   |
| $\int \mathbf{a} \mathbf{b}$   | Integration                    | <code>int(</code>      |
| $\sum \mathbf{a}$  | Summe                          | <code>sum(</code>      |
| $\sum_{\mathbf{a}=\mathbf{b}} \mathbf{c}$  |                                | <code>sum(;;;</code>   |
| $[\mathbf{a} \dots \mathbf{b}]$  | Wertebereich (Vektor)          | <code>range(;</code>   |
| $[\mathbf{a}; \mathbf{a} \dots \mathbf{b}]$  |                                | <code>range(;;;</code> |

# 4. Programmbedienung

Hier werden die Grundfunktionen von SMath Studio behandelt, die für den Umgang mit Rechenblättern notwendig und hilfreich sind. Das betrifft das Hilfesystem mit Dynamischer Hilfe und dem Interaktiven Handbuch, Globale Einstellungen und Optionen. Eingegangen wird auch auf wichtige Verzeichnisse, die Zwischenablage und das Drucken von Dokumenten.

## 4.1. Benutzeroberfläche

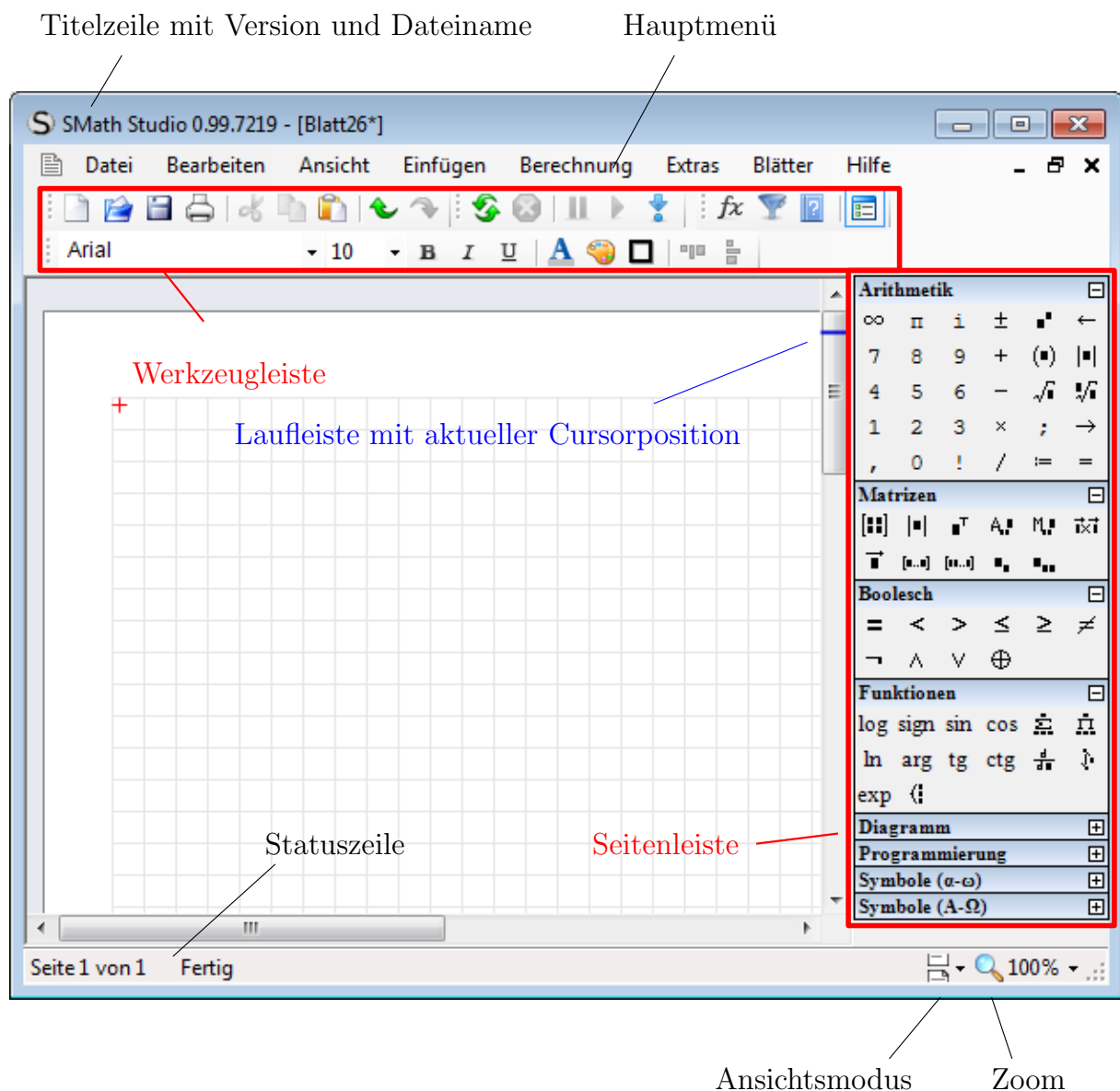





Abbildung 4.1.: Elemente der Benutzeroberfläche

Die Benutzeroberfläche besteht aus den Elementen

- **Titelzeile** mit Namen des aktuellen Rechenblatts. Bei einer portablen Installation wird auch die Programmversion angezeigt
- **Hauptmenü** (nicht konfigurierbar),
- **Werkzeugleiste** (Anordnung der Teilleisten kann durch Ziehen an den Marken  verändert werden),
- **Seitenleiste** mit Symbol- und Funktionspaletten. Diese kann über den Knopf  ein- und ausgeschaltet werden. Die einzelnen Paletten können mit dem Knopf  eingeklappt werden.
- **Statuszeile** mit Zoomsteuerung und Auswahl des Ansichtsmodus (Einzelseiten, Druckbereich, unbegrenztes Blatt),
- **Rechenblatt**, eventuell mehrere davon, je nach Fenstereinstellung. Rechenblätter werden im Abschnitt 5 detailliert behandelt.
- Senkrechte und waagerechte **Laufleiste**.

### 4.1.1. Hauptmenü

Das Hauptmenü befindet sich unterhalb der Titelzeile und hat die folgenden Einträge:



**Datei** Funktionen, die man bei einem Programm zur Erstellung von Dokumenten erwartet (Speichern einschließlich Export in andere Formate, Seiteneinstellungen, Druck). Im Unterpunkt „Eigenschaften“ kann man Metadaten festlegen (Autor, Inhaltsangabe und ähnliches).

**Bearbeiten** Standard-Bearbeitungsfunktionen (Rückgängig, Wiederherstellen, Ausschneiden, Kopieren, Einfügen, Löschen, Alles markieren, Suchen, Aktualisieren). Diese Funktionen wird man wohl eher mit ihren Tastaturkürzeln aufrufen.

**Ansicht** Ein- und Ausschalten von Hilfsfenstern (Debugger, Ausgabefenster) und Eingabehilfen (dynamische Hilfe, Gitterraster, Mehrseitenansicht).

**Einfügen** Zugriff auf die Grundbausteine (Bereiche) von SMath-Dokumenten. Das Angebot hängt von den installierten Anwendungserweiterungen (Plug-ins) ab.

**Berechnen** Automatischer oder manueller Start von Berechnungen, Verhalten bei Fehlern.

**Extras** Zugriff auf die Erweiterungsverwaltung und Programmeinstellungen.

**Blätter** Zugriff auf die offenen Rechenblätter.

**Hilfe** Beispiele, Handbuch, Spendenkonto, Versionsinfo, Online-Version

Die Untereinträge des Hauptmenüs werden im Anhang E beschrieben.

### 4.1.2. Werkzeugleiste

Die einzelnen Gruppen der Werkzeugleiste können im Bereich unter dem Hauptmenü frei angeordnet werden.



Neues Blatt, Öffnen, Speichern, Drucken



Ausschneiden, Kopieren, Einfügen



Rückgängig, Wiederherstellen



Schriftart, Schriftgröße, Fett, Kursiv, Unterstreichung, Schriftfarbe, Hintergrundfarbe,



Vertikal ausrichten, horizontal ausrichten



Funktion einfügen, Einheit einfügen, Interaktives Handbuch



Neu berechnen, Berechnung stoppen




Seitenleiste aus/an

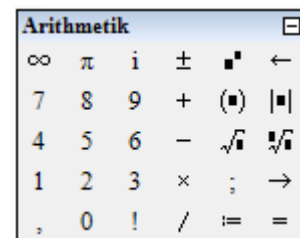


Pause, Fortsetzen, Einzelschritt

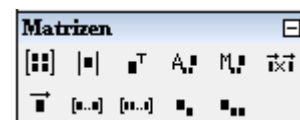
### 4.1.3. Seitenleiste

Die Seitenleiste bietet den Schnellzugriff auf Symbole, Operatoren und Funktionen, ohne dass man deren Tastaturkürzel kennen muss. Die Seitenleiste kann über die Taste  der Werkzeugleiste an- und ausgeschaltet werden.

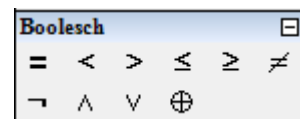
**Arithmetik:** Symbole und Funktionen für die Grundrechenoperationen und einige weitere arithmetische Operationen.



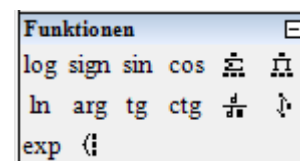
**Matrizen:** Funktionen für das Rechnen mit Matrizen und Vektoren.



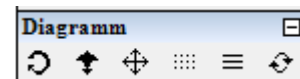
**Boolesch:** logische (boolesche) Operationen.



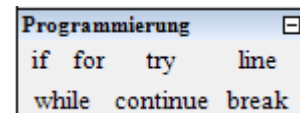
**Funktionen:** diverse Funktionen.



**Diagramm:** Funktionen zur Steuerung der 2D- und 3D-Grafikanzeige.



**Programmierung:** Funktionen für die Ablaufsteuerung in Funktionsdefinitionen (Programmen)



**Symbole ( $\alpha - \omega$ ):** griechische Kleinbuchstaben.

| Symbole ( $\alpha - \omega$ ) |          |            |            |            |         |
|-------------------------------|----------|------------|------------|------------|---------|
| $\alpha$                      | $\beta$  | $\gamma$   | $\delta$   | $\epsilon$ | $\zeta$ |
| $\eta$                        | $\theta$ | $\iota$    | $\kappa$   | $\lambda$  |         |
| $\mu$                         | $\nu$    | $\xi$      | $\omicron$ | $\pi$      | $\rho$  |
| $\sigma$                      | $\tau$   | $\upsilon$ | $\phi$     | $\chi$     |         |
| $\psi$                        | $\omega$ |            |            |            |         |

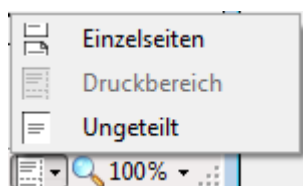
**Symbole ( $A - \Omega$ ):** griechische Großbuchstaben. Verwenden Sie nur solche, die sich eindeutig von lateinischen Großbuchstaben abheben.

| Symbole ( $A - \Omega$ ) |          |          |          |           |          |
|--------------------------|----------|----------|----------|-----------|----------|
| $A$                      | $B$      | $\Gamma$ | $\Delta$ | $E$       | $Z$      |
| $H$                      | $\Theta$ | $I$      | $K$      | $\Lambda$ | $M$      |
| $N$                      | $\Xi$    | $O$      | $\Pi$    | $P$       | $\Sigma$ |
| $T$                      | $Y$      | $\Phi$   | $X$      | $\Psi$    | $\Omega$ |

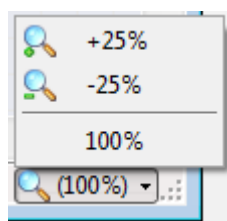
#### 4.1.4. Statuszeile

Die Statuszeile enthält folgende Elemente:

- Anzeige der aktuellen Seite und der Seitenanzahl des aktuellen Rechenblatts
- aktuelle Rechenzeit/Berechnungsstatus
- Anzeigemodus (mit Seitenrändern, Druckbereich mit Seitenumbruch, unbegrenztes Blatt).



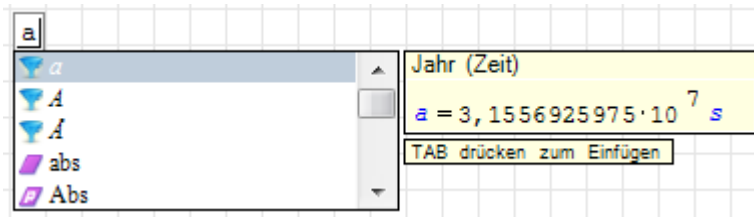
- Zoom-Einstellung des Rechenblatts.










## 4.2. Dynamische Hilfe

Eine Besonderheit von SMath Studio ist die Dynamische Hilfe. Wann immer Sie im Rechenblatt eine Formel eintippen, klappt ein dynamisches Fenster auf, welches Ihnen mögliche Ergänzungen und deren Beschreibung oder Definition anzeigt. Sie können in der Liste mit den Pfeiltasten auswählen und die Auswahl mit  $\left[ \leftrightarrow \right]$  bestätigen.

Die dynamische Hilfe kennt nicht nur die eingebauten Funktionen, Maßeinheiten und Konstanten, sondern auch die von Ihnen selbst definierten. Darüber hinaus kennt sie alle Funktionen und Konstanten aus eventuell installierten Anwendungserweiterungen (Plugins).

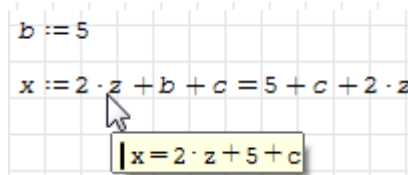


Die Symbole in der Liste haben folgende Bedeutung:

- Eingebaute Konstante oder Anweisung (Eulersche Zahl  $e$ , imaginäre Einheit  $i$ , aber auch Programmanweisungen wie `break` und `continue`)
-  selbiges, aber vom Benutzer überschrieben (tun Sie das nur, wenn Sie wissen, was Sie tun)
-  Eingebaute Funktion
-  Variable oder Funktion aus einem Plug-in (extern oder intern)
-  Benutzerdefinierte Variable oder Funktion
-  Eingebaute Maßeinheit oder physikalische Konstante
-  Benutzerdefinierte Maßeinheit oder physikalische Konstante
-  Code-Baustein

Bei selbstdefinierten Bezeichnern können Sie eine Beschreibung angeben, die in der dynamischen Hilfe erscheint, siehe Abschnitt 5.13.

Die dynamische Hilfe erscheint nicht nur, wenn man einen Bezeichner verwendet, sondern auch wenn er definiert wird. Stellt man die Maus auf einen Formelbereich mit einer Definition, dann wird in einem Hilfefenster der gespeicherte Ausdruck angezeigt.



## 4.3. Interaktives Handbuch

SMath Studio bietet die Möglichkeit, einen Satz miteinander verlinkter Rechenblätter als sogenanntes „Referenzmaterial“ zu definieren. Dieses kann man aufrufen über

- **Werkzeugleiste** > 
- **Hauptmenü** > **Hilfe** > **Referenzmaterial**

Standardmäßig erscheint dort ein kleines mathematisches Nachschlagewerk.

In der THB-Distribution ist ein SMath-Handbuch installiert, welches aus etwa 500 Einzeldokumenten besteht. Das Handbuch ist in Kapitel gegliedert. Die blauen unterstrichenen Texte sind [Hyperlinks auf SMath-Dokumente](#). Mittels Strg-Mausklick werden sie aktiviert. Das aktuell geladene Dokument wird durch das Zieldokument ersetzt, so wie man dies auch von Webbrowsern kennt.

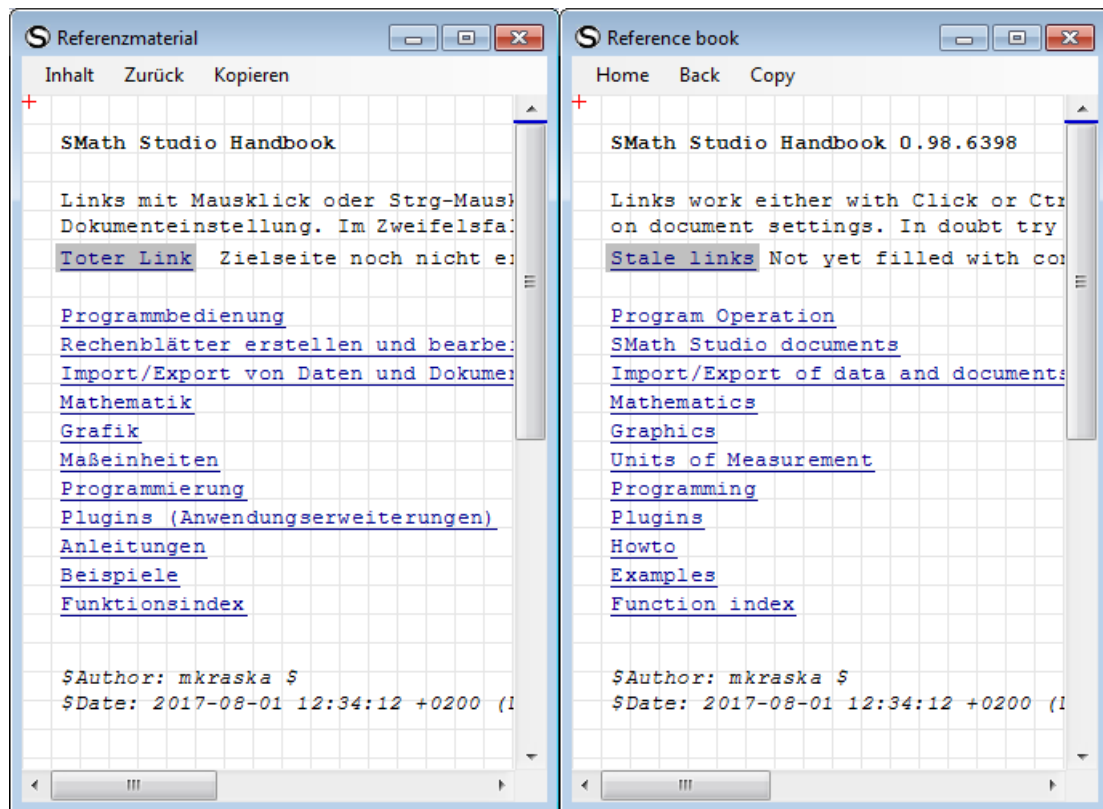


Abbildung 4.2.: Startseite des interaktiven Handbuchs in deutscher und englischer Version.

Zu manchen Links (insbesondere im Funktionsindex) existieren noch keine Dokumente. Solche toten Links sind grau hinterlegt.

Leider sind die Schaltflächen Inhalt, Zurück und Kopieren funktionslos. Daher wurden entsprechende Navigationslinks in den Unterdokumenten selbst angelegt.

|   |  |
|---|--|
| $\text{Rkadapt}(y_0; x_a; x_e; m; D(x; y))$   |  |
| Bestandteil des Plugins   | <a href="#">ODE Solver</a>               |
| Mathcad-kompatibler Löser für gewöhnliche Differenzialgleichungssysteme nach dem Runge-Kutta-Fehlberg Verfahren mit adaptiver Schrittweite. |  |
| $y_0$   | Startwerte für die abhängigen Variablen  |
| $x_a$   | Anfangswert für die unabhängige Variable |
| $x_e$   | Endwert für die unabhängige Variable     |
| $m$   | Zahl der Ausgabeschritte                 |
| $D(x; y)$   | Ableitungsvektor                         |
| Die Funktion wird wie in der Einführung beschrieben verwendet.  |  |
| <a href="#">Einführung</a>  |  |
| <a href="#">Gewöhnliche Differenzialgleichungen</a>   |  |
| <a href="#">Plugin ODE Solver</a>   |  |
| <a href="#">Funktionsindex</a>  |  |
| <a href="#">Inhalt</a>  |  |

Abbildung 4.3.: Beispiel einer Handbuchseite mit Navigations-Hyperlinks

Die Handbuchseiten sind schreibgeschützt, die enthaltenen Berechnungen werden aber tatsächlich ausgeführt und können auch in eigene Rechenblätter kopiert werden.

Die Dateien des interaktiven Handbuchs findet man im Installationsverzeichnis unter *extensions\books*. Man kann sie auch direkt in SMATH Studio öffnen, die Navigation mit Hilfe der Querverweise funktioniert dort genauso.

## 4.4. Die Zwischenablage

### 4.4.1. Aus SMATH in die Zwischenablage kopieren

Ausdrücke in Formelbereichen und Text in Textbereichen können (dunkelblau) markiert und mit  $\text{Strg} + \text{C}$  in die Zwischenablage übernommen werden. Der Inhalt steht auch Anwendungen außerhalb SMATH als einfacher Text im internen Format (siehe Abschnitt 6.1.4).

Beliebige Bereiche kann man als Ganzes hellblau markieren (Ziehen mit der Maus) und als Rastergrafik in die Zwischenablage kopieren. Sie können dann in beliebige Anwendungen als Bild eingefügt werden.

**Tabellenbereiche** können über **Kontextmenü > Export > Copy to clipboard** in die Zwischenablage übernommen werden und stehen dort als HTML-Tabelle, einfacher Text oder Bild zur Verfügung (z.B. für das Einfügen in Office-Dateien).

In **erweiterten Textbereichen** kann wie üblich kopiert werden, der Inhalt der Zwischenablage steht extern als einfacher Text oder als HTML zur Verfügung.



**Snapshot-Bereiche** (Abschnitt 4.6.2) können ihren Inhalt ebenfalls in die Zwischenablage kopieren (Kontextmenü).

#### 4.4.2. Einfügen aus der Zwischenablage

Aus der Zwischenablage können folgende Arten von Daten eingefügt werden:

- **Bilder:** Rastergrafiken werden als Bildbereich eingefügt, der anschließend in der Größe geändert und verschoben werden kann.
- **Text:** Gültige SMath-Ausdrücke im internen SMath-Format können eingefügt werden:
  - in bestehende Formelbereiche (wenn die Einfügemarke darin steht) oder
  - als neue Formelbereiche (wenn die Einfügemarke auf dem Blatt steht).
- Enthält die Zwischenablage Leerzeichen, wird der Inhalt als Textbereich eingefügt. Befindet man sich in einem Formelbereich, wird der Text nur in bestehende Zeichenketten (diese können auch leer sein) eingefügt.

### 4.5. Verzeichnisse

In SMath Studio spielen drei Verzeichnisse eine Rolle, die jeweils mit einer Funktion angezeigt werden können:

```

Aktuelles Verzeichnis
CurrentDirectory("") = "C:\FHB\Software\SMath\SMath 99.7219\"

Dokumentverzeichnis
DocumentDirectory("") = "C:\FHB\Software\SMath\SMath Skript\Bilder\"

Einstellungsverzeichnis
SettingsDirectory("") = "C:\FHB\Software\SMath\SMath 99.7219\"

```

Außerdem ist noch das Installationsverzeichnis wichtig. Dafür gibt es keine Funktion, aber wenn man SMath Studio ohne Datei startet, dann wird das aktuelle Verzeichnis auf das Installationsverzeichnis gesetzt, so dass man sich dieses mit `CurrentDirectory("")` anzeigen lassen kann.

Arbeitet man mit externen Dateien, dann ist es in der Regel sinnvoll, das aktuelle Verzeichnis auf das Dokumentverzeichnis zu setzen. So beziehen sich relative Pfadnamen auf das Verzeichnis, wo das aktuell offene Rechenblatt liegt:

```

dir := CurrentDirectory(DocumentDirectory(""))
dir = "C:\FHB\Software\SMath\SMath Skript\Bilder\"

```

## Installationsverzeichnis

Das Installationsverzeichnis ist das Verzeichnis, wo SMath installiert wurde, z.B. `D:\Programme\SMath\`. Dort liegt die ausführbare Datei `SmathStudio_Desktop.exe`. Wird SMath Studio über das Startmenü gestartet, dann ist dieses Verzeichnis auch das aktuelle Verzeichnis (*current directory*).

Im Installationsverzeichnis befinden sich weitere Unterverzeichnisse:

**book** Hier liegt das im Standardumfang mitgelieferte Referenzmaterial (Mathematik-Formelsammlung).

**entries** Enthält momentan nur die Maßeinheiten- und Konstantendefinition.

**EULA** Text der Lizenzvereinbarung (*end user license agreement*).

**examples** Mitgelieferte Beispieldateien.

**extensions** In der portablen Version legt die Erweiterungsverwaltung (siehe Kapitel 12) hier die aus der Online-Galerie heruntergeladenen Erweiterungen (Plug-ins, Code-Bausteine, Beispiele) ab. Hier befinden sich auch die vorinstallierten Plug-ins und das interaktive Handbuch.

In der Standardversion (per Installer installiert) liegen diese Erweiterungen im Einstellungsverzeichnis.

**lang** Texte für die verschiedenen Sprachversionen der Benutzeroberfläche.

**plugins** Hier liegen die standardmäßig installierten Plug-ins (Anwendungserweiterungen).

**representation** Hier liegt nur die XML-Datei `settings.prop`, welche die Schriftarten in der Formelanzeige definiert.

**snippets** Hier liegen die standardmäßig mitgelieferten Bausteine (*snippets*), siehe Kapitel 10.5.

## Einstellungsverzeichnis

Dies ist das Verzeichnis, in welchem SMath benutzerspezifische Daten ablegt, darunter auch die Datei `settings.inf` mit den Programmeinstellungen. Der Erweiterungsmanager legt hier die Verzeichnisstruktur *Extensions* an, in der die heruntergeladenen Erweiterungen abgelegt werden.

Dieses Verzeichnis wird auch von den Funktionen `wfile()`, `rfile()` oder `dfile()` zum Datenaustausch benutzt (siehe Abschnitt 6.6.7).

Man kann das Verzeichnis mit der Funktion `SettingsDirectory()` aus dem Plugin *Custom Functions* ermitteln:

```
SettingsDirectory(■) = "C:\Users\kraska\AppData\Roaming\SMath\"
SettingsDirectory(0) = "C:\Users\kraska\AppData\Roaming\SMath\"
```

Bei der Installer-Version liegt dieses Verzeichnis in der Windows-Anwenderdatenstruktur, bei der portablen Version ist das Einstellungsverzeichnis mit dem Installationsverzeichnis identisch.

## Aktuelles Verzeichnis

Einige Dateifunktionen wie `ImportData()` oder `exportData.CSV()` benötigen Dateinamen oder Pfade, die entweder relativ oder absolut sein können.

Relative Namen beziehen sich auf das aktuelle Verzeichnis, absolute Namen enthalten den vollständigen Pfad.

Das aktuelle Verzeichnis ist das Verzeichnis, wo SMath gestartet wurde. Wird das Programm ohne zu öffnende Datei gestartet, dann ist das im Installationsverzeichnis.

Wurde eine .sm-Datei als registrierter Dateityp im Dateimanager gestartet, dann wird das Verzeichnis, wo die .sm-Datei liegt, das aktuelle Verzeichnis.

Das aktuelle Verzeichnis ändert sich nicht, wenn eine Datei an anderer Stelle gespeichert wird.

Mit der Funktion `CurrentDirectory()` aus dem Plug-in *Mathcad File Access Functions* kann das aktuelle Verzeichnis ermittelt und gesetzt werden:

```
CWD := CurrentDirectory("")  
  
CurrentDirectory("") = "C:\FHB\Software\SMath\SMath Skript\Bilder\  
  
CWD = "C:\FHB\Software\SMath\SMath Skript\Bilder\  
  
CurrentDirectory("d:\tmp") = "C:\FHB\Software\SMath\SMath Skript\Bilder\  
  
CurrentDirectory(CWD) = "C:\FHB\Software\SMath\SMath Skript\Bilder\  

```

## Dokumentverzeichnis

Das Dokumentverzeichnis ist das Verzeichnis, in welchem das aktuell offene Dokument liegt. Man kann es mit der Funktion `DocumentDirectory()` anzeigen. Mit der Kombination `CurrentDirectory(DocumentDirectory(""))` kann man sicherstellen, dass das aktuelle Verzeichnis auf das Dokumentverzeichnis gesetzt ist.

Damit das funktioniert, muss allerdings das aktuelle Rechenblatt bereits gespeichert und nach dem Speichern mindestens einmal ausgeführt worden sein. Auch nach dem Wechseln in ein anderes Rechenblatt muss man dieses zunächst ausführen (neu berechnen), damit das Dokumentverzeichnis richtig gesetzt ist.

## Systemverzeichnisse

Unter Windows können einige Systemverzeichnisse mit der Funktion `GetFolderPath()` aus dem Plug-in *MathcadFileAccess* bestimmt werden. Die Funktion muss mit einer Kennzahl aufgerufen werden und liefert dann den zugehörigen Pfad als Zeichenkette:

|                          |                    |                    |                 |               |             |
|--------------------------|--------------------|--------------------|-----------------|---------------|-------------|
| PROGRAMS := 2            | PERSONAL := 5      | FAVORITES := 6     | STARTUP := 7    | RECENT := 8   | SENDTO := 9 |
| STARTMENU := 11          | MUSIC := 13        | DESKTOP := 16      | TEMPLATES := 21 | APPDATA := 26 |             |
| LOCALAPPDATA := 28       | INTERNETCASH := 32 | COOKIES := 33      | HISTORY := 34   |               |             |
| COMMONAPPDATA := 35      | SYSTEM := 37       | PROGRAMFILES := 38 | PICTURES := 39  |               |             |
| COMMONPROGRAMFILES := 43 |                    |                    |                 |               |             |

Hier einige Beispiele:

```
GetFolderPath (PERSONAL) = "C:\Users\kraska\Documents"
GetFolderPath (FAVORITES) = "C:\Users\kraska\Favorites"
GetFolderPath (COOKIES) = "C:\Users\kraska\AppData\Roaming\Microsoft\Windows\Cookies"
GetFolderPath (HISTORY) = "C:\Users\kraska\AppData\Local\Microsoft\Windows\History"
GetFolderPath (COMMONAPPDATA) = "C:\ProgramData"
GetFolderPath (SYSTEM) = "C:\windows\system32"
```

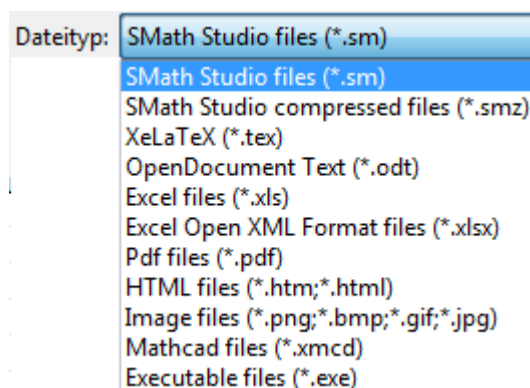
## 4.6. Import/Export

SMath Studio verfügt über umfangreiche Möglichkeiten zum Import oder Export von Rechenblättern, Daten und Bildern. Diese Funktionen sind an verschiedenen Stellen verfügbar:

- Im Hauptmenü unter **Datei > Speichern unter**. Dort kann das ganze Rechenblatt in verschiedenen Formaten exportiert werden.
- Als Schreib- oder Lesefunktionen für bestimmte Dateitypen (Text, Tabellenkalkulation, Bilder, Audio, allgemeine Binärdateien), siehe Abschnitt 6.14.
- Der Image-Bereich (siehe Abschnitt 5.8) kann externe Bild-Dateien anzeigen und Bildinhalte exportieren oder zum Editieren mit passenden Anwendungen öffnen.
- Der Snapshot-Bereich (siehe Abschnitt 4.6.2) kann Ausschnitte des Rechenblatts als Bilddatei exportieren.

### 4.6.1. Export mit „Speichern unter“

Das aktuelle Rechenblatt kann mit **Datei > Speichern unter** exportiert werden. Das Exportformat wird mit Dateityp gewählt:



**SMath Studio XML** (.sm) Standardformat für SMath Studio-Dateien.

**SMath Studio (komprimiert)** (.smz) komprimiertes Format.

**XeLaTeX Dokument** (.tex) Mit XeLaTeX zu kompilierende Textdatei. Das erfordert eine LaTeX-Installation, z.B. MikTeX unter Windows.

**ODF Textdokument** (.odt) Open Document Foundation Text. Kann mit OpenOffice oder LibreOffice oder Word geöffnet werden.

**Excel Tabelle** (.xls) Export des Rechenblatts in eine Excel-Tabelle. Bei Formeln wird, wenn möglich, das numerische Ergebnis exportiert, ansonsten die Darstellung im internen Textformat. Bilder, Tabellen und sonstige Bereiche werden als Bild exportiert.

**Open XML Excel Tabelle** (.xlsx) Export des Rechenblatts in eine Excel-Tabelle. Diese Exportfunktion ist instabil und fehlerhaft (Bug Report [SS-2388](#)). Nutzen Sie das XLS-Format.

**PDF** (.pdf) Das ist die beste Art, PDFs aus SMath Studio-Dokumenten zu erzeugen. Alternativ ginge das auch mit einem PDF-Drucker wie z.B. PDF24 über **Datei > Drucken**. Dort können aber Fehler bei der Skalierung der Bereiche auftreten.

**HTML** Das Rechenblatt wird in eine HTML-Datei exportiert. Formeln und andere Bereiche werden als Bilder exportiert, die in einem separaten Verzeichnis liegen. Diese Exportfunktion ist instabil, es kann zu Programmabstürzen kommen.

**Bild** (.png, .gif, .jpg, .bmp) Export des Rechenblatts in ein Bild (ohne Seitentrennung). Das Standardformat ist PNG, kann aber durch Angabe einer Dateiendung beim Export festgelegt werden.

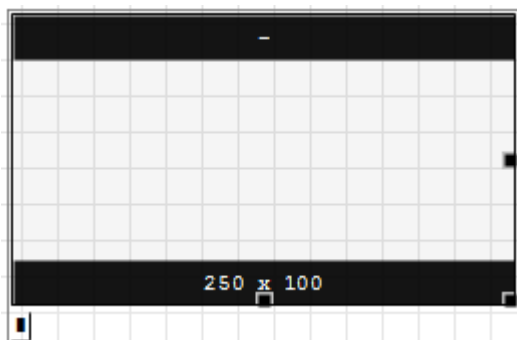
**Ausführbare Datei** (.exe) Export des Rechenblatts als selbstständig lauffähiges Programm (SMath Viewer, siehe Kapitel [11](#)).

**Mathcad Datei** (.xmcd). Export in eine Datei, die mit Mathcad 15 geöffnet werden kann.

#### 4.6.2. Snapshot-Bereich für Bildschirmfotos (Screenshots)

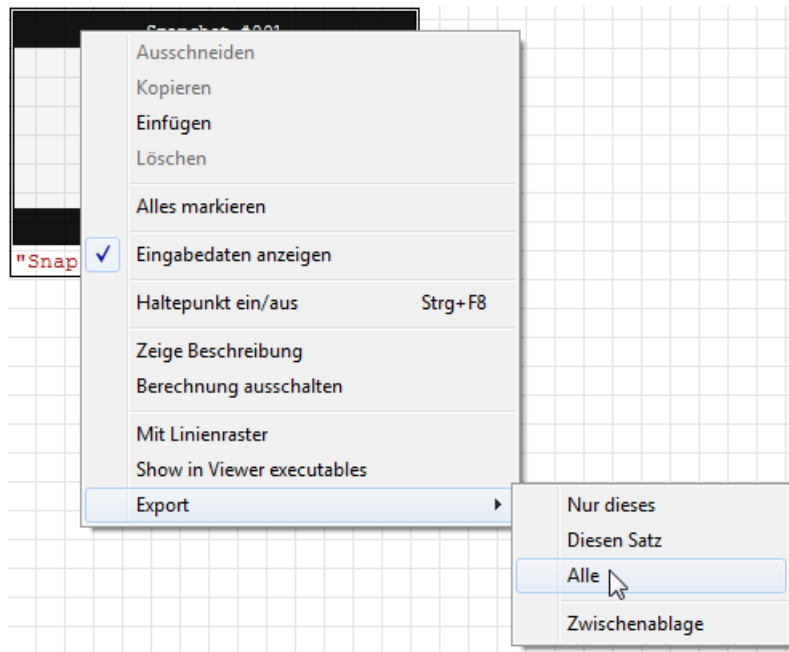
Wenn SMath-Rechnungen in Arbeiten dokumentiert werden sollen, die nicht selbst mit SMath Studio erstellt werden, dann kann man dies mit Screenshots tun. Das vorliegende Handbuch enthält hunderte davon. Der Snapshot-Bereich kann die Erzeugung von solchen Screenshots automatisieren. Der Bereich definiert einen Rahmen, innerhalb dessen der Inhalt des Rechenblatts abfotografiert und in eine Datei exportiert werden kann.

Ein Snapshot-Bereich wird eingefügt mit **Einfügen > Snapshot**



Im Platzhalter ist eine Zeichenkette anzugeben, aus der dann die Dateinamen beim Export gebildet werden. Position und Größe des Bereichs werden wie bei Diagrammen angepasst. Um den Bereich als ganzes zu verschieben, wird er blau markiert. Zur Größenänderung muss der Bereich aktiviert werden (zwischen die Streifen klicken). Dann erscheinen Randmarken zum Ziehen mit der Maus.

Im unteren Balken wird die Bildgröße in Pixeln angezeigt, der obere Balken zeigt den Dateinamen des Bildes. Mehrere Snapshot-Bereiche mit gleichem Text im Platzhalter bilden einen Satz (*set*), innerhalb dessen die Bilder eine dreistellige Nummer erhalten. Diese wird an den Name angehängt (*name\_xxx.png*).



Im Kontextmenü (rechte Maustaste) stehen die folgenden Optionen und Funktionen zur Verfügung:

**Eingabedaten anzeigen:** Damit wird der Platzhalter für den Dateinamen ein- oder ausgeblendet.

**Zeige Beschreibung:** Wie jeder Bereich kann auch der Snapshot-Bereich mit einer Beschreibung versehen werden (siehe Abschnitt 5.13).

**Berechnung ein-/ausschalten:** Damit wird der Bereich aktiviert oder deaktiviert. Ein deaktivierter Bereich wird bei der Erzeugung der Dateinamen nicht mitgezählt. Aktivieren und Deaktivieren von Bereichen ändert also die Dateinamen der erzeugten Screenshots.

**Mit Linienraster:** Das Linienraster des Rechenblatts kann beim Screenshot abgeschaltet werden.

**Show in Viewer executables:** In exportierten Anwendungen zeigen

**Export:** Hier wird der eigentliche Export veranlasst. Es stehen folgende Möglichkeiten zur Verfügung:

**Nur dieses:** Nur das aktuelle Bild exportieren

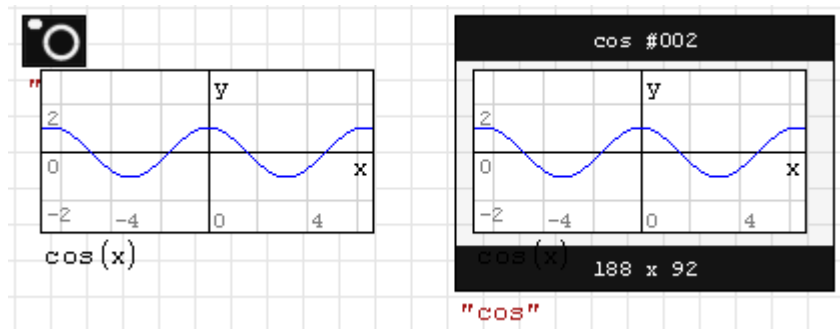
**Diesen Satz:** Alle Bilder mit dem gleichen Namen exportieren. Die Bilder werden fortlaufend nummeriert: *name\_xxx.png*

**Alle:** Alle Snapshot-Bereiche des Rechenblatts exportieren,

**Zwischenablage:** Bild in die Zwischenablage übernehmen (alternativ zum Windows Snipping Tool)

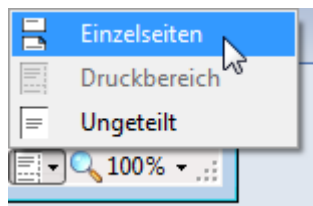
Beim Export öffnet sich ein Dialog zur Auswahl des Exportverzeichnisses, dieses ist vor-eingestellt auf das Dokumentverzeichnis (da, wo das aktuelle Rechenblatt liegt). Daher wird in der Regel einfach eine Bestätigung reichen.

Um die Bearbeitung der unter dem Snapshot-Bereich liegenden Bereiche zu erleichtern, kann der Bereich durch Doppelklick geschrumpft und wieder erweitert werden.



## 4.7. Seitenansicht

In der Statuszeile kann man zwischen verschiedenen Darstellungsformen des Rechenblatts unterscheiden.



**Einzelseiten** Das Rechenblatt wird mit Seitenrändern oben und unten sowie eventuellen Kopf- und Fußzeilen angezeigt. Bereiche, die über den unteren Rand hinausragen würden, werden auf die nächste Seite verschoben.

**Druckbereich** Das Rechenblatt wird in der verfügbaren Druckbreite und mit markierten Seitenumbrüchen angezeigt. Bereiche werden gegebenenfalls durch den Seitenumbruch geteilt.

**Ungeteilt** Das Rechenblatt wird ohne Begrenzung an den Rändern angezeigt, eine Rand- und Umbruchkennzeichnung erfolgt nicht.

Für den Ausdruck oder die PDF-Ausgabe ist die Einzelseitenansicht sinnvoll, denn sie unterstützt einen sinnvollen Seitenumbruch.

Für die Ansicht am Bildschirm sind die beiden anderen Varianten besser geeignet, da sie am Seitenumbruch keinen Platz verschwenken.

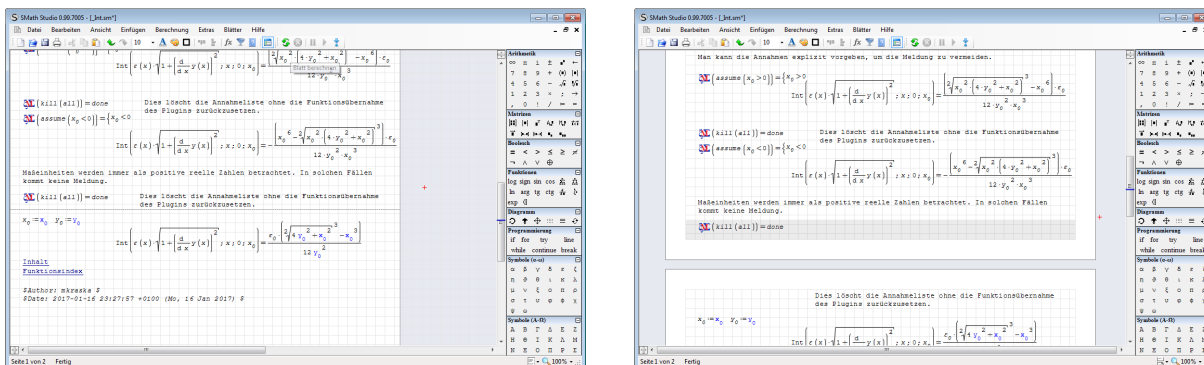


Abbildung 4.4.: Ansichtsmodus: Druckbereich (links), Einzelseiten (rechts).

## 4.8. Drucken

Der System-Druckdialog wird aufgerufen mit **Datei > Drucken** oder **[Strg] P**.

Mit dem Dialog **Datei > Blatteinstellungen** wird das Seitenformat vorgegeben. Zudem gibt es einige Optionen, die das Erscheinungsbild beim Druck gegenüber der Ansicht im Programm modifizieren.

## 4.9. Einstellungen

Die Einstellungen sind auf mehrere Dialoge und Funktionen verteilt.

**Einfügen > Hintergrund...** Es kann ein Hintergrundbild für das aktuelle Rechenblatt eingestellt werden.

**Datei > Blatteinstellungen** Format, Ränder, Kopf- und Fußzeile, Formatierungseinstellungen für den Druck des aktuellen Blatts, Abschnitt 4.9.2.

**Datei > Dateieigenschaften** Metadaten (Autor, Institution usw) für das aktuelle Rechenblatt, Abschnitt 4.9.3. Diese Daten können auch mehrsprachig angegeben werden und sind relevant, wenn die Datei auf dem SMATH-Portal eingestellt werden soll.

Metadaten können auch mit dynamischen Feldern im Rechenblatt abgefragt werden (Abschnitt 5.3)

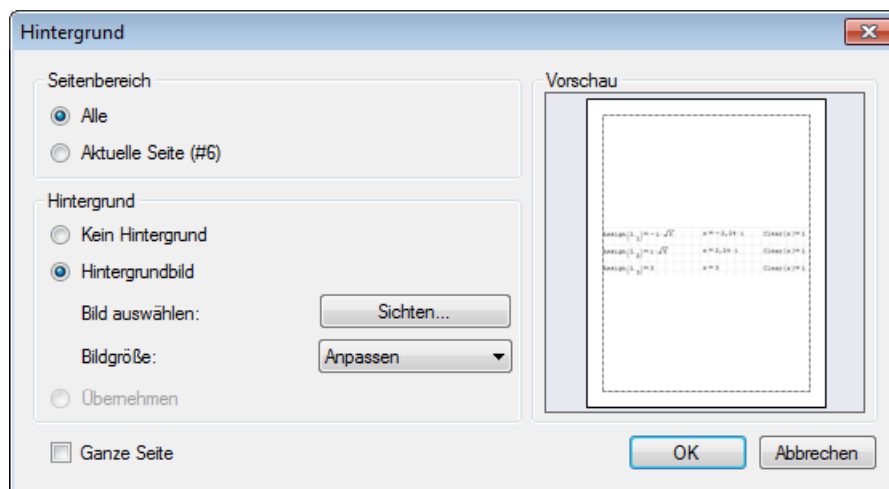
**Extras > Einstellungen** Generelle Voreinstellungen zu Berechnung und Benutzeroberfläche, Abschnitt 4.9.4.

### 4.9.1. Hintergrundbilder



Unter **Einfügen** > **Hintergrund...**

kann man Hintergrundbilder vorgeben. Dabei kann man ein Standardbild für alle Seiten des Dokuments vorgeben und dies dann für einzelne Seiten mit anderen Bildern überschreiben oder abschalten.



Anwendungen sind z.B. eigene Rahmen, Firmenlogos oder ähnliches.

Optionen sind:

- Ganze Seite:
  - deaktiviert: Bild wird in den Bereich innerhalb der Ränder eingepasst
  - aktiviert: Bild wird ins ganze Papierformat eingepasst
- Bildgröße:
  - Anpassen: Das Bild wird ohne Verzerrung so weit skaliert, dass es vollständig in die Fläche passt (zentriert).
  - Strecken: Das Bild wird durch Streckung in beide Richtungen eingepasst, so dass es die ganze Fläche ausfüllt.
  - Ausfüllen: Das Bild wird ohne Verzerrung soweit skaliert, dass es die ganze Fläche ausfüllt. Dabei wird es eventuell beschnitten.

Hintergrundbilder sind nur in der Druckbereichs- oder Einzelseitenansicht (Einstellung in der Statuszeile) sichtbar.

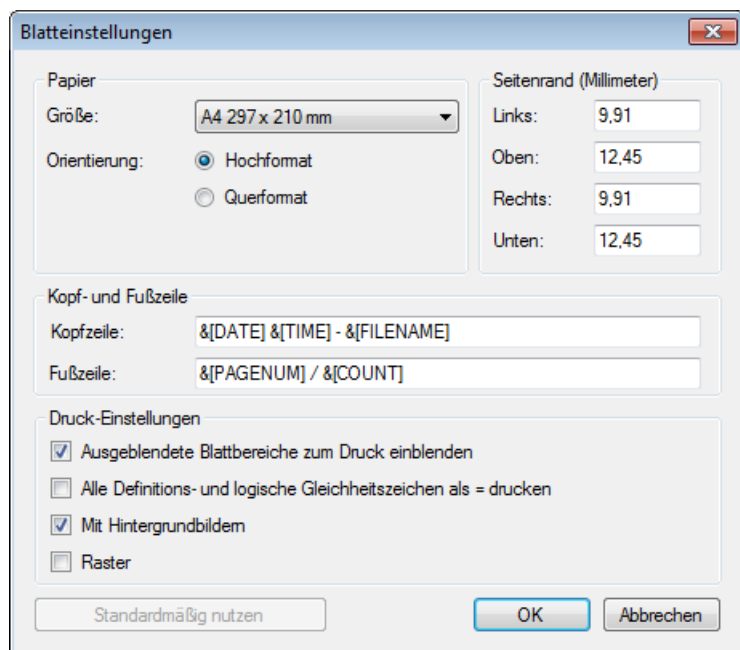
Unter **Datei** > **Blatteinstellungen** kann man entscheiden, ob die Hintergrundbilder mitgedruckt werden sollen.

### 4.9.2. Dialog „Blatteinstellungen“

Aufruf: **Datei** > **Blatteinstellungen**

Die Blatteinstellungen gelten nur für das aktuelle Dokument. Seit der Version 6988 (Februar 2019) kann man die aktuelle Einstellung als Voreinstellung speichern („Standardmäßig nutzen“).

Alternativ zu **Einfügen** > **Kopf- und Fußzeile** können einfache Kopf- und Fußzeilen auch in den Blatteinstellungen definiert werden.



| Platzhalter | Inhalt         |
|-------------|----------------|
| &[DATE]     | Datum          |
| &[TIME]     | Uhrzeit        |
| &[FILENAME] | Dateiname      |
| &[PAGENUM]  | aktuelle Seite |
| &[COUNT]    | Seitenzahl     |

Abbildung 4.5.: Dialog „Blatteinstellungen“ und Feldnamen für dynamische Inhalte in Kopf- und Fußzeile. Werden Kopf- und Fußzeilen mittels **Einfügen** > **Kopf- und Fußzeile** definiert, dann ist dieser Bereich in den Blatteinstellungen inaktiv.

#### Weitere Optionen:

##### Ausgeblendete Blattbereiche zum Drucken einblenden

**Alle Definitions- und logische Gleichheitszeichen als = drucken** Standardmäßig ist := der Zuweisungsoperator und das logische Gleichheitszeichen ist fett. Bei der Druckausgabe kann man aber für beide das normale Gleichheitszeichen benutzen. Davon ist abzuraten, denn damit werden diese Unterschiede im Ausdruck nicht mehr sichtbar.

**Hintergrundbilder drucken** Hintergrundbilder können beim Druck weggelassen werden.

**Raster** Das Gitterraster kann mitgedruckt werden.

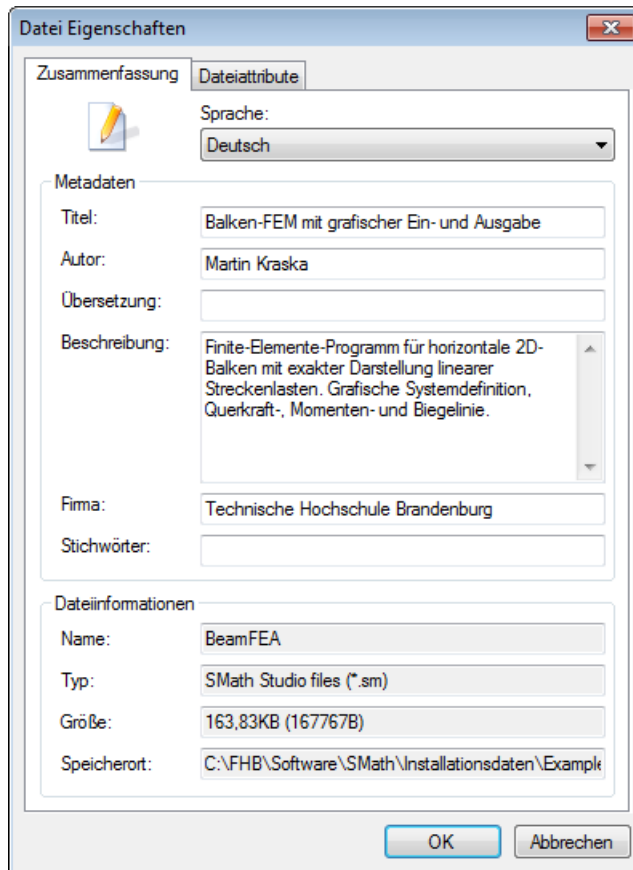
### 4.9.3. Dialog „Datei Eigenschaften“

Aufruf: **Datei** > **Eigenschaften**

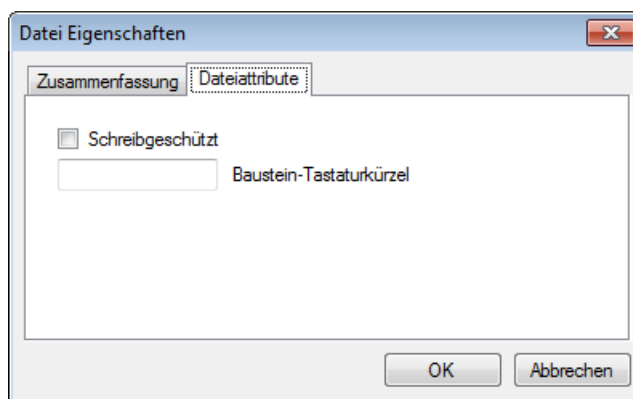
#### Einstellungsseite „Zusammenfassung“

Die Einträge sind alle selbsterklärend. Die Felder sind wichtig, wenn die Datei als Code-Baustein (*snippet*, siehe Abschnitt 10.5) oder Beispiel (*example*, siehe Abschnitt 12.2.2) in der Online-Galerie verwendet werden soll. Dann werden diese Daten im entsprechenden Auswahldialog angezeigt. Beim Export als ausführbare Anwendung (siehe Kapitel 11) werden die englischsprachigen Angabe zu Titel, Autor und Firma im Dateimanager angezeigt.

Zudem kann man mit dynamischen Feldern auf die hier hinterlegten Informationen zugreifen (Abschnitt 5.3).



### Einstellungsseite „Dateiattribute“

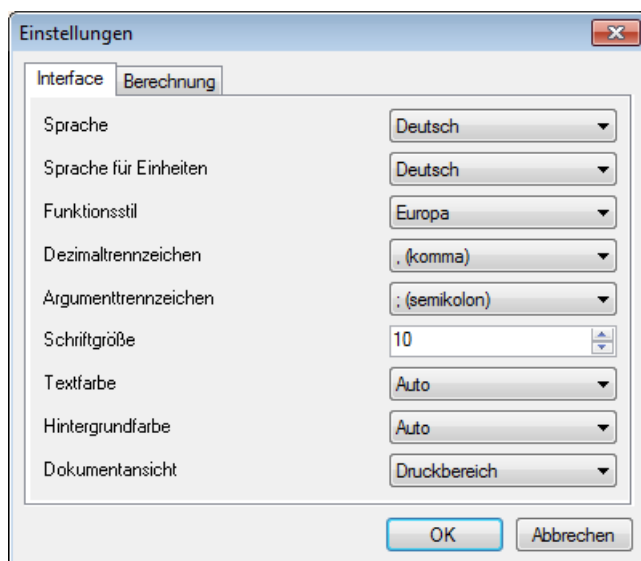


**Baustein-Tastaturkürzel** Name, unter dem das Rechenblatt in ein anderes Rechenblatt eingefügt werden kann (Code-Baustein, siehe Abschnitt 10.5). Dafür muss die Datei im Verzeichnis „Snippets“ des Installationsverzeichnis liegen.

#### 4.9.4. Dialog „Einstellungen“

Aufruf: **Extras**> **Einstellungen**

## Einstellungsseite „Interface“



**Sprache** Sprache der Bedienoberfläche. Eine breite Auswahl wird angeboten. Die vorliegende Anleitung setzt „Deutsch“ voraus.

**Sprache für Einheiten** Darstellung der Maßeinheiten. Neben dem Standard „English“ gibt es deutsche, dänische und russische Versionen (bei letzterer kommen dann kyrillische Buchstaben vor). Die vorliegende Anleitung setzt „Deutsch“ voraus.

**Funktionsstil** Steuert die Namen der Winkelfunktionen und ihrer Umkehrungen. „Weltstandard“ bedeutet  $\text{asin}$ ,  $\text{acos}$ ,  $\text{cot}$  usw. „Europa“ bedeutet  $\text{arcsin}$ ,  $\text{arccos}$ ,  $\text{ctg}$  usw.

**Dezimaltrennzeichen** Trennzeichen vor den Nachkommastellen von Dezimalbrüchen. Für deutsche Benutzer ist „Komma“ angemessen. Alternativen sind „Punkt“, wie es im Englischen üblich ist, oder „Semikolon“, was eine nirgendwo gebräuchliche Variante ist.

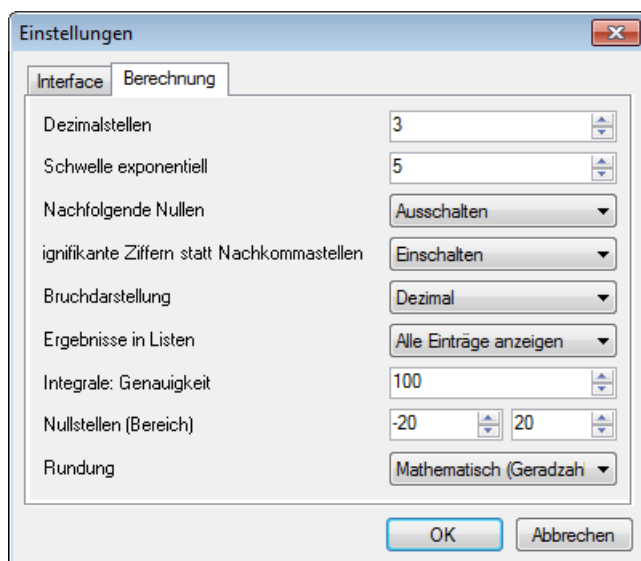
**Argumenttrennzeichen** Trennzeichen zwischen den Argumenten von Funktionen mit mehreren Parametern. Wenn das Dezimaltrennzeichen auf „Komma“ steht, sollte hier „Semikolon“ verwendet werden.

**Schriftgröße** Voreinstellung der Schriftgröße für neu eingefügte Text- oder Formelbereiche. Die Voreinstellung ist 10 Punkt. Die dargestellte Größe kann auch durch Mausrad-Zoom (nach Klick auf den Hintergrund des Rechenblatts) verstellt werden.

**Textfarbe, Hintergrundfarbe** Voreinstellung für neu eingefügte Text- oder Formelbereiche. Es stehen vordefinierte Grundfarben und die Voreinstellung „Auto“ zur Verfügung. Es gibt keinen guten Grund, hier rumzuspielen.

**Dokumentansicht** Voreinstellung für die Anzeige von Seitenumbruch und Rand auf dem Rechenblatt. Diese kann für die laufende Sitzung unten rechts in der Statuszeile geändert werden.

## Einstellungsseite „Berechnung“



**Dezimalstellen** Voreinstellung für die angezeigten Nachkommastellen oder signifikanten Ziffern (0...15). Kann im Kontextmenü für markierte Formelbereiche geändert werden. Gegebenenfalls wird wissenschaftlich gerundet. Nullen am Ende des Nachkommateils werden nur angezeigt, wenn dies im Kontextmenü des jeweiligen Formelbereichs angewählt wird.

**Schwelle exponentiell** Voreinstellung für den Übergang auf abgetrennte Zehnerpotenzen (0...15). Gilt für negative und positive Exponenten. Kann im Kontextmenü für einzelne Bereiche geändert werden.

**Nachfolgende Nullen** Voreinstellung für die Anzeige nachfolgender Nullen. In den Handbuchbeispielen wird dies standardmäßig ausgeschaltet, damit ganzzahlige Werte nicht mit Nachkommastellen angezeigt werden.

**Signifikante Ziffern statt Nachkommastellen** Voreinstellung für die Anzeigegenauigkeit (feste Zahl von Nachkommastellen oder signifikante Ziffern). In den Handbuchbeispielen ist die Voreinstellung „Einschalten“, also Vorgabe der Zahl signifikanter Ziffern.

**Bruchdarstellung** „Dezimal“, „Mit Bruchstrich“, „Auto“,

**Ergebnisse in Listen** „Alle Einträge anzeigen“, „Wiederholte Werte nur einmal“, siehe Abschnitt 6.9.1,

**Integrale: Genauigkeit** Intervallzahl bei numerischer Integration, in 50er Schritten einstellbar.

**Nullstellen (Bereich)** Intervall, in welchem mit der Funktion `solve()` nach Nullstellen gesucht wird.

Die hier gezeigte Einstellung besagt, dass Zahlen auf drei signifikante Stellen gerundet werden und dabei nachfolgende Nullen bei Dezimalbrüchen weggelassen werden.

# 5. Das Rechenblatt

Jede SMath-Datei enthält ein Rechenblatt, also ein zweidimensionales Gebiet, auf dem Bereiche unter- und nebeneinander frei angeordnet werden können. Es gibt standardmäßig folgende Bereichstypen:

- Formel (Abschnitt 6.1)
- Text (Abschnitt 5.5)
- 2D-Diagramm (Abschnitt 8.2)
- 3D-Diagramm (Abschnitt G.4)
- Bild (Rasterbild, statisch)
- Blattbereich (zur Gruppierung anderer Bereiche, ausblendbar, Abschnitt 5.14)
- Trennlinien (erzeugen einen Seitenumbruch, Abschnitt 5.14)

Darüber hinaus können externe Plug-ins weitere Bereichstypen beisteuern. Wichtige Beispiele sind:

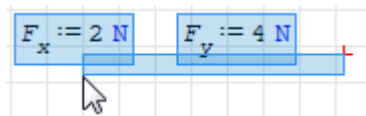
- Image (vielseitig verwendbarer Bildbereich aus dem Plug-in *Image Region*, Abschnitt 5.8). Dieser Bereich eignet sich gut zur Erstellung editierbarer eingebetteter Skizzen.
- X-Y Plot, ein Diagrammbereich, der viel flexibler als das Standard-2D-Diagramm ist (Abschnitt 8.3).
- Erweiterter Textbereich (*writer region*), der flexibler als der Standard-Textbereich ist (Abschnitt 5.6).
- Steuerelemente wie Schieberegler, Schaltknöpfe und Auswahllisten, Abschnitt 5.12,
- Tabellen, Abschnitt 5.9
- Modeller, Abschnitt 8.4

Wie jeder Bereich hat auch das Rechenblatt selbst ein Kontextmenü (mit der rechten Maustaste aufrufen). Dieses bietet Zugriff auf elementare Bearbeitungsfunktionen für das aktuelle Rechenblatt.

## 5.1. Bereiche bearbeiten

Bereiche können auf dem Rechenblatt beliebig, auch überlappend, angeordnet werden.

Bevor man sie verschieben, löschen oder kopieren kann, müssen sie markiert werden. Das geschieht durch Ziehen mit der Maus, dabei werden die Bereiche hellblau eingefärbt.



Ausnahme: Bei gedrückter **Strg**-Taste kann man mit der linken Maustaste einen noch nicht markierten Bereich und anderswo auf das Rechenblatt ziehen. Die Kopie ist anschließend hellblau markiert.

Die Auswahl kann bei gehaltener Steuerungstaste **Strg** mit der Maus erweitert oder reduziert werden.

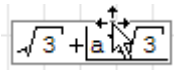
Anschließend kann man die markierten Bereiche

- Verschieben (durch Ziehen mit der Maus oder mit den Pfeiltasten)
- Löschen (**Kontextmenü** > **Löschen** oder mit der Taste **Entf**)
- Ausschneiden, also Löschen und in die Zwischenablage übernehmen (**Kontextmenü** > **Ausschneiden** oder **Strg**-**X**)
- Kopieren (**Kontextmenü** > **Kopieren** oder mit **Strg**-**C**)
- Duplizieren (durch Ziehen bei gedrückter **Strg**-Taste)

Ausgeschnittene oder kopierte Bereiche können an der Einfügemarke ins Rechenblatt eingefügt werden (**Strg**-**V** oder **Kontextmenü** > **Einfügen**).

Die Befehle „Ausschneiden“, „Löschen“, „Einfügen“ und „Kopieren“ finden sich auch im Hauptmenü unter **Bearbeiten**.

Bereiche, die den Mausfokus haben (aktiviert sind), können auch an ihrem Rand mit der Maus verschoben werden.



Den **vertikalen Abstand zwischen Bereichen** kann man ändern, indem man den Cursor dazwischensetzt und mit **↵** Zwischenraum einfügt oder mit **⇐** Zwischenraum entfernt.

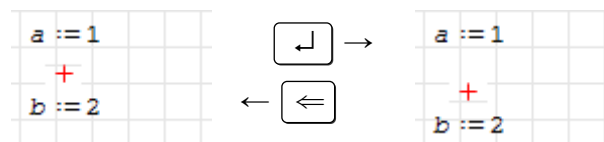


Abbildung 5.1.: Zwischenraum vergrößern und verkleinern

## 5.2. Kopf- und Fußzeilen

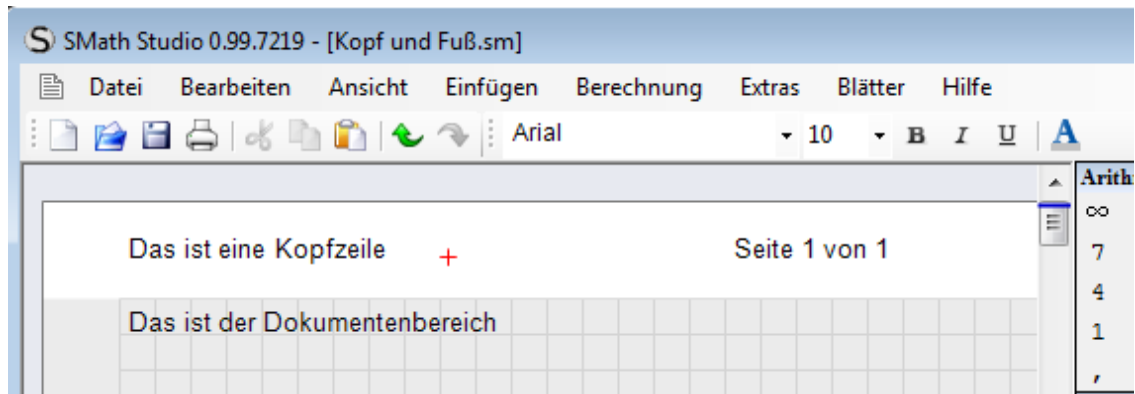
Die Seiten eines Dokuments können mit Kopf- und Fußzeilen versehen werden:

**Einfügen** > **Kopf- und Fußzeile** > **Kopfzeile** oder **Fußzeile**

Damit wird der gesamte obere und untere Rand (wie unter **Datei** > **Blatteinstellungen** festgelegt) editierbar. Änderungen werden für alle Seiten übernommen.

Alternativ kann man die gerade in der Bearbeitung befindliche Kopf- oder Fußzeile zur Ausnahme erklären (**Einfügen** > **Kopf- und Fußzeile** > **Anders gestalten**). Dann betreffen die Änderungen nur diese Seite.

In der Einzelseitenansicht kann man Kopf- und Fußzeilen zum Editieren doppelklicken. Durch Doppelklick außerhalb kann man diesen Modus wieder verlassen.

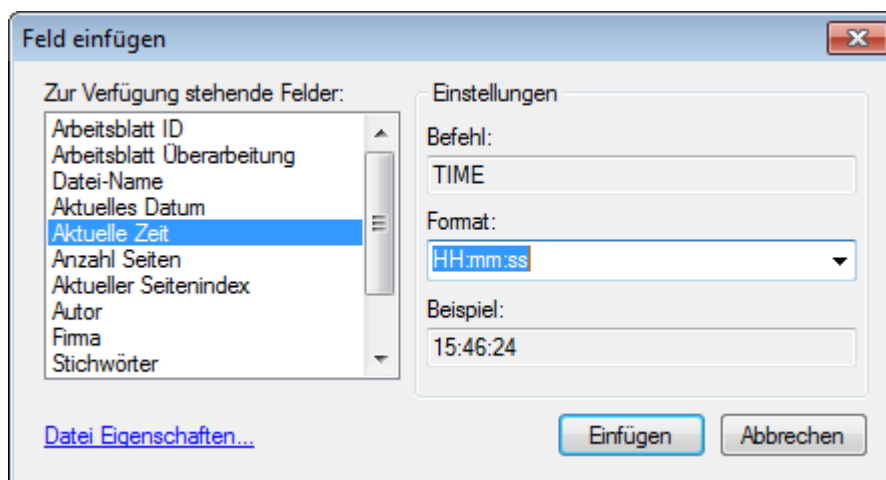


Für Seitenzahlen und ähnliches können dynamische Felder verwendet werden.

### 5.3. Dynamische Felder

Dynamische Felder sind spezielle Variablen, deren Inhalt sich aus Systemeigenschaften (Datum, Uhrzeit, Dateiname) oder den unter **Datei**> **Eigenschaften** hinterlegten Metadaten ergibt.

**Einfügen**> **Feld...**



### 5.4. Formelbereich

In Formelbereichen spielt sich die eigentliche Rechnung ab, hier wird definiert und dokumentiert, was zu berechnen ist. Auf die mathematischen Möglichkeiten wird in Kapitel 6 eingegangen.

Formelbereiche entstehen auf dem Rechenblatt an der Cursorposition  $+$ , wenn Buchstaben oder Operatoren mit der Tastatur oder aus Menüs oder der Seitenleiste einfügt. Der Inhalt von Formelbereichen wird mathematisch formatiert und ausgewertet.

Formelbereiche können als Ganzes einheitlich formatiert werden (der Bereich muss dazu hellblau markiert oder aktiv sein, also Mausfokus haben). Die folgenden Werkzeuge stehen dafür in der Werkzeugleiste zur Verfügung:





Schriftgröße, Textfarbe, Hintergrundfarbe, Rahmen ein/aus

Die Attribute Schriftart, Fett, Kursiv und Unterstreichung sind in Formelbereichen unwirksam.

Formelbereiche können in Textbereiche eingefügt werden und sind dort auch mathematisch aktiv.

## 5.5. Textbereich

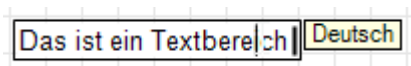
Ein Textbereich enthält Text, der nicht mathematisch formatiert und nicht mathematisch ausgewertet wird. Er ist für kürzere Kommentare geeignet und kann nur einheitlich formatiert werden. Es ist jedoch möglich, weitere Bereiche einzubetten, z.B.

- Formelbereiche (diese sind dann auch aktiv)
- Textbereiche, z.B. für abweichende Formatierung

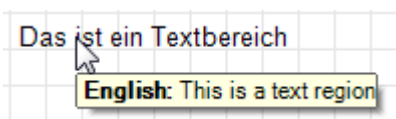
**Textbereiche erzeugen:**

- " (Anführungszeichen) am Cursor im Rechenblatt eintippen
- Text eintippen, der Leerzeichen enthält (bevor Rechenoperatoren auftreten, sonst wird der Bereich endgültig als Formel betrachtet)
- **Einfügen** > **Textbereich** aufrufen

Im Kontextmenü kann man die **Sprache auswählen**. Bei neuen Textbereichen ist sie auf die Sprache der Benutzeroberfläche voreingestellt und wird beim Schreiben angezeigt:



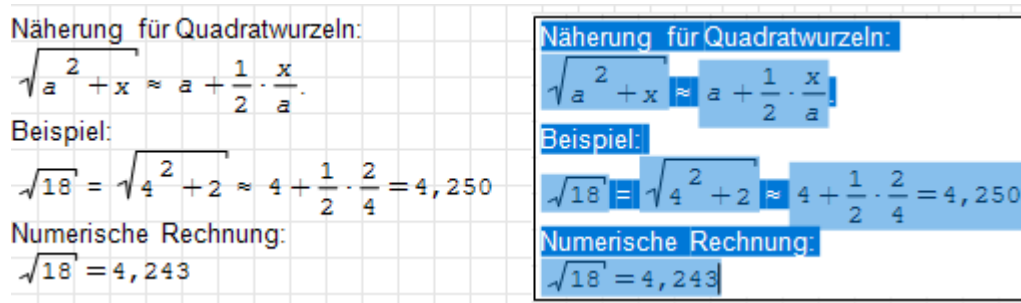
Steht der Mauszeiger auf einem nicht aktivierten (angeklickten) Textbereich, erscheinen die alternativen Sprachversionen, soweit sie angegeben sind:



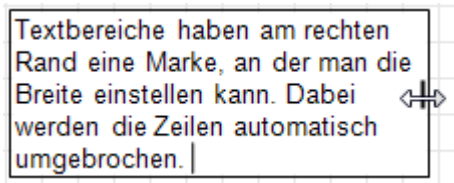
Textbereiche können als Ganzes einheitlich formatiert werden:



- Schriftart: **Werkzeugleiste** >
- Schriftgröße: **Werkzeugleiste** >
- Schriftstil, z.B. **fett** ( $\text{Strg} - \text{B}$ ), *kursiv* ( $\text{Strg} - \text{I}$ ) oder unterstrichen ( $\text{Strg} - \text{U}$ )
- Rahmen an/aus: **Werkzeugleiste** >
- Hintergrundfarbe: **Werkzeugleiste** >
- Textfarbe: **Werkzeugleiste** >

In den laufenden Text können Formelbereiche eingefügt werden mit **Einfügen** > **Formel**.



Textbereiche haben am rechten Rand eine Marke, an der man die Breite einstellen kann. Dabei werden die Zeilen automatisch umgebrochen.



Ein Zeilenumbruch kann auch mit  -  erzwungen werden.

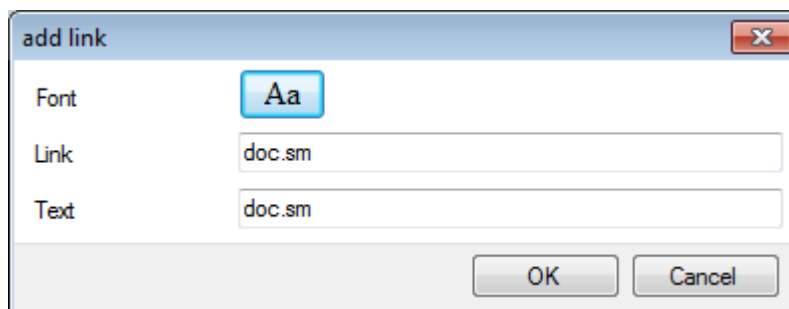
### 5.5.1. Textbereiche mit Verweis (Hyperlink)

Textbereiche können als Verweis (Hyperlink) auf andere SMath-Dokumente dienen. Das interaktive Handbuch (siehe Abschnitt 4.3) in SMath Studio macht davon umfangreich Gebrauch. Verweise sind wie auch im Internet üblich, durch Unterstreichung und blaue Schrift gekennzeichnet. Aktiviert werden Verweise durch **Strg** + Klick. Das aktuelle Dokument wird dann durch das Verweisziel ersetzt, bei eventuellen Änderungen wird vorheriges Speichern angeboten.

Soll auf ein anderes Ziel als eine SMath-Datei verwiesen werden, kann man den Hyperlink-Bereich (siehe Abschnitt 5.10) verwenden.

#### Textbereich mit Verweis einfügen

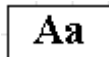
Dafür bietet das Plug-in *Text Region Utilities* einen extra Punkt: **Einfügen** > **Textbereich (Utilities)** > **add a new link to a worksheet**. Dies öffnet einen Dialog, in welchem das Verweisziel und der sichtbare Text eingegeben werden können.



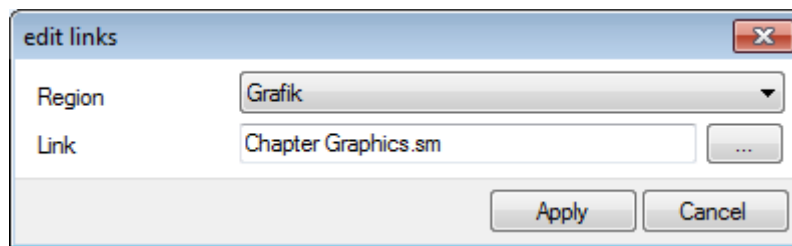
#### Verweisziel für bestehenden Textbereich angeben

Das Verweisziel kann für bestehende Textbereiche nicht direkt geändert werden. Daher wird auch hier der Umweg über den Toolbox-Bereich genommen.

1. Einfügen des Toolbox-Bereichs mit **Einfügen** > **Textbereich (Utilities)** > **insert a toolbox**



2. Kontextmenü der Toolbox öffnen (rechte Maustaste) und den Eintrag „edit links...“ auswählen:
3. Der Dialog „edit links“ bietet eine Auswahlliste mit allen bestehenden Textbereichen. Im Eingabefeld „Link“ kann das Verweisziel angegeben werden (muss eine SMath-Datei, .sm sein). Mit dem Knopf rechts daneben kann das Verweisziel interaktiv ausgewählt werden.



4. Die Toolbox kann nun wieder gelöscht werden.

## 5.6. Erweiterter Textbereich (Writer region)

Der Writer-Bereich war lange Zeit die einzige Möglichkeit, Text flexibel und bequem zu formatieren. Er wird durch das *Plug-in Writer Region* bereitgestellt. Seit 2019 hat der Standard-Textbereich aufgeholt. Weiterhin bestehende Unterscheidungsmerkmale des erweiterten Textbereichs (*Writer region*) sind:

- flexible Formatierung auch einzelne Worte oder Buchstaben mit Hilfe einer eigenen Werkzeugleiste,
- eingebettete Hyperlinks, auch mehrere mit unterschiedlichen Sprungzielen,
- eingebettete Formeln sind inaktiv.
- Zeilenabstand und horizontale Ausrichtung sind einstellbar.

Er wird eingefügt mit **Einfügen** > **Writer region** oder dem Tastenkürzel **Strg**-**W**

Text kann wie im Standard-Textbereich eingefügt werden. Es gelten die Tastenkürzel

**Strg**-**↑** Zeilenumbruch

**Strg**-**G** vorhergehenden Buchstaben griechisch machen (siehe Abschnitt 6.1.2)

Einige Einstellungen können über das Kontextmenü vorgenommen werden:

**Sprache** Der Writer-Bereich unterstützt mehrsprachige Dokumente.

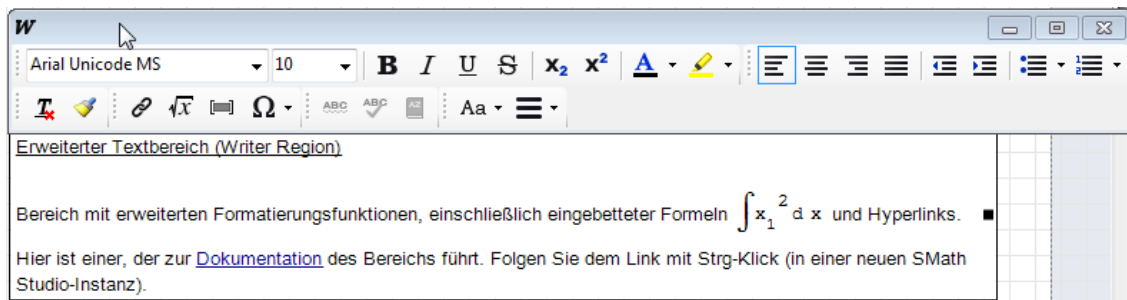
**Page alignment** (Ausrichtung). Links, rechts, zentriert, außerhalb des Druckbereichs

**Wrap** (Zeilenumbruch/Textbreite) Ausgeschaltet, Seitenbreite, Einstellbar (durch Ziehen an der Randmarke rechts)


**Line spacing** (Zeilenabstand) Voreinstellung 1,15 (diese kann angepasst werden)

**Settings** Ruft die Werkzeugleiste für die Formatierungswerkzeuge auf (alternativ mit


Strg--S )

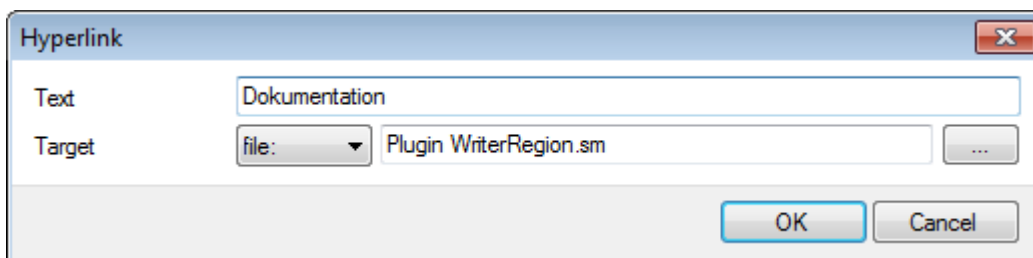


### 5.6.1. Formeln und Hyperlinks

**Formeln** werden eingefügt mit Strg--M . Das erzeugt einen Platzhalter, in welchem man wie in einem Formelbereich editieren kann.

Durch Doppelklick werden solche Formeln zum Editieren geöffnet.

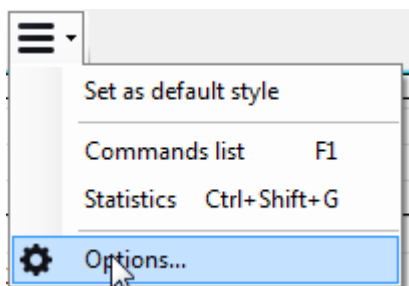
**Hyperlinks** werden mit Strg--H eingefügt oder editiert. Wenn schon Text markiert ist, wird dieser als Text übernommen, dann sind noch der Typ und das Ziel anzugeben.

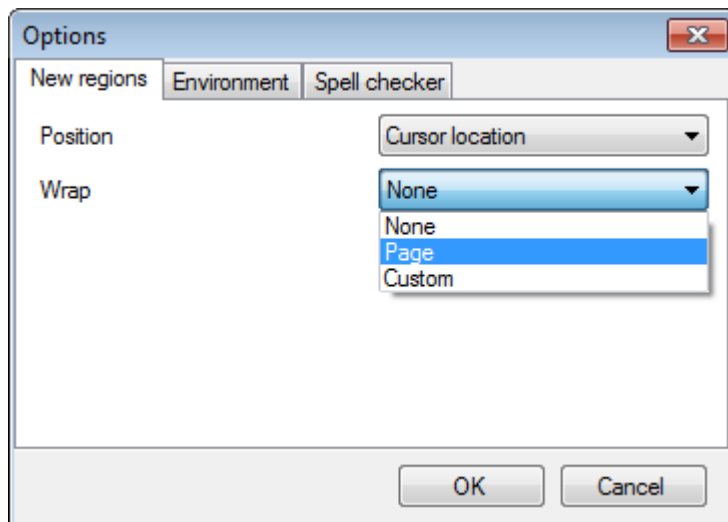


### 5.6.2. Voreinstellungen

Es empfiehlt sich, die Voreinstellungen für den Writerbereich an die eigenen Vorlieben anzupassen:

Writer-Werkzeugleiste > More.. > Options





**Position** Einfügemarke neuer Writerbereiche (an der Einfügemarke oder über die ganze Seitenbreite)

**Wrap** Bereichsbreite und Zeilenumbruch

**None** Bereich so breit wie der Text (kein Zeilenumbruch)

**Page** Bereich so breit wie die Seite (mit Zeilenumbruch)

**Custom** Bereichsbreite einstellbar durch Ziehen an der Randmarke (mit Zeilenumbruch)

## 5.7. Diagrammbereiche

Brauchbar sind in SMATH Studio der Standard 2D-Diagrammbereich, das Plug-in X-Y Plot sowie die Maxima-Diagrammbereiche<sup>1</sup>. Die Möglichkeiten und Verwendung der Diagrammbereiche werden im Kapitel 8 behandelt. Hier wird nur angegeben, wie man sie erzeugt.

---

<sup>1</sup>Es gibt noch weitere Bereiche, wie den MS Chart-Bereich, allerdings liegen da keine Anwendungserfahrungen vor.

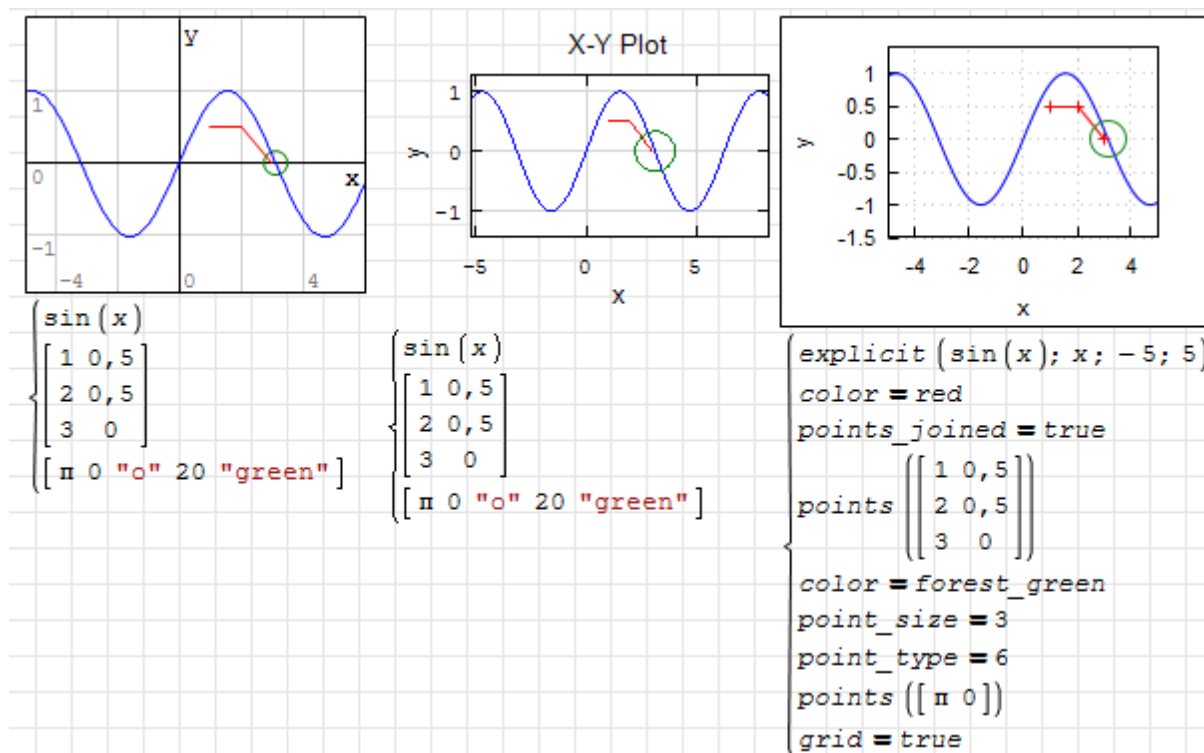


Abbildung 5.2.: 2D-Diagrammbereiche. Links: Standard, Mitte: Plug-in X-Y Plot, Rechts: Maxima Draw2D-Bereich

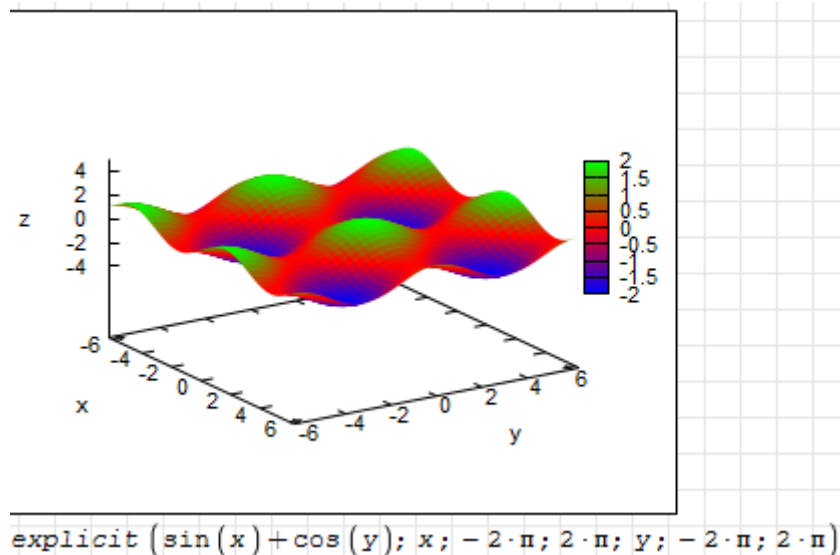


Abbildung 5.3.: Maxima Draw3D-Diagrammbereich

Diagrammbereiche werden erzeugt mit

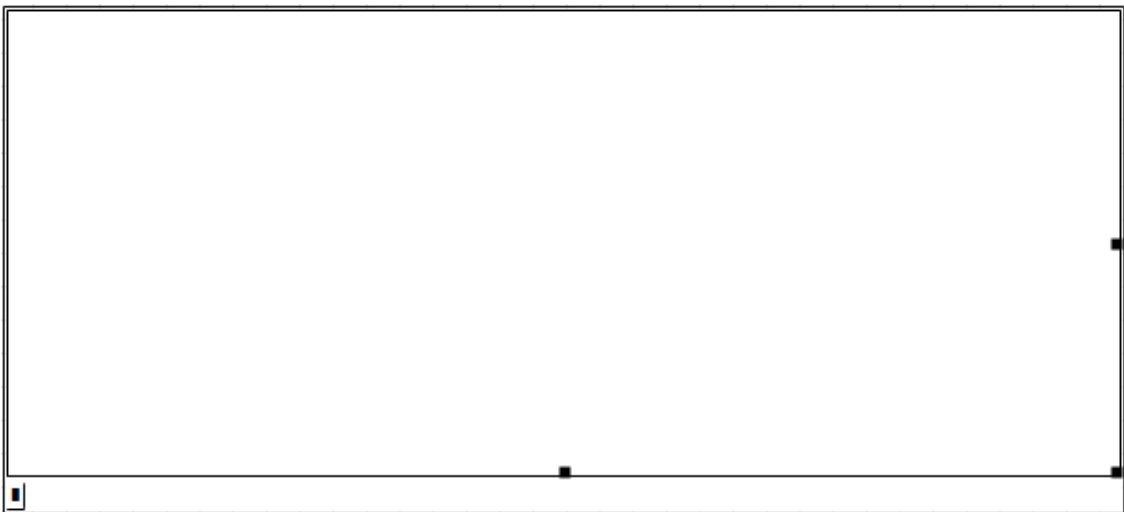
- Standard 2D **Einfügen > Diagramm > Zweidimensional** oder **@**
- Standard 3D **Einfügen > Diagramm > Dreidimensional**
- Maxima 2D **Einfügen > Maxima > Draw2D()**
- Maxima 3D **Einfügen > Maxima > Draw3D()**
- X-Y Plot **Einfügen > Diagramm > X-Y Plot**

## 5.8. Image-Bereich

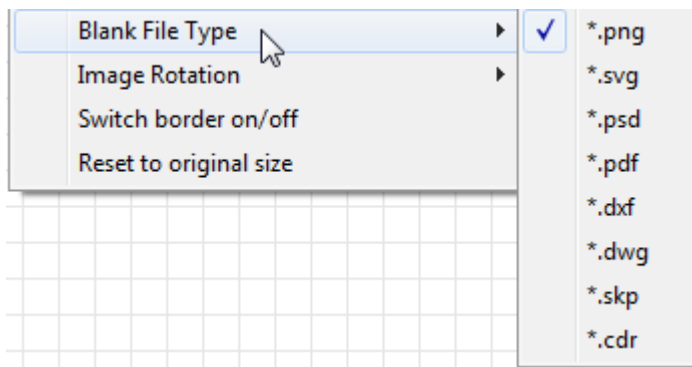
Der Image-Bereich bietet die Möglichkeit

- Vektor- oder Rastergrafiken ins Dokument einzubetten
- Solche eingebetteten Grafiken zu importieren, exportieren oder zum Editieren mit den unter Windows dafür eingestellten Programmen zu öffnen
- Matrix-Variablen als Rasterbild anzuzeigen

Dieser wird vom Plug-in *ImageRegion* bereitgestellt und mit **Einfügen** > **Image** eingefügt.



Anschließend kann man mit Rechtsklick das Dateiformat des anzulegenden Bildes wählen. Voreingestellt ist das PNG-Format:



Bei Doppelklick auf den leeren Bereich öffnet sich das Windows-Standardprogramm für das gewählte Dateiformat. Das wird in Abschnitt 3.7 am Beispiel des PNG-Formats und des Programms Paint erläutert.

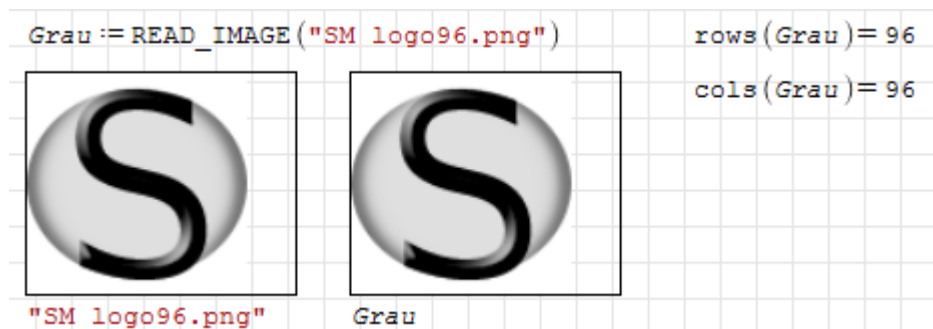
Im Kontextmenü stehen weitere Funktionen bereit

| Option                        | Funktion  |
|-------------------------------|---|
| <b>Eingabedaten anzeigen</b>  | Einblenden des Platzhalters (nur sinnvoll, wenn Dateien oder Matrizen dargestellt werden sollen)                                      |
| <b>Open With</b>              | Öffnen der eingebetteten Bilddatei mit einem wählbaren Programm   |
| <b>Import</b>                 | Bild aus einer Datei lesen und einbetten  |
| <b>Export</b>                 | Bild in eine Datei schreiben  |
| <b>Blank File Type</b>        | Typ des einzubettenden Bildes wählen (PNG, SVG und viele andere sind möglich). Wird nur angeboten, solange der Bereich noch leer ist. |
| <b>Image Rotation</b>         | Drehen des Bildes um Vielfache von 90°  |
| <b>Switch border on/off</b>   | Rahmenlinie ein-/ausschalten  |
| <b>Reset to original size</b> | Ursprünglichen Maßstab wiederherstellen (Rasterbilder in natürlicher Auflösung darstellen)  |

Tabelle 5.1.: Kontextmenü-Optionen des Image-Bereichs

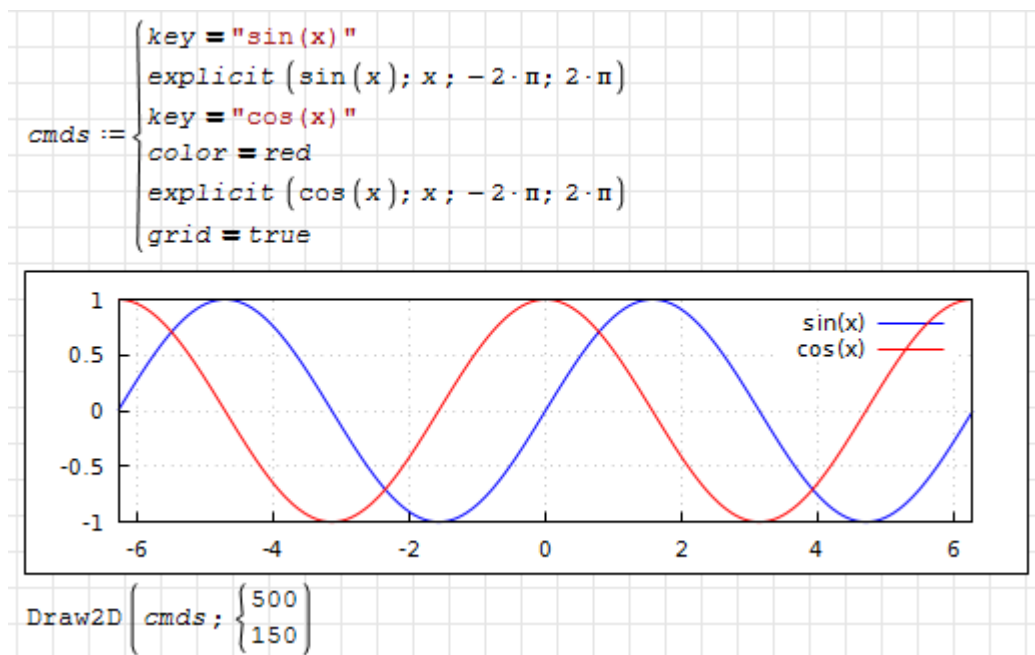
Der Image-Bereich kann auch dynamische Inhalte anzeigen:

- Externe Dateien, dafür ist der Dateiname im Platzhalter anzugeben,
- Als Matrix gegebene Rasterbilder, dafür ist der Variablenname anzugeben.



Die Funktionen `Draw2D()` und `Draw3D()` aus dem *Maxima*-Plug-in erzeugen Bilddateien, die mit dem Image-Bereich angezeigt werden können. Sie liefern als Rückgabewert den Dateinamen, der in den Platzhalter des Image-Bereichs eingetragen werden kann. Anschließend muss man noch mit **Kontextmenü** > **Reset to original size** die Größe des Bereichs an das Bild anpassen oder kann es alternativ beliebig skalieren. Näheres zu diesen Diagrammfunktionen finden Sie in Abschnitt 8.1.





## 5.9. Tabellenbereich

Achtung! Dieser Bereichstyp befindet sich in der Entwicklung. Es ist nicht sicher, dass Tabellen von späteren Versionen noch gelesen werden können.

Das Plug-in *TableRegion* stellt einen Tabellenbereich bereit, der für die Anzeige von Text, Zahlen oder SMATH-Formeln genutzt werden kann. Besondere Stärken sind:

- Darstellbarkeit auch komplex verschachtelter Matrizen und Listen,
- Umgang mit Maßeinheiten,
- Zahlenformatierung,
- Darstellbarkeit von Formeln oder Symbolen wie im Formelbereich.

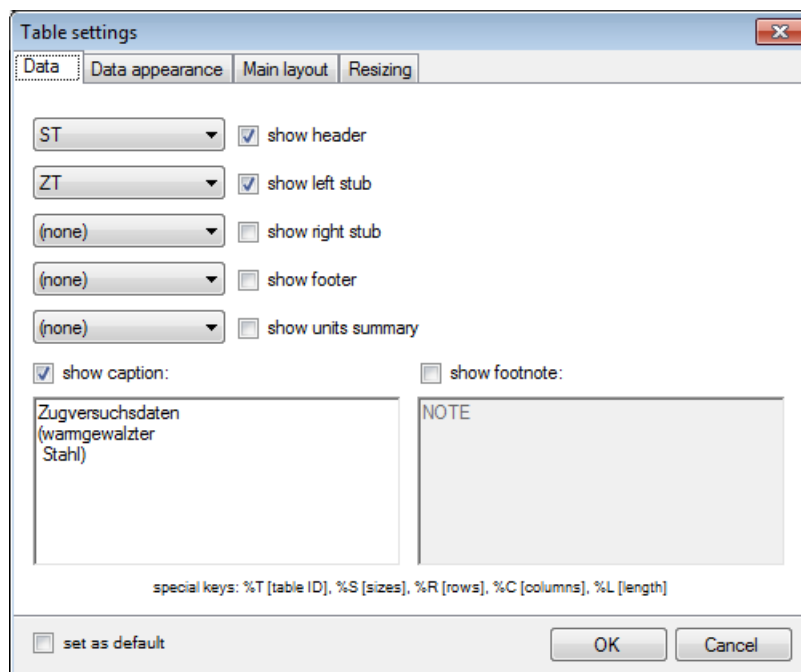
Eingefügt wird er mit **Einfügen** > **Table** oder mit  $\uparrow$  - Strg - T



In den Platzhalter trägt man eine Datenmatrix ein. Für die Zeilentitel und Spaltentitel kann man Vektoren und Matrizen definieren, die dann im Einstellungsdialog ausgewählt werden können.

| $ZT := \begin{bmatrix} \text{"Test1"} \\ \text{"Test2"} \\ \text{"Test3"} \end{bmatrix}$ | $ST := \begin{bmatrix} \text{"S:R.e"} & \text{"S:A.G"} \\ \text{"S:MPa"} & \text{"\%"} \end{bmatrix}$ | $Daten := \begin{bmatrix} 234 & 12 \\ 236 & 13 \\ 324 & 22 \end{bmatrix}$ | <table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th></th> <th>R<sub>e</sub></th> <th>A<sub>G</sub></th> </tr> <tr> <th></th> <th>MPa</th> <th>%</th> </tr> </thead> <tbody> <tr> <td>Test1</td> <td>234</td> <td>12</td> </tr> <tr> <td>Test2</td> <td>236</td> <td>13</td> </tr> <tr> <td>Test3</td> <td>324</td> <td>22</td> </tr> <tr> <td colspan="3">Zugversuchsdaten<br/>(warmgewalzter<br/>Stahl)</td> </tr> </tbody> </table> |  | R <sub>e</sub> | A <sub>G</sub> |  | MPa | % | Test1 | 234 | 12 | Test2 | 236 | 13 | Test3 | 324 | 22 | Zugversuchsdaten<br>(warmgewalzter<br>Stahl) |  |  |
|--|---|---|---|--|----------------|----------------|--|-----|---|-------|-----|----|-------|-----|----|-------|-----|----|--|--|--|
|  | R <sub>e</sub>  | A <sub>G</sub>  |   |  |                |                |  |     |   |       |     |    |       |     |    |       |     |    |  |  |  |
|  | MPa   | %   |   |  |                |                |  |     |   |       |     |    |       |     |    |       |     |    |  |  |  |
| Test1  | 234   | 12  |   |  |                |                |  |     |   |       |     |    |       |     |    |       |     |    |  |  |  |
| Test2  | 236   | 13  |   |  |                |                |  |     |   |       |     |    |       |     |    |       |     |    |  |  |  |
| Test3  | 324   | 22  |   |  |                |                |  |     |   |       |     |    |       |     |    |       |     |    |  |  |  |
| Zugversuchsdaten<br>(warmgewalzter<br>Stahl)   |   |   |   |  |                |                |  |     |   |       |     |    |       |     |    |       |     |    |  |  |  |
|  |   |   | Daten   |  |                |                |  |     |   |       |     |    |       |     |    |       |     |    |  |  |  |

Doppelklick öffnet den Einstellungsdialog, der auf vier Seiten umfangreiche Anpassungen ermöglicht. Voraussetzung ist, dass bereits eine Datenmatrix angezeigt wird.



**Table settings > Data:** Auf der Seite „Data“ kann man folgende Einstellungen vornehmen (bei einer Datenmatrix mit  $m$  Zeilen und  $n$  Spalten):

**header** Spaltentitel (oben), Matrix mit einer oder  $n$  Spalten und mindestens einer Zeile (im Beispiel 2 Spalten mit 2 Zeilen)

**left stub** Zeilentitel (links), Matrix mit einer oder  $m$  Zeilen und mindestens einer Spalte (im Beispiel 1 Spalte mit 3 Zeilen)

**right stub** Zeilentitel (rechts), Matrix mit einer oder  $m$  Zeilen und mindestens einer Spalte

**footer** Spaltentitel (unten), Matrix mit einer oder  $n$  Spalten und mindestens einer Zeile

**units summary** Maßeinheiten für die Zeilen oder Spalten (als Spaltenvektor mit  $m$  Elementen bzw. als Zeilenvektor mit  $n$  Elementen anzugeben).

Zudem kann man eine Anmerkung und eine Unterschrift angeben.

Hier ein weiteres Beispiel:

$Body := \begin{bmatrix} 11 & 12 \\ 21 & 22 \end{bmatrix} km$   
 $Header := ["Header"]$   
 $Footer := ["Footer"]$   
 $LS := ["LS"]$   $US := \begin{bmatrix} km \\ m \end{bmatrix}$   $RS := ["RS"]$

|          |        |       |        |    |
|----------|--------|-------|--------|----|
|          | Header |       |        |    |
| LS       | 11     | 12    | 1000 m | RS |
|          | 21000  | 22000 | 1 m    |    |
|          | Footer |       |        |    |
| Table 12 |        |       |        |    |

Body

**Table settings**

Data | **Data appearance** | Main layout | Resizing

Header  show header

LS  show left stub

RS  show right stub

Footer  show footer

US  show units summary

show caption:  show footnote:

Table %T NOTE

**Table settings > Data appearance:** Einstellung des Zahlenformats, der Schriftart und der Farben. Die Einstellungen beziehen sich auf den jeweils in der Liste ausgewählten Tabellenteil:

**Table settings**

Data | **Data appearance** | Main layout | Resizing

Body

Font

Alignment

Width

Height

Number format

Decimal places

Significant figures

Background

Foreground

Styles

Padding

Border size

Border color

Border style

set as default

**Symbolische Ausdrücke:** Im Tabellenbereich können auch symbolische Ausdrücke angezeigt werden, man gibt diese als Zeichenketten im internen Textformat an, die mit **S:** beginnen, also z.B. "**S:a^b**" für  $a^b$ . Hier ein Beispiel, das die Leistungsfähigkeit des Tabellenbereichs demonstriert, dabei sind auch Schriftarten verstellt worden.

```
misc := [
  1      n      i      sin(|x|)
  0      {1     0      3 [ A B_A C^2 ]
          e
  ∞      0      1      1 m
  0      "x^0 := 3431711" 0      x m
  0      "S:x^n+x^{n-1}+...+x" 0      "S:B.A+2" ]
```

|   |   |      |                    |   |                |
|---|---|------|--------------------|---|----------------|
| 1 | 3.14  | 0+1i | sin( x )           |   |                |
| 0 | 1   | 0    | 1                  | 2 |                |
|   | 2.72  |      | 3                  | A | B <sub>A</sub> |
| ∞ | 0   | 1    | 1 m                |   |                |
| 0 | x <sup>0</sup> := 3431711                   | 0    | x m                |   |                |
| 0 | x <sup>n</sup> + x <sup>n-1</sup> + ... + x | 0    | B <sub>A</sub> + 2 |   |                |

Table 10  
misc

**Große Tabellen:** Bei großen Tabellen kann man die Größe des dargestellten Bereichs beschränken. Durch Auslassungszeichen (Ellipsen) wird angedeutet, dass weiterer Inhalt vorhanden ist. Durch Klicken auf die Auslassungszeichen kann man durch die Daten navigieren.

```
n := 20
M := Random(n; n)      ZT := [1..n]      ST := ZTT
```

|     |     |       |       |       |       |       |     |
|-----|-----|-------|-------|-------|-------|-------|-----|
|     | ... | 4     | 5     | 6     | 7     | 8     | ... |
| ... | ... | ...   |       |       |       |       | ... |
| 4   |     | 0,099 | 0,725 | 0,973 | 0,787 | 0,303 |     |
| 5   |     | 0,898 | 0,812 | 0,815 | 0,424 | 0,252 |     |
| 6   | ... | 0,423 | 0,601 | 0,585 | 0,977 | 0,626 | ... |
| 7   |     | 0,821 | 0,194 | 0,713 | 0,248 | 0,752 |     |
| 8   |     | 0,886 | 0,977 | 0,220 | 0,296 | 0,746 |     |
| ... | ... | ...   |       |       |       |       | ... |

Table 11

M

Im **Kontextmenü** (rechte Maustaste) des Tabellenbereichs sind folgende Einträge interessant:

**Display input data** Platzhalter anzeigen,

**Expand to table limits** Tabelle so vergrößern, dass alle Daten angezeigt werden,

**Fit to page size** Tabelle an Seitengröße anpassen (in beide Richtungen so groß wie möglich),

**Shrink to page size** Tabelle auf Seitengröße reduzieren

**Export> Save as** Speichern als Grafikdatei oder HTML,

**Export> Copy to clipboard** Kopieren in die Zwischenablage, von dort als Tabelle oder Text in Word oder OpenOffice einfügbar.

## 5.10. Hyperlink-Bereich

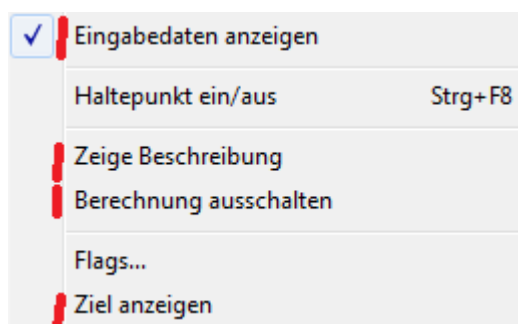
Mit diesem Bereich kann man aktive Hyperlinks auf andere Dokumente anlegen. Gegenüber Hyperlinks in Textbereichen (Standard oder erweitert) bietet der Bereich erweiterte Funktionen:

- das Sprungziel wird in den Platzhalter geschrieben und kann eine Zeichenkette oder zeichenkettenwertige Variable sein. Damit kann das Sprungziel dynamisch berechnet werden.
- Es wird ein zielabhängiges Symbol angezeigt.









Der Hyperlink-Bereich wird vom Plug-in *Hyperlink Region* bereitgestellt. Eingefügt wird er mit **Einfügen> Hyperlink** oder mit  $\boxed{\text{Strg}}-\boxed{\text{H}}$ .



In den Platzhalter kann man das Verweisziel als Zeichenkette oder Variablenname (mit passender Zeichenkette als Inhalt) einfügen. Das Erscheinungsbild kann im Kontextmenü mit vier Optionen beeinflusst werden:

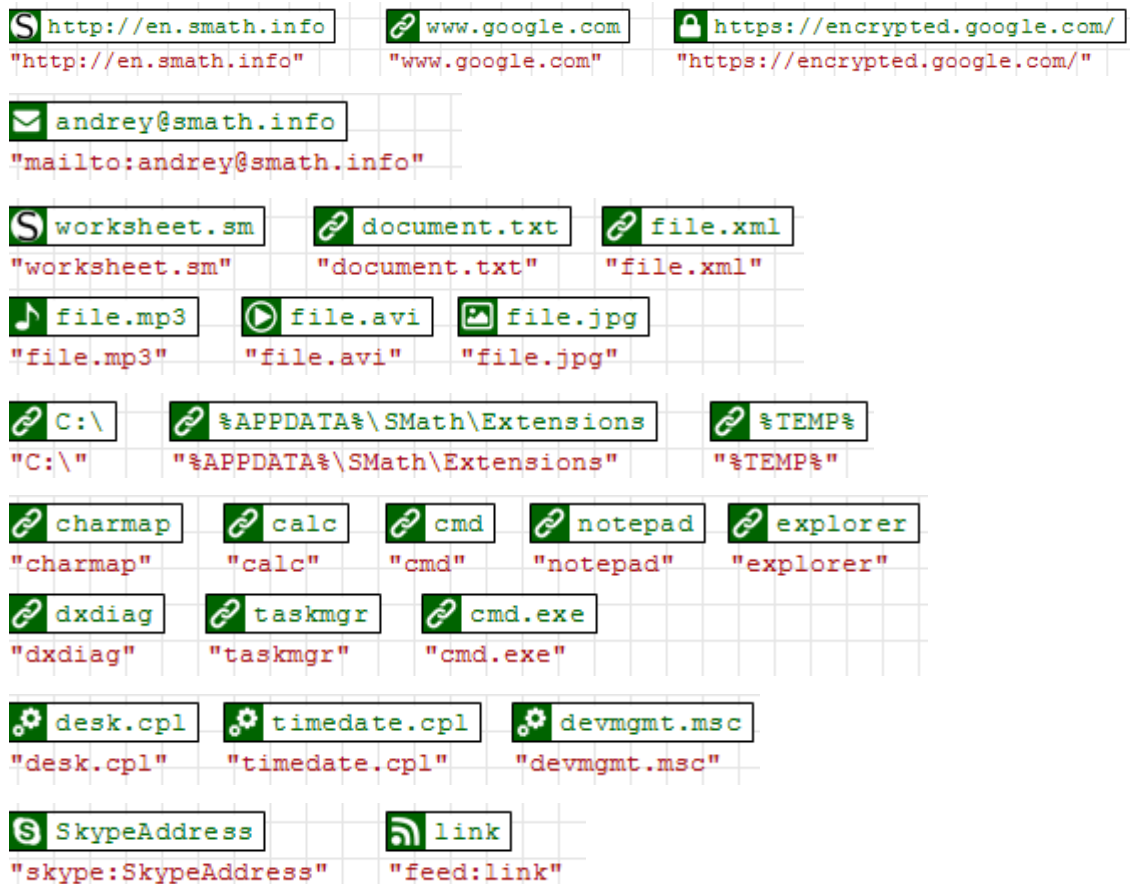


Im folgenden Beispiel sind die Standardeinstellungen hervorgehoben:

|   |                                 |   |                               |
|---|---------------------------------|---|-------------------------------|
|  | <u>Eingabedaten anzeigen AN</u> |  | Eingabedaten anzeigen AUS     |
| "link"  |                                 |   |                               |
|  | Zeige Beschreibung AN           |  | <u>Zeige Beschreibung AUS</u> |
| "link"  |                                 | "link"  |                               |
|  | <u>aktiviert</u>                |  | deaktiviert                   |
| "link"  |                                 | "link"  |                               |
|  | Ziel anzeigen AN                |  | <u>Ziel anzeigen AUS</u>      |
| "link"  |                                 | "link"  |                               |

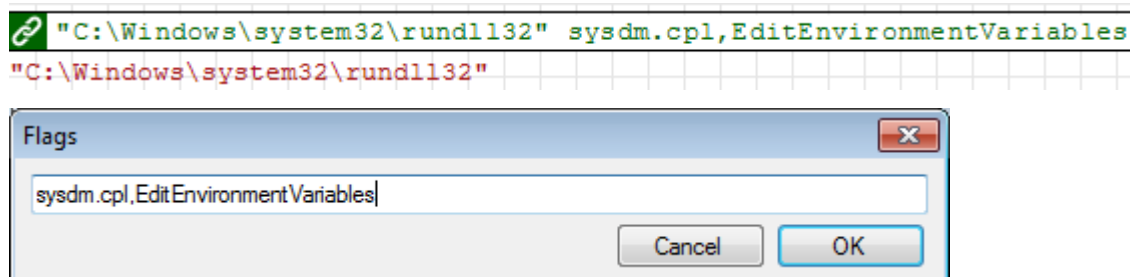
Das Ziel wird als SMath-Zeichenkette angegeben. Aus der Zwischenablage kann aber auch einfacher Text eingefügt werden.

Hyperlink-Bereiche können auf die verschiedensten Ziele verweisen, z.B. Dateien, Verzeichnisse, Internet-Seiten oder Programme. Je nach Typ passt sich das Symbol im Hyperlink-Bereich an. Hier einige Beispiele:



Dateien und Programme werden mit dem Windows-Befehl *run* gestartet. Relative Pfadangaben beziehen sich auf das aktuelle Verzeichnis, das nicht unbedingt das Verzeichnis ist, in dem sich das aktuelle Dokument befindet.

Mit dem Kontextmenü-Eintrag „Flags“ kann man Kommandozeilenparameter an das aufgerufene Programm übergeben, z.B.

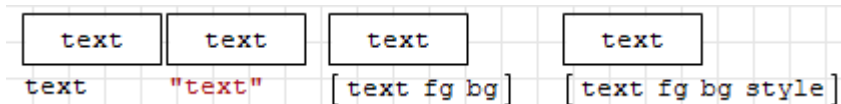


## 5.11. Formatierte Textausgabe (Formatted Label)

Dieser Bereich wird vom Plug-in *Conditionally Formatted Label* bereitgestellt. Das ist eine Anzeige, bei der Text, Textfarbe und Hintergrundfarbe durch den Inhalt der Variable im Platzhalter gesteuert werden können. Eingefügt wird er über **Einfügen** > **Formatted label**.



In den Platzhalter kommt eine Zeilenmatrix (Vektoren oder Listen gehen auch) mit dem Text, der Vordergrundfarbe, Hintergrundfarbe und dem Erscheinungsbild:

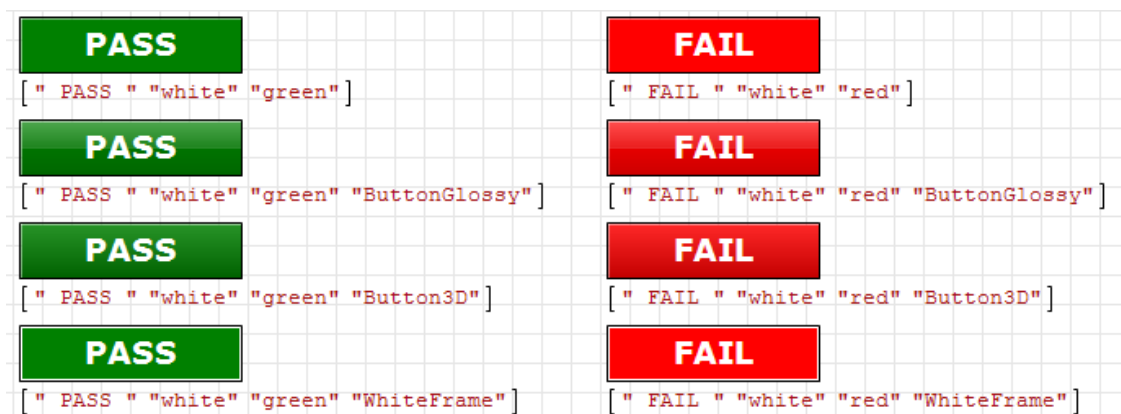


**text** (Zeichenkette oder undefinierter Bezeichner) darzustellender Text,

**fg** (Zeichenkette) Vordergrundfarbe, als Name, rgb oder argb,

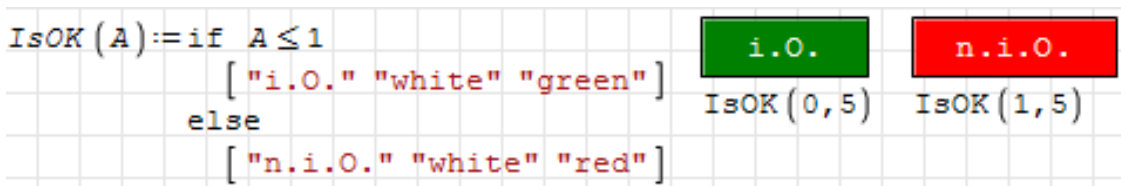
**bg** (Zeichenkette) Hintergrundfarbe,

**style** (Zeichenkette) Darstellungsart, zulässige Werte: "ButtonGlossy", "Button3D", "WhiteFrame".



Doppelklick auf den Bereich öffnet einen Dialog zur Auswahl der Schriftart und -größe.

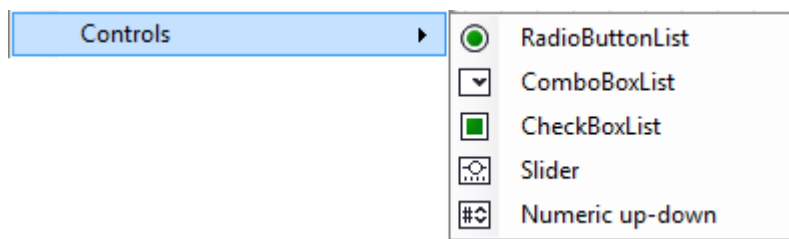
Im folgenden Beispiel werden die Hintergrundfarbe und der Text abhängig von einem Zahlenwert gesetzt.



## 5.12. Steuerelemente (Controls)

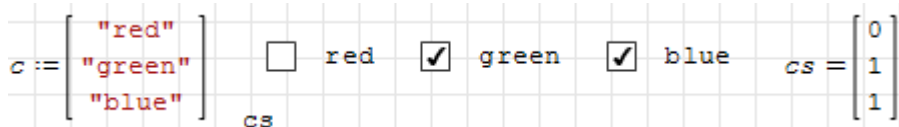
Es gibt besondere Bereiche für die interaktive Veränderung von Variablenwerten. Das geht natürlich zunächst immer mit einer Zuweisung in einem Formelbereich. In manchen Situation möchte man es allerdings bequemer haben oder einen bestimmten Wertebereich oder einzelne Werte zur Auswahl anbieten.

SMATH Studio verfügt dafür über Steuerelemente (interaktive Bereiche), die durch verschiedene Plug-ins bereitgestellt werden.



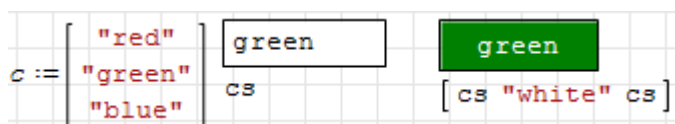
Alle Steuerelemente haben diverse Einstellmöglichkeiten, die über das Kontextmenü („Settings“) oder durch Doppelklick auf die aktive Auswahl erreicht werden können. Bei einigen können auch Einstellungen über Variablenwerte im Rechenblatt dynamisch geändert werden.

**CheckBoxList** Tastenfeld zur Auswahl einer beliebigen Anzahl und Kombination gegebener Möglichkeiten. Diese werden durch einen Vektor oder eine Matrix vorgegeben (Name im Einstellungsfenster wählen). Man kann wählen, ob die Indizes, die Datenzeilen oder die Werte einer wählbaren Spalte zurückgegeben werden sollen. Als Beschriftung wird die erste Spalte benutzt.

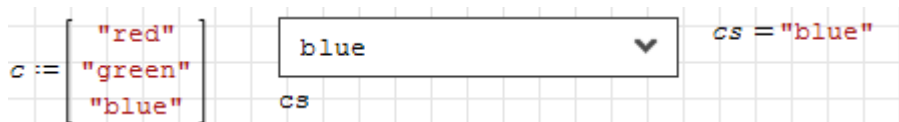


In den Einstellungen sind weitere Formatierungsmöglichkeiten verfügbar.

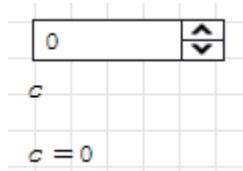
**Combobox** Auswahlbox, aus einem gegebenen Vektor kann ein Element ausgewählt werden. Im Rechenblatt bewirkt ein Klick auf den Bereich das Weiterschalten der Auswahl. Im Viewer wird die Auswahl als Drop-Down-Menü dargestellt. Im Beispiel hier wird `cs` aus dem Vektor `c` ausgewählt und beeinflusst den Text und die Farbe einer bedingt formatierten Anzeige.



**ComboBoxList** Auswahlbox, bei der eine von mehreren gegebenen Möglichkeiten gewählt werden kann. Die Möglichkeiten werden als Liste oder Matrix vorgegeben. Bei einer Matrix wird die erste Spalte für die Beschriftung verwendet. Die Ergebnisvariable im Platzhalter kann über ein Einstellungsfenster auf den Index, die Datenzeile oder einen bestimmten Spalteneintrag der Auswahl konfiguriert werden.



**Numeric-up-down** Zahleneingabefeld mit Schaltflächen zum schrittweisen Erhöhen oder Vermindern des Zahlenwerts oder dessen Änderung mit dem Mausrad. Man kann dabei Bereichsgrenzen und Schrittweite angeben.



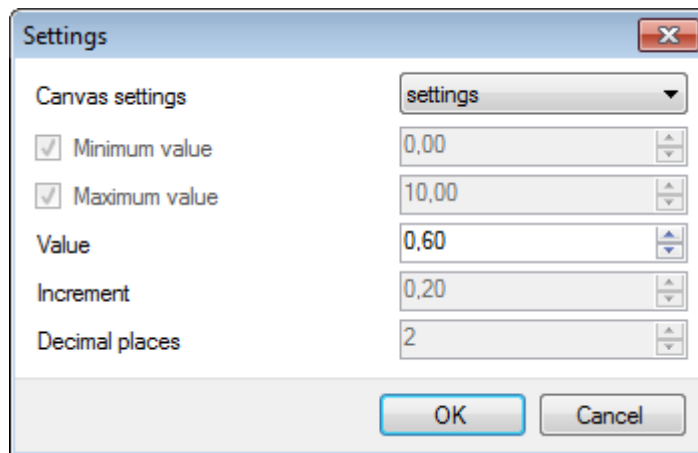


Die Einstellungen sind sowohl über das Kontextmenü als auch dynamisch über Variablen im Rechenblatt steuerbar.

```

settings := {
  "min" = 0
  "max" = 10
  "increment" = 0,2
  "decimal places" = 2
}
res
res = 0,6

```



**RadioButtonList** Bereich, bei dem durch Drücken eines Knopfes genau eine der gegebenen Möglichkeiten ausgewählt wird.

- Eingabe: Optionen als Liste oder Matrix
- Auswahl: Mausklick, Mausrad oder +/- -Tasten.
- Ergebnis: Auswahlindex, Eintrag in einer wählbaren Spalte oder ganze Datenzeile,
- Einstellungen: Doppelklick auf den Bereich oder Kontextmenü > Settings.

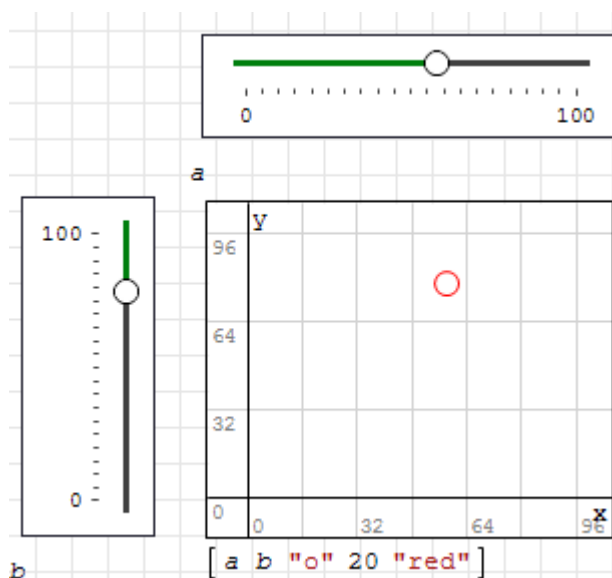
```

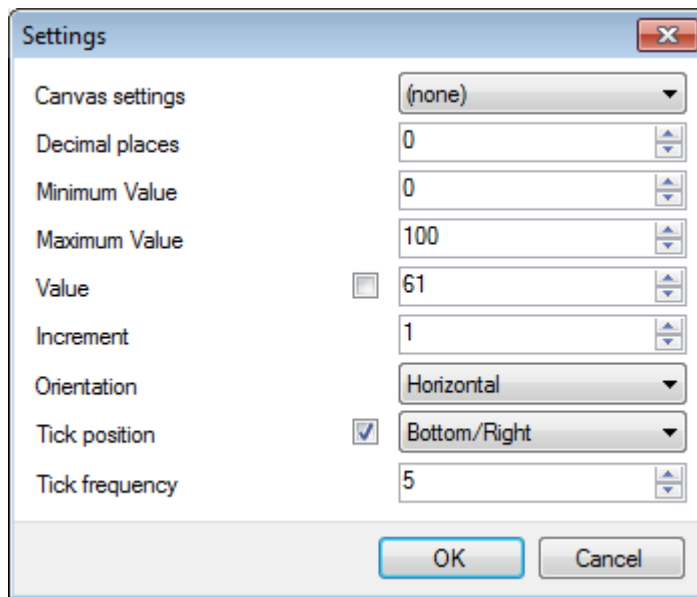
c := [ "red"
      "green"
      "blue" ]
cs

```

red   
 green   
 blue   
blue
  
[ cs "white" cs ]

**Slider** Schieberegler zur interaktiven Einstellung von Zahlenwerten



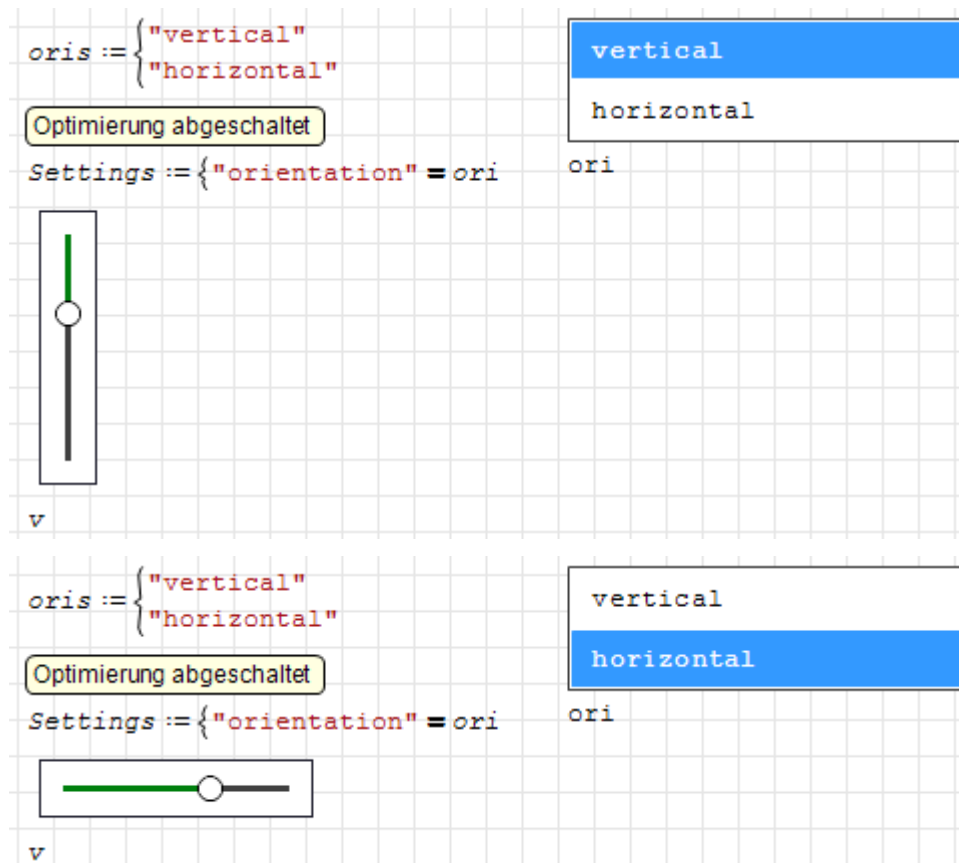


Die Einstellungen können dynamisch (durch Variablen im Rechenblatt) geändert werden, indem eine Liste mit Schlüssel=Wert-Paaren definiert wird. Der Name dieser Liste ist in den Einstellungen unter „Canvas settings“ einzutragen. Schlüssel und Wert sind als Zeichenketten oder Zahlen anzugeben. Gültige Schlüssel sind in der folgenden Tabelle angegeben.

| Schlüssel  | Wert  | Bedeutung   |
|--|---|---|
| "increment"  | Zahl $> 10^{-15}$                                     | Schrittweite  |
| "min", "minimum",<br>"min value",<br>"minimum value" | Zahl  | Minimaler Wert  |
| "max", "maximum",<br>"max value",<br>"maximum value" | Zahl  | Maximaler Wert  |
| "value"  | Zahl  | Darzustellender Wert<br>(nicht interaktiv<br>verstellbar) |
| "initial value"                                      | Zahl  | Anfangswert   |
| "orientation"  | "horizontal" oder<br>"vertical"                       | Ausrichtung   |
| "tick frequency"                                     | Zahl $> 0$  | Skalenteilung   |
| "tick position"                                      | "top/left",<br>"bottom/right",<br>"both", oder "none" | Skalenposition  |

Tabelle 5.2.: Steuerparameter des Schiebereglers (Slider). Leerzeichen sowie Groß-/Kleinschreibung sind in den Zeichenketten nicht signifikant.

Beispiel: Hier wird die Ausrichtung des Schiebereglers durch die Auswahl in der Combobox gesteuert.

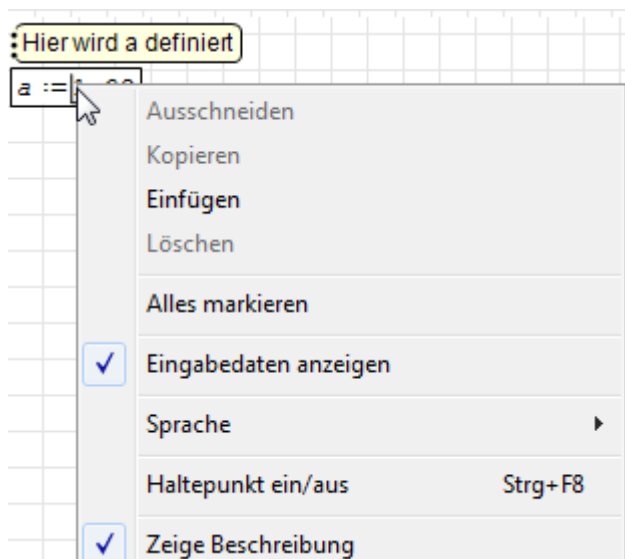


## 5.13. Beschreibungstexte

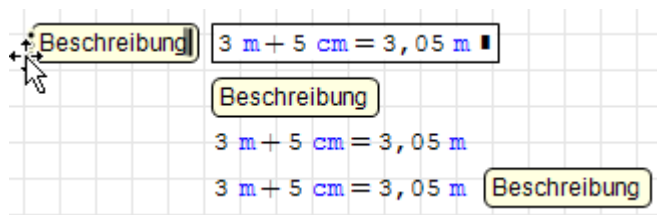
SMath besitzt einen Mechanismus, mit dem Beschreibungstexte für Bereiche angegeben werden können.

Für die bloße Kommentierung eines Rechenblattes ist dies unflexibler und umständlicher als die Benutzung von Textbereichen.

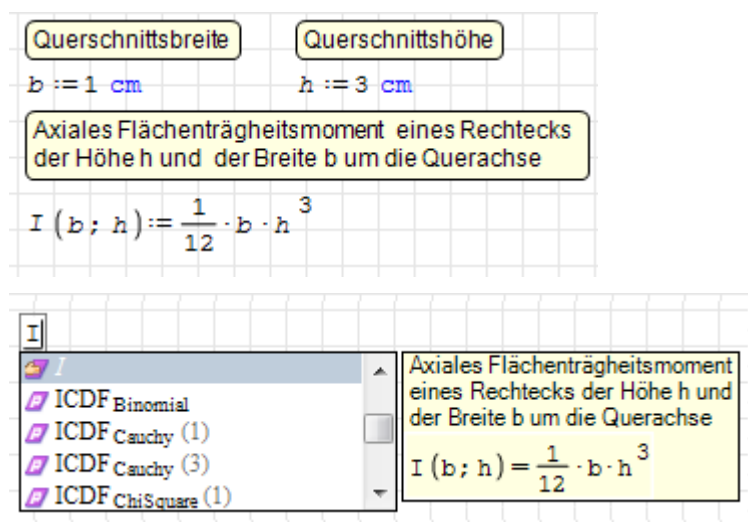
Sehr interessant ist allerdings die Verwendung von Beschreibungstexten bei Definitionen. Diese Texte werden in der dynamischen Hilfe angezeigt.



Beschreibungstexte können mit der Maus an verschiedene Seiten des jeweiligen Bereichs gezogen werden. Fassen Sie beim Ziehen am linken gepunkteten Rand des gelben Feldes an.



Werden in einem Formelbereich Variablen oder Funktionen definiert, so werden die Beschreibungstexte von der dynamischen Hilfe benutzt (spätestens nach einmaliger Neuberechnung des Rechenblatts):



Beschreibungstexte an Definitionen können auch mit der Funktion `description()` abgefragt werden. Bei Funktionen muss die korrekte Anzahl von Argumenten angegeben werden, da sich die Definitionen je nach deren Anzahl unterscheiden können.

```
description(b) = "Querschnittsbreite"
description(h) = "Querschnittshöhe"
description(I(■; ■)) = "Axiales Flächenträgheitsmoment eines"
```

Beschreibungstexte werden im SMath Viewer (Kapitel 11) für Formelbereiche als Beschriftungstext benutzt, der den Formelinhalt ersetzt.

## 5.14. Trennlinien und Blattbereiche

Trennlinien und Blattbereiche dienen der Strukturierung des Rechenblatts. Beim Export in den SMath Viewer haben sie besondere Formatierungswirkung (Kapitel 11).

**Trennlinien** werden eingefügt mit **Einfügen** > **Trennlinie** oder mit **Strg**-**↵**.

Im Rechenblatt erzeugen sie einen Seitenumbruch, der nur in der Druckansicht oder in der Einzelseitenansicht von Bedeutung ist. Im Viewer erzeugen sie eine neue Registerkarte.

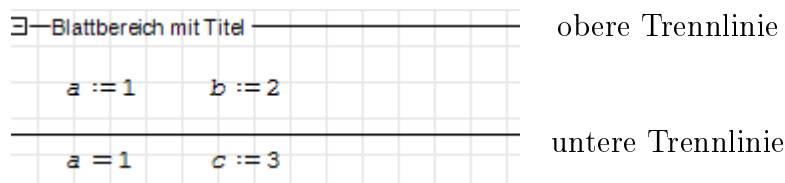
Trennlinien können mit einem Titel versehen werden, der im Viewer als Titel für die Registerkarte benutzt wird.



Trennlinien können wie alle anderen Bereiche auch nach Anklicken oder Markieren mit der Maus oder den Pfeiltasten verschoben werden.

**Blattbereiche** gruppieren den Inhalt zwischen einer oberen und einer unteren Trennlinie. Die obere Trennlinie hat eine Schaltbox zum Ausblenden des Blattbereichs und kann einen Titeltext erhalten. Die untere Trennlinie kann nicht beschriftet werden und erzeugt im Unterschied zur normalen Trennlinie keinen Seitenumbruch.

Ein Blattbereich wird eingefügt mit **Einfügen** > **Blattbereich**.

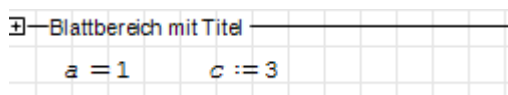


Die Trennlinien können dann nach Bedarf verschoben werden, um den gewünschten Bereich einzugrenzen.

Blattbereiche können auch verschachtelt werden. Unabhängig davon, in welcher Reihenfolge sie erzeugt wurden, bilden immer die nächstgelegenen oberen und unteren Trennlinien eine Bereichsebene.

Beim Löschen einer oberen oder unteren Trennlinie wird immer auch die andere zugehörige Trennlinie entfernt, so dass die Struktur stets eindeutig bleibt.

Ein Blattbereich kann durch Anklicken der Box  ausgeblendet werden. Dann sieht man nur noch die obere Trennlinie mit dem Titel und eine Box , mit der man den Bereich wieder einblenden kann.



Im Viewer (siehe Kapitel 11) werden ausgeblendete Blattbereiche nicht angezeigt, eingblendete Bereiche werden mit einer Box umrahmt.

## 5.15. Mehrsprachige Arbeitsblätter

SMath Studio ist von vornherein auf die Unterstützung verschiedener Sprachen angelegt. Das betrifft nicht nur die Benutzeroberfläche, sondern auch die Arbeitsblätter selbst. Dies wurde umfangreich im interaktiven Handbuch (siehe Abschnitt 4.3) benutzt, welches durchgehend zweisprachig (deutsch und englisch) vorliegt.

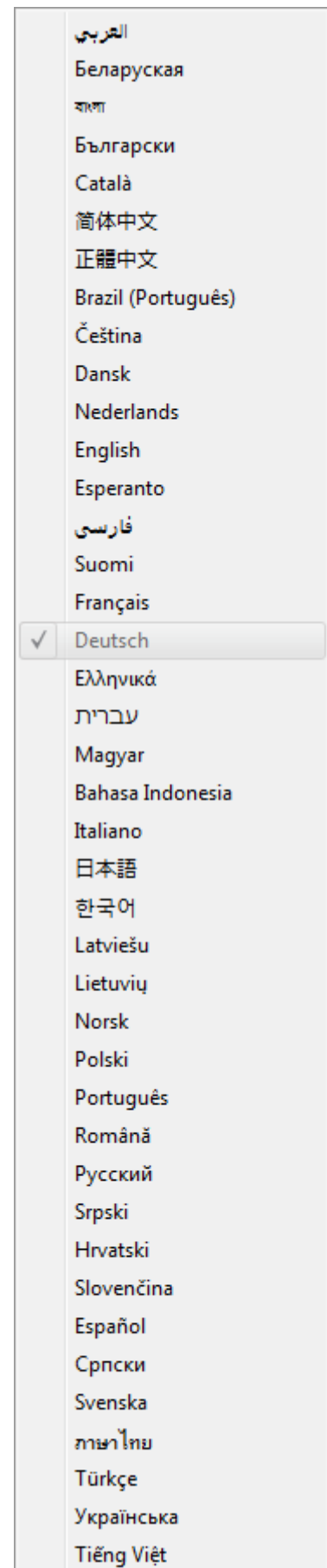
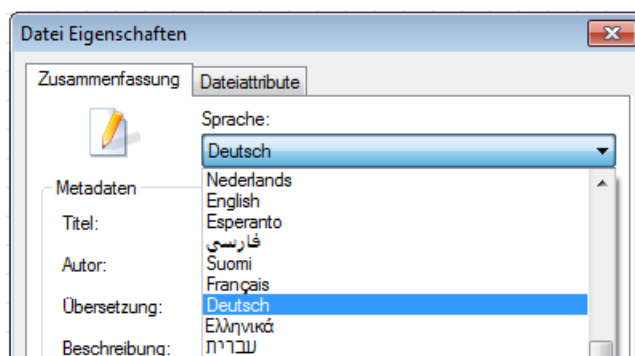
Mehrsprachige Elemente werden in der Sprache angezeigt, die für die Benutzeroberfläche eingestellt ist (**Extras** > **Einstellungen** > **Sprache**)

Alternative Sprachvarianten können angegeben werden für

- Dateieigenschaften,
- Text in Standard- und erweiterten Textbereichen (Writer),
- Beschreibungstexte an Formel- und Diagrammbereichen,
- Trennlinien und Überschriften von Blattbereichen.

Die Spracheinstellung ist bei Textbereichen, Beschreibungstexten, Trennlinien und Blattbereichsüberschriften über **Kontextmenü** > **Sprache** erreichbar. In nebenstehendem Bild sind die bereits vorhandenen Sprachversionen (Deutsch, Englisch, Französisch) angehakt und die aktuelle Einstellung (Deutsch) ist ausgegraut.

Die Angaben zu den Dateieigenschaften (Titel, Autor, Beschreibung, Firma, Stichworte) können in mehreren Sprachen angegeben werden. Der Dialog wird aufgerufen mit **Datei** > **Eigenschaften**. Dort kann man unter „Sprache“ auswählen, in welcher Sprache man die Angaben machen will. Theoretisch könnte man dies für alle verfügbaren Sprachen parallel tun.




## 5.16. Regeln für lesbare Rechenblätter

SMath Studio dient der Durchführung und Dokumentation von technischen Berechnungen. Es ersetzt sozusagen die Handrechnung mit Papier und Bleistift und Taschenrechner. Damit auch Außenstehende die Berechnung einfacher nachvollziehen können, sind folgende Regeln hilfreich:

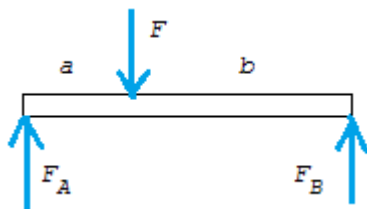
- Erläutern Sie mit Text und Skizzen oder Bildern die **Aufgabenstellung/das Problem**, so dass man nicht in anderen Dokumenten nachschauen muss. Ein Scan oder Screenshot kann da schon hilfreich sein.
- Verwenden Sie für Größen, die sich ändern können, stets **Variablen**, nicht die Zahlenwerte. Das gilt ganz besonders für gegebene Größen.
- Verwenden Sie Formelzeichen, die in Ihrem Fachgebiet üblich sind und die Sie auch in der Handrechnung verwenden würden. Machen Sie dabei Gebrauch von **Textindizes**.

|           |       |                 |
|-----------|-------|-----------------|
| Gut:      | $F_x$ | $\sigma_{\max}$ |
| Schlecht: | Fx    | sigma_max       |

- Geben Sie zu jedem vorkommenden **Formelzeichen** eine Erklärung. Wenn Sie Formelzeichen oder Formeln im Text verwenden wollen, fügen Sie einen Formelbereich mit **Alt+** oder **Einfügen > Formel** ein.

Dies ist ein Textbereich mit drei Formelbereichen und automatischem Zeilenumbruch. Die Diagonale eines Rechtecks mit den Kantenlängen  $a$  und  $b$  hat die Länge  $c := \sqrt{a^2 + b^2}$


- Machen Sie reichlich Gebrauch von **Skizzen**. Für das schnelle Erstellen von groben Skizzen ist der Image-Bereich (Abschnitt 5.8) mit MS Paint geeignet. Beschriftungen erzeugen Sie durch überlagerte Formelbereiche. So stellen Sie sicher, dass die Beschriftung lesbar und passend zu den Berechnungsformeln ist. Jedes Koordinatensystem, jede Kraft oder Abmessung, die Sie in der Rechnung verwenden oder auf die Sie sich beziehen, sollten in einer Skizze gekennzeichnet sein.



Freunde der Vektorgrafik können den Image-Bereich auch auf SVG-Format umstellen und Skizzen z.B. mit Inkscape erzeugen. Fotos und andere Bilder können Sie aus der Zwischenablage ins Rechenblatt kopieren und skalieren.

- Wenn Sie externe Daten verwenden (Bilder, Tabellen), dann geben Sie deren Namen im Rechenblatt als Kommentar mit an, damit Sie diese später wiederfinden, falls Sie etwas ändern wollen. Daten kann man auch aus Excel- oder Textdateien importieren (Abschnitt 6.14).

- Verwenden Sie die eingebauten Mechanismen für die **Maßeinheitenverwaltung**. Geben Sie bei Eingangsdaten die Maßeinheiten so an, wie sie in der Quelle vorkommen (Handbuch, Aufgabenstellung). Geben Sie bei Ergebnissen die üblichen Maßeinheiten an.

|            |                                  |   |
|------------|----------------------------------|---|
| Kraft      | $F := 10 \text{ N}$              |   |
| Hebelarm   | $h := 20 \text{ mm}$             |   |
| Drehmoment | $M := F \cdot h = 0,2 \text{ J}$ | Unübliche Einheit für Drehmoment  |
|            | $M = 20 \text{ N cm}$            |  M= <input type="text" value="-"/> N <input type="text" value="Tab"/> *cm <input type="text" value="Tab"/> |

- Geben Sie Ergebnisse mit sinnvollen Maßeinheiten an, so dass die Zahlenwerte möglichst zwischen 1 und 1000 liegen. Sorgen Sie per Voreinstellung dafür, dass immer **drei signifikante Stellen** angegeben werden (siehe Abschnitt 3.4). Weichen Sie davon nur ab, wenn es für den Leser wichtig ist.
- Seien Sie mit **Formatierungen** genauso sparsam wie bei Bleistiftrechnungen auf dem Papier. Zum Hervorheben von Textbereichen, z.B. für Zwischenüberschriften, sollten **Fettschrift** und Unterstreichung reichen. Ergebnisse können Sie einrahmen. Manchmal ist es sinnvoll, gegebene Daten hervorzuheben, z.B. hellgelb zu hinterlegen. Das ist besonders dann sinnvoll, wenn Sie ein Rechenblatt erstellen, das andere dann mit ihren eigenen Daten benutzen sollen. Wenn Sie Farben einführen, dann stellen Sie sicher, dass der Leser verstehen kann, welche Farbe welche Bedeutung hat.
- **Inaktive Formeln:** Wenn Sie Berechnungen manuell anstellen müssen (z.B. symbolische Umformungen, die Sie mit SMath nicht hinbekommen), dann benutzen Sie Formelbereiche einfach als Formeleditor ohne Rechenfunktion. Dafür gibt es drei Möglichkeiten:
  - Benutzen Sie für Gleichungen das (fette) boolesche Gleichheitszeichen aus der Palette „Boolesch“ der Seitenleiste. Damit wird die Formel ein logischer Ausdruck, der nicht ausgewertet wird (da weder eine Zuweisung erfolgt, noch ein Ergebnis angezeigt wird).
  - Deaktivieren Sie die Formel: **Kontextmenü** > **Berechnung ausschalten**. Die Formel wird dann ausgegraut angezeigt. Daher ist diese Variante nicht gut lesbar.
  - Schreiben Sie die Terme der Gleichung in einen Textbereich. Das Gleichheitszeichen steht zwischen zwei Formeln:

|                                      |                  |
|--------------------------------------|------------------|
| Logischer Ausdruck                   | $M = F \cdot h$  |
| Deaktivierte Gleichung               | $M := F \cdot h$ |
| Textbereich mit zwei Formelbereichen | $M = F \cdot h$  |

- Nebenrechnungen, die für das Verständnis nicht wesentlich sind, können in ausgeblendeten Blattbereichen versteckt werden (Abschnitt 5.14).
- Verschenden Sie keinen Platz, nutzen Sie die Möglichkeit, Inhalte auch nebeneinander darzustellen. Das verbessert die Lesbarkeit am Bildschirm.



## Checkliste für lesbare Rechenblätter

- Aufgabenstellung enthalten?
- Ausreichend Variablen mit passenden Namen und Maßeinheiten eingeführt und erläutert?
- Alle nötigen Skizzen vorhanden und sauber beschriftet?
- Rechenverfahren benannt und nachvollziehbar?
- Ergebnisse korrekt dargestellt (signifikante Stellen, Maßeinheiten)?
- Platzsparend und sparsam formatiert?


Rechenblätter können eigenständige Dokumente sein oder ganz oder in Ausschnitten in andere Dokumente einbinden (z.B. Abschlussarbeiten, Berichte).

Welche der beiden Varianten gewählt werden soll, hängt von den Umständen ab:

Ein eigenständiges SMath-Dokument ist sinnvoll,

- wenn Sie alternativ eine Handrechnung mit Papier und Bleistift anfertigen und abheften würden,
- wenn Sie auf Komfortfunktionen zur Dokumentstrukturierung verzichten können (Abschnitte, Formatvorlagen, Nummerierungen...),
- wenn keine Publikationsqualität erforderlich ist (Sie also hinsichtlich der Formatierung frei sind).
- als Anhang zu technischen Berichten.

## 5.17. Beispiel

Dieses Beispiel ist für Einsteiger gedacht, für die Hinweise zur Tastatureingabe nützlich sein können. In ähnlicher Form (mit Zufallszahlen) ist es im Interaktiven Handbuch zu finden unter  > **Funktionsindex** > **linterp()**.

Interpolation ist die Berechnung von Funktionswerten in den Zwischenräumen zwischen gegebenen Datenpunkten ( $x$ ,  $y$ -Paaren). Wenn man Punkte sucht, die außerhalb des Bereichs der gegebenen Werte liegen, dann spricht man von Extrapolation, die aber häufig von Interpolationsverfahren gleich mit erledigt wird.

Im folgenden Rechenblatt können Sie die Eigenschaften der Interpolation nachvollziehen. Die Daten sind als Vektoren für die  $x$ -Werte und  $y$ -Werte gegeben. Die Werte werden mit der Funktion `linterp()` linear interpoliert und die Funktion gemeinsam mit den Datenpunkten dargestellt.

Interpolation

$$X := \begin{bmatrix} 1 \\ 3 \\ 4 \\ 7 \\ 10 \end{bmatrix} \quad Y := \begin{bmatrix} 1 \\ 2 \\ 2 \\ 5 \\ 5 \end{bmatrix}$$

$$f(x) := \text{linterp}(X; Y; x)$$

$$P := \{f(x)\}$$

```
for i ∈ [1..rows(X)]
  Pi+1 := [Xi Yi "o" 10 "red"]
```



P =

$$P = \left( \begin{array}{c} \text{linterp} \left( \begin{bmatrix} 1 \\ 3 \\ 4 \\ 7 \\ 10 \end{bmatrix}; \begin{bmatrix} 1 \\ 2 \\ 2 \\ 5 \\ 5 \end{bmatrix}; x \right) \\ [1 \ 1 \ "o" \ 10 \ "red"] \\ [3 \ 2 \ "o" \ 10 \ "red"] \\ [4 \ 2 \ "o" \ 10 \ "red"] \\ [7 \ 5 \ "o" \ 10 \ "red"] \\ [10 \ 5 \ "o" \ 10 \ "red"] \end{array} \right)$$

Die Anzeige von P muss umgestellt werden mit:

**Rechte Maustaste>  
Optimierung>  
symbolisch**

Sonst beschwert sich SMATH, dass x nicht definiert ist.

Hinweise zur Tastatureingabe

Matrix definieren: X: **Strg+M**

Element einfügen: ;

Element löschen: X

f(x **Leer** :linterp **Tab**

Liste definieren: P:sys **Tab**

Element löschen: X

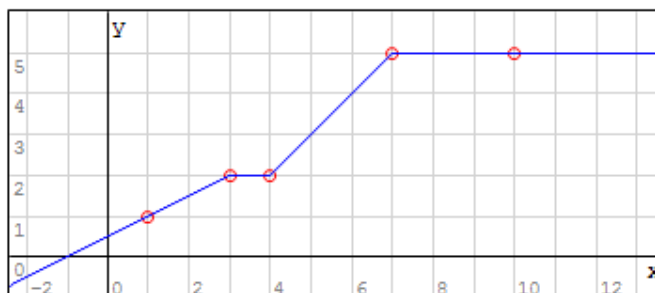
for **Tab** i rang **Tab** 1 rows **Tab** X

P[i+1 **Leer** : **Strg+M** (1 Zeile, 5 Spalten)

X[i Y[i "o



@



P

**Ausschnitt verändern:**

- \* Achsen verschieben: in Diagramm klicken und Ziehen
- \* Zoom x und y: Mausrad
- \* Zoom x: Shift-Mausrad
- \* Zoom y: Strg-Mausrad

Hinweis: SMATH stellt neben der linearen Interpolation zwei weitere Funktionen zur Verfügung:

- `cinterp()` kubische Spline-Interpolation
- `ainterp()` kubische Spline-Interpolation nach Akima [Erläuterung des Verfahrens \(TU Wien\)<sup>2</sup>](#). Das Verfahren berechnet vor der Interpolation an den Stützstellen die Neigung aus einer 5-Punkte-Umgebung. In unserem Beispiel funktioniert das nur für den mittleren Punkt.

<sup>2</sup>[https://www.ads.tuwien.ac.at/docs/lva/mmgdv/k1\\_\\_\\_011.htm](https://www.ads.tuwien.ac.at/docs/lva/mmgdv/k1___011.htm)

# 6. Mathematik

In SMath Studio können Sie Berechnungen dokumentieren und durchführen, indem Sie auf einem Rechenblatt Formelbereiche anordnen. Diese sind gleichzeitig eine Berechnungsvorschrift und deren lesbare Dokumentation.

## 6.1. Formelbereich

Formelbereiche entstehen auf dem Rechenblatt an der Cursorposition  $+$ , indem Sie einfach los tippen. Sie entstehen auch, wenn Ziffern, Formelzeichen oder Funktionen aus Menüs oder der Seitenleiste eingefügt werden. Formelbereiche kann man über **Einfügen > Formel** auch in Textbereiche einfügen. Der Inhalt von Formelbereichen wird mathematisch formatiert und ausgewertet. Die Mathematik wird weiter unten erläutert, hier folgen erstmal Hinweise zum Bedienkonzept des Formelbereichs.

### 6.1.1. Einfügemarke in Formelbereichen

Die Markierung  $\underline{\quad}|\quad$  in Formelbereichen besteht aus zwei Teilen:

- Unterstreichung:  $\underline{\quad}$  Sie kennzeichnet den **Wirkungsbereich** (*scope*) der Markierung
- Senkrechte Einfügemarke:  $|\quad$  Sie kennzeichnet den **Ort** der nächsten Eingabe

Die Wirkung mancher Aktionen hängt davon ab, welche Ausdrücke in der Formel markiert sind. Die Markierung wird mit der Maus vor oder nach einem Bezeichner gesetzt und kann mit der Leertaste erweitert werden.

|            |                    |                    |                    |                    |
|------------|--------------------|--------------------|--------------------|--------------------|
| Markierung | $\underline{a+b} $ | $\underline{a+b} $ | $\underline{a+b} $ | $\underline{a+b} $ |
| Eingabe /  | $a+\frac{b}{ }$    | $\frac{a+b}{ }$    | $a+\frac{ }{b}$    | $\frac{ }{a+b}$    |
| Eingabe *  | $a+b\cdot $        | $(a+b)\cdot $      | $a+  \cdot b$      | $  \cdot (a+b)$    |

Abbildung 6.1.: Wirkung von Eingaben in Formeln abhängig von der Markierung

### 6.1.2. Griechische Buchstaben

Griechische Buchstaben können in Formel- und Textbereichen eingegeben werden durch

- Buchstabe tippen, dann  $\boxed{\text{Strg}}-\boxed{\text{G}}$ . Die Zuordnung zeigt Tabelle 6.1.
- Auswahl aus den Seitenpanels „Symbole (A-Ω)“ oder „Symbole (α-ω)“


| Name    | groß        | klein           | Name    | groß       | klein        |
|---------|-------------|-----------------|---------|------------|--------------|
| Alpha   | A <b>A</b>  | a $\alpha$      | Ny      | N <b>N</b> | n $\nu$      |
| Beta    | B <b>B</b>  | b $\beta$       | Xi      | X $\Xi$    | x $\xi$      |
| Gamma   | G $\Gamma$  | g $\gamma$      | Omikron | O <b>O</b> | o $\omicron$ |
| Delta   | D $\Delta$  | d $\delta$      | Pi      | P $\Pi$    | p $\pi$      |
| Epsilon | E <b>E</b>  | e $\varepsilon$ | Rho     | R <b>P</b> | r $\rho$     |
| Zeta    | Z <b>Z</b>  | z $\zeta$       | Sigma   | S $\Sigma$ | s $\sigma$   |
| Eta     | H <b>H</b>  | h $\eta$        | Tau     | T <b>T</b> | t $\tau$     |
| Theta   | Q $\Theta$  | q $\theta$      | Ypsilon | U <b>Y</b> | u $\upsilon$ |
|         |             | J $\vartheta$   | Phi     | F $\Phi$   | f $\varphi$  |
| Iota    | I <b>I</b>  | i $\iota$       |         |            | j $\phi$     |
| Kappa   | K <b>K</b>  | k $\kappa$      | Chi     | C <b>X</b> | c $\chi$     |
| Lambda  | L $\Lambda$ | l $\lambda$     | Psi     | Y $\Psi$   | y $\psi$     |
| My      | M <b>M</b>  | m $\mu$         | Omega   | W $\Omega$ | w $\omega$   |

Tabelle 6.1.: Griechische Buchstaben und die zugehörigen lateinischen Buchstaben, aus denen sie mit  $\boxed{\text{Strg}}\text{-}\boxed{\text{G}}$  entstehenden. Rot: Nicht als griechische Buchstaben erkennbare Zeichen, bitte nicht verwenden.

### 6.1.3. Maßeinheiten

Maßeinheiten und andere vordefinierte Konstanten werden in Formelbereichen **blau** und in aufrechter Schrift (nicht kursiv) dargestellt. Letzteres entspricht den Regeln für wissenschaftliche Arbeiten. Dort sind Variablen kursiv und Maßeinheiten aufrecht zu schreiben.

Maßeinheiten werden durch Variablen gleichen Namens nicht überschrieben. Intern werden sie durch ein vorangestelltes Hochkomma ( $'$ ) markiert. Zur Eingabe von Maßeinheiten gibt es verschiedene Optionen:

- Symbol tippen, gegebenenfalls dynamische Hilfe korrigieren ( $\boxed{\uparrow}$  oder  $\boxed{\downarrow}$ ) und mit  $\boxed{\Leftrightarrow}$  bestätigen. Wird die Auswahl nicht mit  $\boxed{\Leftrightarrow}$  bestätigt, ist der eingetippte Bezeichner eine normale Variable. So kann man die Variable  $m$  definieren, ohne die Einheit  $m$  unbrauchbar zu machen.
- Hochkomma tippen, dann Einheitsname. Die dynamische Hilfe erscheint, muss aber nicht bestätigt werden.
- **Einfügen** > **Einheit** > Dialog „Einheit einfügen“
- **Werkzeugleiste** > 
- $\boxed{\text{Strg}}\text{-}\boxed{\text{W}}$  ruft den Dialog „Einheit einfügen“ ebenfalls auf.

In der Liste „Dimension/Eigenschaft“ kann man eine Vorauswahl treffen.

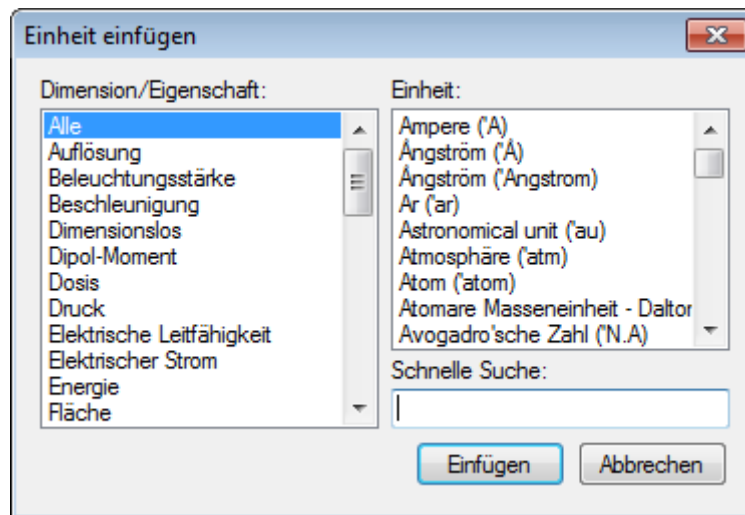


Abbildung 6.2.: Dialog „Einheit einfügen“

Auf das Rechnen mit Maßeinheiten wird näher im Kapitel 9 eingegangen.

#### 6.1.4. Internes Format

Der Inhalt jedes Formelbereichs wird intern als eine Textzeile dargestellt (eindimensionale oder serialisierte Form). Diese Darstellung wird in der Zwischenablage und beim Schreiben und Lesen symbolischer Ausdrücke mit den Funktionen `rfile()` und `wfile()` verwendet.

Das interne Format entspricht weitgehend der Zeichenfolge, mit der Ausdrücke von der Tastatur eingegeben werden.

Viele Operatoren werden intern durch Funktionen dargestellt und können mitunter auch nur durch Verwendung des Funktionsnamens eingegeben werden (siehe Anhang B).

Man kann das interne Format eines Formelbereiches oder eines Teils davon ermitteln, indem man ihn markiert und in die Zwischenablage kopiert. Fügt man den Inhalt in einen existierenden Textbereich ein, erhält man die interne Darstellung.

Ein Beispiel: Ein markierter Ausdruck

$$d_{A1} := \sqrt[4]{I_{A1} \cdot \frac{64}{\pi}}$$

wird mit **Strg**-**C** in die Zwischenablage kopiert. Er ist dort als

```
d.A1:nthroot(I.A1*64/π,4)
```

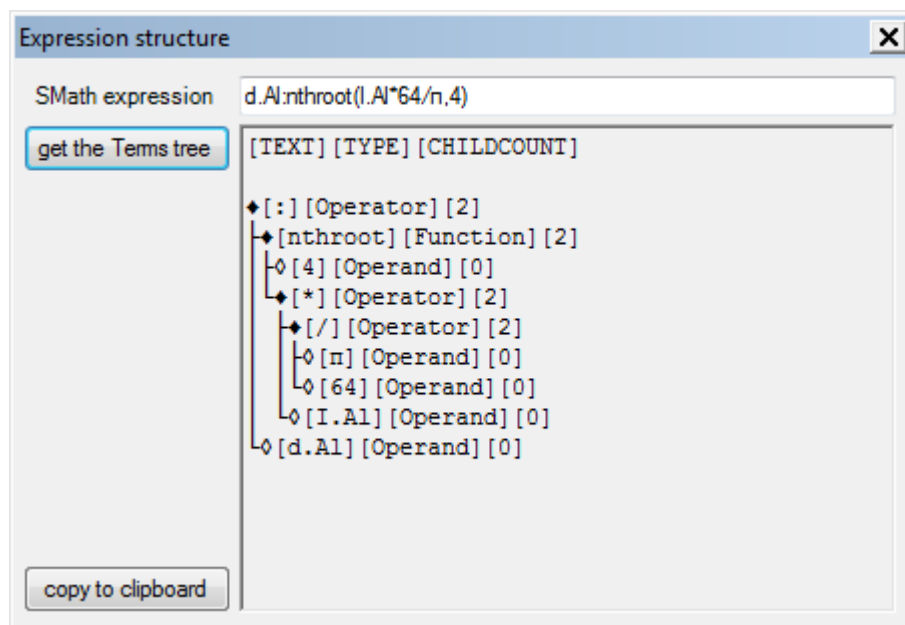
gespeichert.

In den `.sm`-Dateien wird dieses Format nicht verwendet, dort kommt eine XML-Darstellung in umgekehrter polnischer Notation (UPN) zur Anwendung. Die Struktur der Formel lässt sich mit dem Entwicklerwerkzeug „Math to terms“ anzeigen:

**Einfügen** > **Development Tools** > **Math's visual debug**

Ausdruck aus der Zwischenablage ins Feld „SMath expression“ einfügen

„get the Terms tree“ drücken



Der Strukturbaum zeigt in jeder Zeile den Text des Elements, den Typ und die Zahl eventueller Operanden.

### 6.1.5. Formatierung

Formelbereiche können als Ganzes einheitlich formatiert werden (der Bereich muss dazu hellblau markiert oder aktiv sein, also Mausfokus haben). Die folgenden Werkzeuge stehen dafür in der Werkzeugleiste zur Verfügung:



Schriftgröße, Textfarbe, Hintergrundfarbe, Rahmen ein/aus.

Die Schriftart in Formeln kann nicht geändert werden. Sie wird durch eine XML-Datei im Installationsverzeichnis definiert.

## 6.2. Grundbegriffe

**Ausdrücke** sind berechenbare Kombinationen aus Variablen, Funktionen, Operatoren, Zahlenkonstanten, Zeichenketten und Maßeinheiten.

**Namen** sind Zeichenfolgen zur Identifikation von Variablen, Funktionen oder Maßeinheiten. Namen dürfen nicht mit einer Ziffer beginnen, sonst sind alle irgendwie eingebbaren Zeichen erlaubt. Selbst Operatorzeichen wie +-/[\*] oder Leerzeichen kann man verwenden, wenn man vorher mit  $\text{Strg}+\text{K}$  einen erweiterten Eingabemodus für Namen einschaltet (dabei wird die Einfügemarke rot). Tun Sie das nur, wenn es der Lesbarkeit Ihrer Dokumente hilft!

**Variable** sind Namen, denen mit dem Zuweisungsoperator := ein Ausdruck (Inhalt der Variable) zugewiesen wurde. Dieser Ausdruck wird eventuell vor der Zuweisung noch vereinfacht. Den tatsächlich gespeicherten Inhalt zeigt die dynamische Hilfe oder das Debugger-Fenster.

Variable werden zunächst in Abschnitt 6.6 erläutert und dann nochmals in Abschnitt 10.1.2 im Zusammenhang mit benutzerdefinierten Funktionen.


**Funktionen** Namen, denen in runde Klammern gesetzte Argumente folgen. Ihnen wird eine Berechnungsvorschrift zugewiesen, in der die Argumente vorkommen.

`Funktionsname(Argument1; Argument2; ...)`

Sie liefern beim Auswerten einen Wert, der von den Argument- (oder Parameter-)werten abhängig ist. Zwischen den Argumenten steht das unter **Extras** > **Einstellungen** > **Interface** einstellbare Argumenttrennzeichen. Die Argumente werden bei Aufruf der Funktion durch Ausdrücke ersetzt.

Funktionen sind entweder eingebaut, kommen aus Anwendungserweiterungen (Plugins) oder werden vom Anwender definiert.

Funktionen mit gleichem Namen, aber unterschiedlicher Argumentanzahl sind verschiedene Objekte.

Auf Funktionen wird in Abschnitt 6.5 eingegangen. Benutzerdefinierte Funktionen werden im Kapitel Programmierung, Abschnitt 10.1, erläutert. Einen Index der meisten eingebauten Funktionen findet man im Interaktiven Handbuch: **Werkzeuge** >  > **Funktionsindex**.

**Operatoren** sind spezielle Symbole, die zugehörige Ausdrücke (die Operanden) mathematisch verknüpfen. Man kann sie als Funktionen mit speziellem Anzeigeformat auffassen. Dadurch wird eine möglichst gute Ähnlichkeit mit der mathematischen Notation auf dem Papier erreicht.

Manche Operatoren werden direkt durch entsprechende Zeichen eingegeben, z.B. `+*/!<>`. Manche Funktionen werden als Operator formatiert, damit sie besser lesbar sind. Für deren Eingabe muss man den Funktionsnamen kennen.

Eine Liste aller Operatoren einschließlich der Funktionen mit Operatordarstellung findet sich in Anhang B.

**Zahlen** sind Folgen von Ziffern, die ein Dezimaltrennzeichen enthalten können. Das Vorzeichen ist in SMATH nicht Bestandteil der Zahlenkonstante, sondern ein unärer Operator (Operator mit genau einem Operand). Zahlen werden auch zur Darstellung logischer Werte benutzt:

- Null für Falsch
- Ungleich Null (typischerweise Eins) für Wahr

**Zehnerpotenzen** werden nicht als Bestandteil der Zahlenkonstante wie in vielen Programmiersprachen üblich (Beispiel: `2.4E-3`), sondern als die entsprechende Rechenoperation dargestellt und eingegeben (Beispiel  $2,4 \times 10^{-3}$ ).

**Komplexe Zahlen** werden als Summe aus dem Realteil (Zahlenkonstante) und dem Imaginärteil (Produkt aus imaginärer Einheit  $i$  und einer Zahlenkonstante) dargestellt.

|                        |                        |  |  |
|------------------------|------------------------|--|--|
| <code>1 = 2 = 0</code> | <code>1 = 1 = 1</code> | <code>25 km = 2,5 · 10<sup>7</sup> mm</code> | <code>e<sup>i · π/4</sup> = 0,707 + 0,707 · i</code> |
|------------------------|------------------------|--|--|

**Zeichenketten** sind beliebige, in Anführungszeichen eingeschlossene Zeichenfolgen. Sie werden einschließlich der Anführungszeichen in rotbrauner Farbe dargestellt. Funktionen für Zeichenketten werden im Kapitel „Programmierung“, Abschnitt 10.4 erläutert.

**Maßeinheiten:** Namen, die mit einem Apostroph beginnen. Das Apostroph wird in der Anzeige unterdrückt und der Name wird blau dargestellt. Multiplikationszeichen vor Maßeinheiten werden unterdrückt. Der Umgang mit Maßeinheiten ist Gegenstand des Kapitels 9.

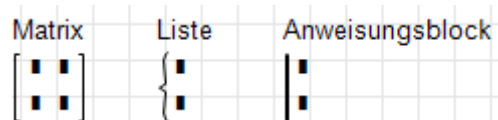
**Felder:** Anordnungen von mehreren Ausdrücken, die durch Elementindizes angesprochen werden können.

SMath kennt drei Arten von Feldern:

**Matrizen:** zweidimensionale Felder, die mit einem oder zwei Indizes angesprochen werden können und die in eckigen Klammern angezeigt werden. Interne Darstellung: `mat()`

**Listen:** eindimensionale Felder, die mit einer geschweiften Klammer von links angezeigt werden. Interne Darstellung: `sys()`

**Anweisungsblöcke** eindimensionale Felder, deren Ausdrücke bei Aufruf zeilenweise ausgeführt werden. Eine Elementadressierung ist in der Regel sinnlos. Sie werden links durch eine vertikale Linie begrenzt. Interne Darstellung: `line()`



**Trennzeichen** Argumenttrennzeichen in Funktionen und Dezimaltrennzeichen in Zahlen sind von der individuellen Programmeinstellung abhängig (**Extras** > **Einstellungen** > **Interface**).

In **Deutschland** ist das Komma als Dezimaltrennzeichen üblich. Daher kann es nicht als Argumenttrennzeichen verwendet werden, wie es in vielen Programmiersprachen üblich ist. Man weicht daher auf das Semikolon aus.

$$\frac{1}{3} = 0,333 \quad f(x; y) := x \cdot y$$

Die **englischsprachige** Standardeinstellung ist der Punkt als Dezimaltrennzeichen (Dezimalpunkt) und das Komma als Argumenttrennzeichen.

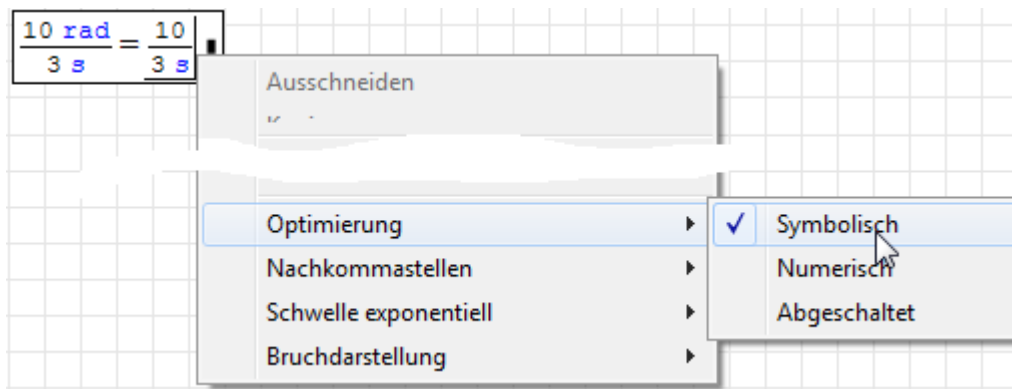
**Kontext** Alle bei der Auswertung eines Ausdrucks bekannten Definitionen bilden den sogenannten Kontext. Bei der Auswertung eines Ausdrucks werden die Definitionen der darin enthaltenen Variablen, Funktionen und Maßeinheiten eingesetzt und dann das Ergebnis berechnet. Den Kontext kann man sich im Debugger-Fenster anschauen (Abschnitt 10.3).

## 6.3. Symbolische und numerische Berechnung

SMath Studio kennt zwei Arten der Berechnung (oder allgemeiner: der Auswertung) von Ausdrücken: symbolisch und numerisch.

Die Auswertungsart heißt im SMath-Programm „Optimierung“ und kann im Kontextmenü eines jeden Formelbereichs überprüft oder umgestellt werden:

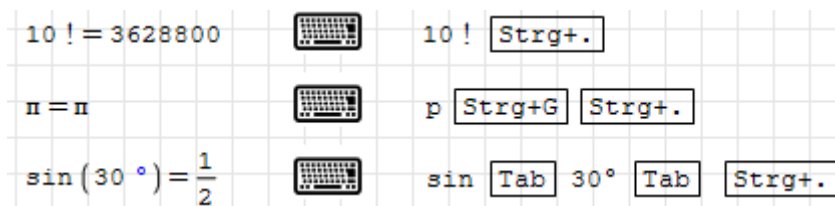




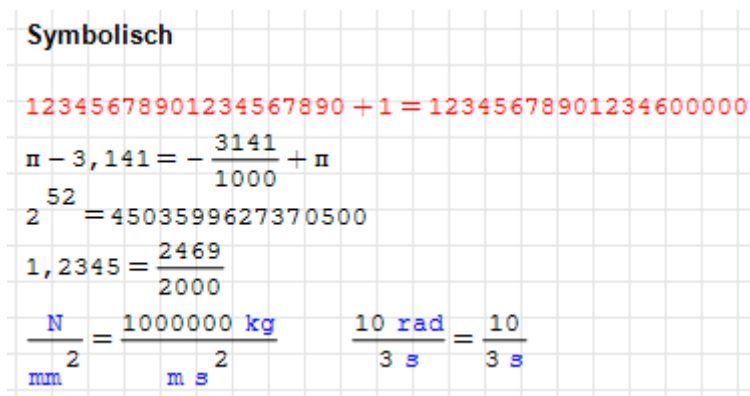
Die Option „abgeschaltet“ kann nützlich sein, wenn man symbolische Vereinfachungen unterdrücken möchte.

Auch ganze Zahlen (Integer) und logische Werte werden in SMATH Studio durch Fließkommazahlen dargestellt.

**Symbolisch:** Es wird versucht, das Ergebnis exakt, ohne Rundung zu bestimmen. Dabei kann SMATH Studio in gewissen Grenzen auch symbolische Umformungen (Vereinfachung) durchführen. Die symbolische Auswertung eines Formelbereichs wird mit **Strg**-**.** oder mit **Seitenleiste** > **Arithmetik**: **→** durchgeführt.



Dezimalbrüche werden als rationale Zahlen dargestellt. Maßeinheiten werden durch Basiseinheiten ausgedrückt.

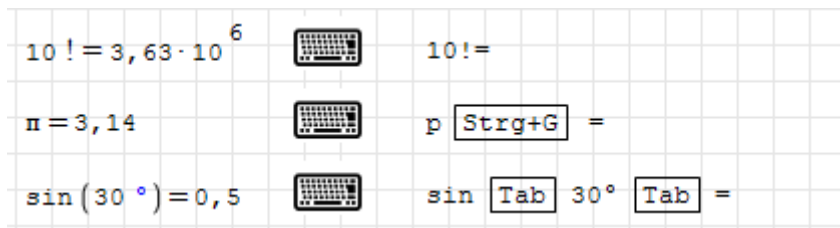


Das rote Beispiel zeigt, dass SMATH auch bei symbolischen Berechnungen die auf 15 Mantissenstellen begrenzte Fließkommadarstellung für Zahlen nach IEEE 754 (1985) verwendet. Anderenfalls würden die Ziffern weiter rechts bei Addition von 1 nicht verloren gehen. Eine exakte Fließkomma-Arithmetik ist in SMATH nicht implementiert.

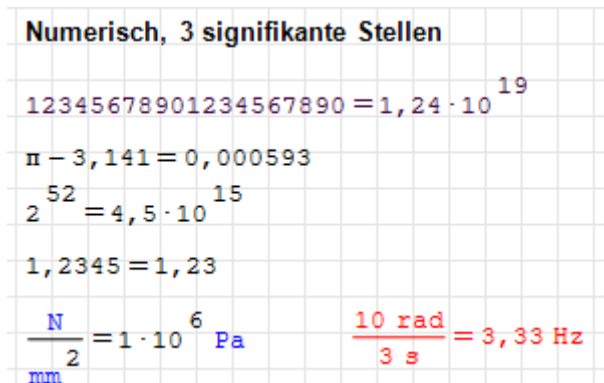
Die Einstellungen zur Ergebnisformatierung haben bei symbolischer Auswertung keinen Einfluss.

**Numerisch:** Es wird versucht, einen Zahlenwert zu berechnen, der dann gemäß den aktuellen Einstellungen für die Ergebnisdarstellung angezeigt wird. Die numerische

Auswertung wird mit = durchgeführt. Das Zeichen kann entweder über die Tastatur oder über **Seitenleiste > Arithmetik: [=]** eingegeben werden.



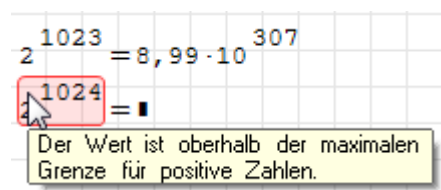
Die Ergebnisformatierung ist hier auf drei signifikante Stellen eingestellt.



SMATH versucht, eine für das Ergebnis sinnvolle Maßeinheit anzugeben. Dies gelingt nicht immer. Das rote Beispiel unten rechts zeigt eine irreführende Verwendung einer abgeleiteten SI-Einheit ([SS-2395](#)). Man kann aber im Zweifelsfall selbst eine Maßeinheit vorgeben ([Abschnitt 9.1](#)).

### 6.3.1. Rechengenauigkeit

Sowohl bei numerischen als auch symbolischen Berechnungen basiert die Zahlendarstellung auf den Fließkommastandard IEEE 754 (1985). SMATH Studio verwendet also keine exakte Gleitkommaarithmetik, wie sie in anderen Mathematikprogrammen (z.B. Maxima) vorhanden ist. Die größte darstellbare Zahl ist  $2^{1023}$




Das entspricht einer internen Darstellung als doppelt genaue (*double precision*) Fließkommazahl mit 64 Bit breiter Kodierung. Dabei stehen 1 Bit für das Vorzeichen, 11 Bits für den Exponenten und 52 Bits für die Mantisse zur Verfügung.

SMATH kann demnach nicht zwischen den Zahlen  $2^{52}$  und  $2^{52} + 1$  unterscheiden, Zahlen können also nur auf 15 Dezimalstellen genau sein. Ziffern an der 16. und weiteren Mantissenstellen sind also unzuverlässig.


$$2^{52} = 4503599627370500$$
$$2^{52} + 1 = 4503599627370500$$
$$\log_{10}(2^{52}) = 15,7$$

Praktisch bedeutet das, dass ab einem Kontostand von zweistelligen Billionenbeträgen Cent-Werte unter den Tisch fallen.

Die Begrenzungen der Zahlendarstellung gelten auch bei symbolischer Auswertung


numerische Auswertung:  `40!=`

$$40! = 8,16 \cdot 10^{47}$$

symbolische Auswertung:  `40!`

$$40! = 8159152832478980000000000000000000000000000000000$$

Will man mit beliebig langen exakten Ganzzahlen rechnen, kann man dies mit Maxima erledigen. Dabei muss man verhindern, dass SMath den Ausdruck  $40!$  vor Aufruf der Funktion auswertet und hinterher das Ergebnis „optimiert“:

Optimierung abgeschaltet  `Maxima("$40!")`

$$\text{SI}("$40!") = 815915283247897734345611269596115894272000000000$$

## 6.4. Format der Ergebnisanzeige

Bei numerischer Auswertung kann die Ergebnisanzeige mit folgenden Einstellungen modifiziert werden.

- Zahl der Nachkommastellen oder signifikanten Ziffern (mit oder ohne nachfolgende Nullen nach dem Komma)
- Exponentialschwelle (ab welcher Zahl von Stellen vor dem Komma auf Exponentialdarstellung umgeschaltet wird)
- Bruchdarstellung (dezimal, mit Bruchstrich, automatisch)
- Abtrennung eines vorgegebenen Faktors (Maßeinheit, Bezugsgröße)
- Art der Rundung (mathematisch, kaufmännisch)

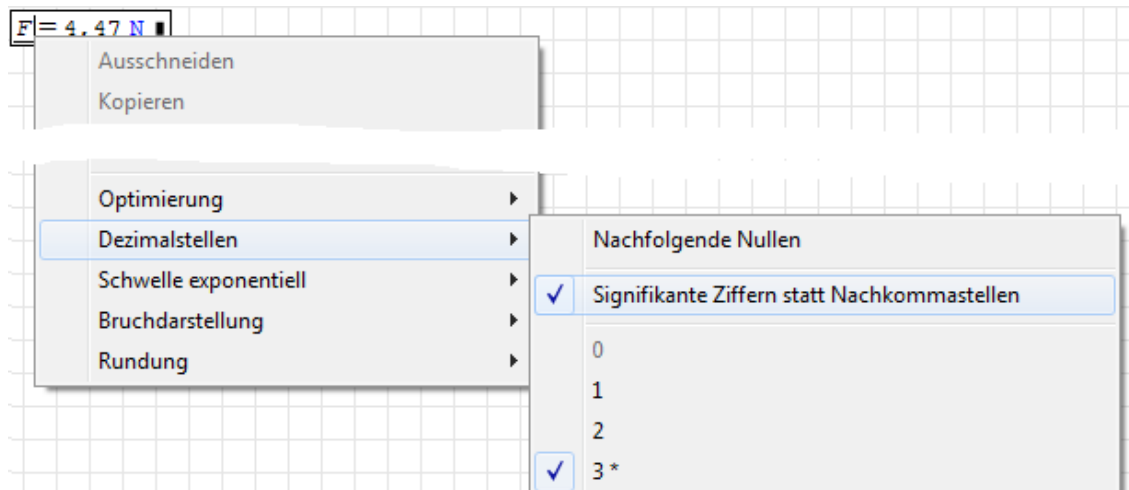
### 6.4.1. Darstellung von Dezimalzahlen

Für die Steuerung der Anzeigegenauigkeit gibt es zwei Varianten:

- Zahl der Nachkommastellen. Diese gibt an, an welcher Stelle nach dem Komma Dezimalbrüche mit Rundung abgeschnitten werden. Damit wird eine absolute Genauigkeit eingestellt.
- Zahl der signifikanten Stellen. Das ist die Anzahl der gültigen Ziffern in der Mantisse. Diese Zahl stellt eine relative Genauigkeit dar.

In beiden Fällen kann angegeben werden, ob rechts stehende Nullen nach dem Komma angezeigt werden sollen. Wenn sie angezeigt werden, dann kann man sich sicher sein, dass diese Ziffern tatsächlich Null sind und nicht nur weggerundet wurden. Andererseits ist die Anzeige von Nullen nach dem Komma besonders bei ganzen Zahlen lästig, so dass man dies lieber nicht zur Voreinstellung macht.

Die Anzeigegenauigkeit wird mit **Kontextmenü** > **Dezimalstellen** eingestellt. Die aktuelle Einstellung für den Formelbereich ist mit einem  $\checkmark$  und die Voreinstellung mit einem \* gekennzeichnet.



Hier einige Beispiele für die Auswirkung (blau hinterlegt):

| Dezimalstellen: 3                                 |                         |                    |             |
|---|-------------------------|--------------------|-------------|
| Signifikante Ziffern statt Nachkommastellen: Ja   |                         |                    |             |
| Nachfolgende Nullen: Nein                         |                         |                    |             |
| $\pi = 3,14$                                      | $10 \cdot \pi = 31,4$   | $12,3456 = 12,3$   | $1 = 1$     |
| Nachfolgende Nullen: Ja                           |                         |                    |             |
| $\pi = 3,14$                                      | $10 \cdot \pi = 31,4$   | $12,3456 = 12,3$   | $1 = 1,00$  |
| Signifikante Ziffern statt Nachkommastellen: Nein |                         |                    |             |
| Nachfolgende Nullen: Nein                         |                         |                    |             |
| $\pi = 3,142$                                     | $10 \cdot \pi = 31,416$ | $12,3456 = 12,346$ | $1 = 1$     |
| Nachfolgende Nullen: Ja                           |                         |                    |             |
| $\pi = 3,142$                                     | $10 \cdot \pi = 31,416$ | $12,3456 = 12,346$ | $1 = 1,000$ |

Die erste Ergebniszeile wird mit 3 signifikanten Ziffern ohne nachfolgende Nullen angezeigt. Diese Einstellung wird im vorliegenden Handbuch standardmäßig benutzt.

Im Kontextmenü kann die **Exponentialschwelle** eingestellt werden, also ab welcher Zehnerpotenz Zahlen in Exponentialform dargestellt werden. Die Voreinstellung für neue Formelbereiche kann unter **Extras** > **Einstellungen** > **Berechnung** > **Schwelle exponentiell** gesetzt werden. Die Voreinstellung „5“ bedeutet, dass Zahlen mit mehr als 5 Stellen vor dem Komma als Produkt aus Mantisse und Zehnerpotenz angezeigt werden. Die Anzeige der Mantisse wird wie bei anderen Zahlen gesteuert.

|                               |                                       |
|-------------------------------|---------------------------------------|
| <b>Exponentialschwelle: 5</b> | $1 \cdot \pi = 3,14$                  |
|                               | $10 \cdot \pi = 31,4$                 |
|                               | $100 \cdot \pi = 314$                 |
|                               | $1000 \cdot \pi = 3140$               |
|                               | $10000 \cdot \pi = 31400$             |
|                               | $100000 \cdot \pi = 3,14 \cdot 10^5$  |
|                               | $1000000 \cdot \pi = 3,14 \cdot 10^6$ |

**Rundung.** Bei der Darstellung wird gegebenenfalls gerundet. Dies erfolgt mathematisch (Geradzahregel) oder kaufmännisch (5 immer betragsmäßig aufrunden). Die Auswahl kann unter **Extras** > **Einstellungen** > **Berechnung** > **Rundung** voreingestellt oder im Kontextmenü des Formelbereichs einzeln gesetzt werden.

| <b>Rundung: mathematisch</b> | <b>Rundung: kaufmännisch</b> |
|------------------------------|------------------------------|
| $2,335 = 2,34$               | $2,335 = 2,34$               |
| $2,345 = 2,35$               | $2,345 = 2,35$               |
| $2,355 = 2,36$               | $2,355 = 2,36$               |




Der rot gekennzeichnete Ausdruck wird falsch gerundet<sup>1</sup>.

### 6.4.2. Abtrennen eines Faktors (Zehnerpotenz, Konstante, Maßeinheit)

Von jedem numerischen oder symbolischen Ergebnis kann man einen beliebig vorgebbaren Faktor abspalten. Sobald man ein angezeigtes Ergebnis anklickt, erscheint ein Platzhalter ■, in den man den gewünschten Faktor eingibt. Dieser Mechanismus ist für Maßeinheiten gedacht, er funktioniert aber auch mit Zahlen und Variablen.

|                              |
|------------------------------|
| $123456 = 1,23 \cdot 10^5$   |
| $10^3$                       |
| $123456 = \blacksquare 10^3$ |
| $\swarrow$                   |
| $123456 = 123 \cdot 10^3$    |


Im folgenden Beispiel wird der Wert von  $\arcsin(1)$  einmal unmodifiziert, dann als Vielfaches von  $\pi$  und dann in der Maßeinheit  $^\circ$  angezeigt:

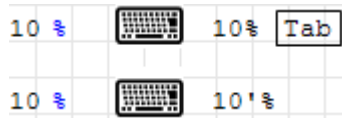
|                         |   |       |     |    |                    |
|-------------------------|---|-------|-----|----|--------------------|
| $\arcsin(1) = 1,57$     |  | arcsi | Tab | 1= | Enter              |
| $\arcsin(1) = 0,5 \pi$  |  | arcsi | Tab | 1= | → → p Strg+G Enter |
| $\arcsin(1) = 90^\circ$ |  | arcsi | Tab | 1= | → → ° Tab Enter    |

<sup>1</sup>[smath.info/bts/Issues/IssueDetail.aspx?id=3518](http://smath.info/bts/Issues/IssueDetail.aspx?id=3518)

### 6.4.3. Prozentrechnung

Für Prozentrechnung gibt es eine Maßeinheit %. Diese kann über die Tastatur als '%', über die dynamische Hilfe oder über den Dialog „Einheit einfügen“ eingegeben werden.

Um etwa z.B. 10% einzugeben, tippen Sie 10'%' oder 10%  :







Das Prozentzeichen steht für einen Faktor von 0,01 und kann damit weitgehend intuitiv verwendet werden.

|                       |                                |                               |
|-----------------------|--------------------------------|-------------------------------|
| 15% von 23            | $15 \% \cdot 23 = 3,45$        |                               |
| 20 von 39             | $\frac{20}{39} = 51,3 \%$      |                               |
| 123 erhöht um 19%     | $123 + 123 \cdot 19 \% = 146$  | $123 \cdot (1 + 19 \%) = 146$ |
| 123 vermindert um 30% | $123 \cdot (1 - 30 \%) = 86,1$ |                               |

Es stehen auch weitere Einheiten wie Promille und ppm (parts per million) zur Verfügung, zu finden unter **Einfügen > Einheit > Dimension/Eigenschaft: Dimensionslos**.

## 6.5. Funktionen und Operatoren

Rechenoperationen können intern und im Rechenblatt entweder durch Funktionen oder durch Operatoren dargestellt werden. Funktionen erkennt man am Namen mit direkt nachfolgenden eingeklammerten Argumenten. Operatoren sind Sonderzeichen zwischen den beteiligten Argumenten oder auch nur spezielle Anordnungen von Argumenten (Beispiel für letzteres: Potenzierung und Elementindex).

|             |   |           |  |
|-------------|---|-----------|--|
| $1 + 1 = 2$ |  | $1+1=$    | Operator +   |
| $a^b$       |  | $a^b$     | Potenzoperator ^   |
| $M_2$       |  | $M[2$     | Funktion el() mit Operator- und Tastenkürzel [                 |
| $ -2  = 2$  |  | $abs(-2=$ | Funktion abs() mit Operator- und Tastenkürzel (Betragsstriche) |

Zur Lesbarkeit von SMATH-Dokumenten trägt die Operator-Formatierung einiger Rechenoperationen bei, die intern als Funktionen geschrieben werden, also als `Funktionsname(Argument1; Argument2...)`.

Im Anhang B sind alle verfügbaren Operatoren mit ihrer Bedeutung und der Tastatureingabe aufgelistet.

In der SMath-Benutzeroberfläche wird als Operator nur das bezeichnet, was auch intern durch Sonderzeichen ohne Klammern um die Argumente dargestellt wird. Bei der Eingabe von Operatoren werden die benötigten Operanden durch Platzhalter (■) dargestellt, die mit Ausdrücken ausgefüllt werden müssen.

Eine Übersicht kann man sich in den Dialogen unter **Einfügen** > **Operatoren** und **Einfügen** > **Funktionen** verschaffen.

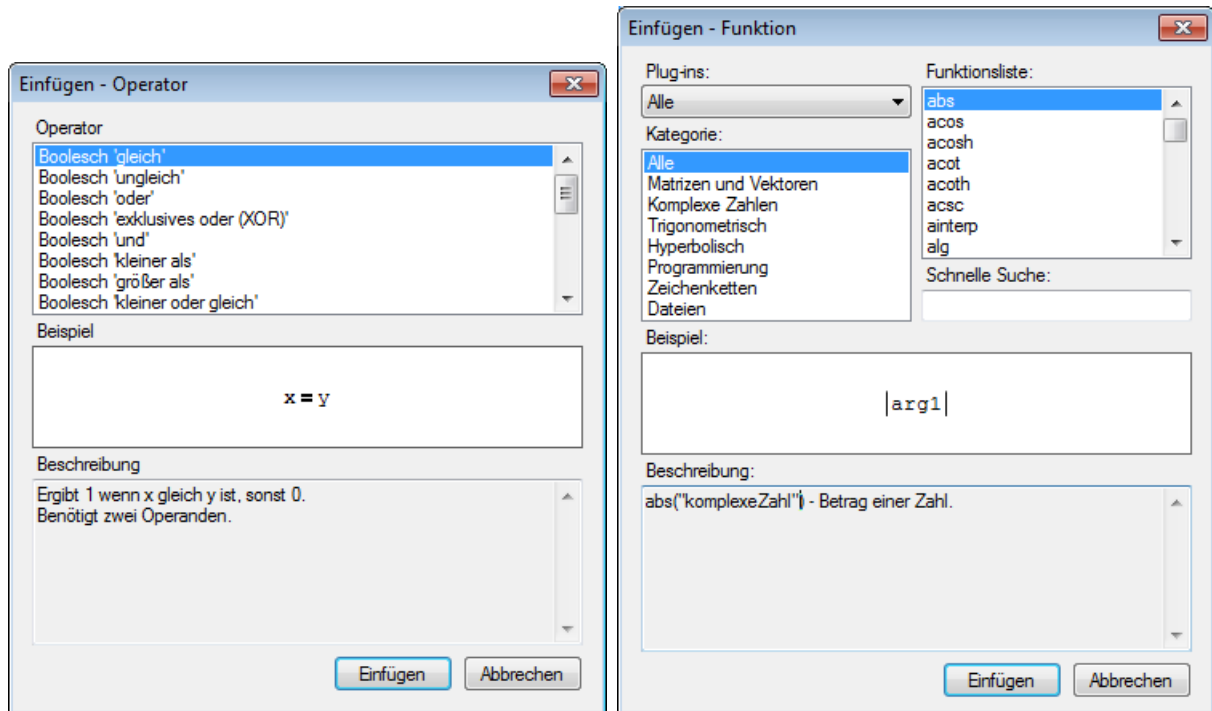


Abbildung 6.3.: Dialoge „Einfügen - Operator“ und „Einfügen - Funktion“

Es gibt vordefinierte Funktionen (Namen erscheinen aufrecht) und benutzerdefinierte Funktionen (Namen erscheinen kursiv). Der Funktionsaufruf besteht aus dem Funktionsnamen und in Klammern den durch Semikolons (Argumenttrennzeichen) getrennten Argumenten.

$$f(x; y) := x \cdot y$$

$$f(2; 3) = 6 \quad \sin(30^\circ) = 0,5$$

Funktionen können mit dem Dialog „Einfügen - Funktion“ ausgewählt werden. Dieser Dialog wird aufgerufen durch

- **Einfügen** > **Funktion**
- Werkzeugleiste >  $fx$
- **Strg** - **E**

In diesem Dialog können die Funktionen nach dem Plug-in, in welchem sie definiert sind, und nach Kategorie gefiltert werden.

Winkelfunktionen erwarten ihr Argument im Bogenmaß. Mit der Maßeinheit  $^\circ$  kann man sich aber behelfen.

$$\sin\left(\frac{\pi}{2}\right) = 1 \quad \sin(90^\circ) = 1$$

Beim Eintippen des Namens einer Funktion in einem Formelbereich bietet die dynamische Hilfe eine Liste passender bekannter Funktionsnamen an.

Die Definition eigener Funktionen wird im Kapitel „Programmierung“ im Abschnitt 10.1 erläutert.

### 6.5.1. Winkelfunktionen

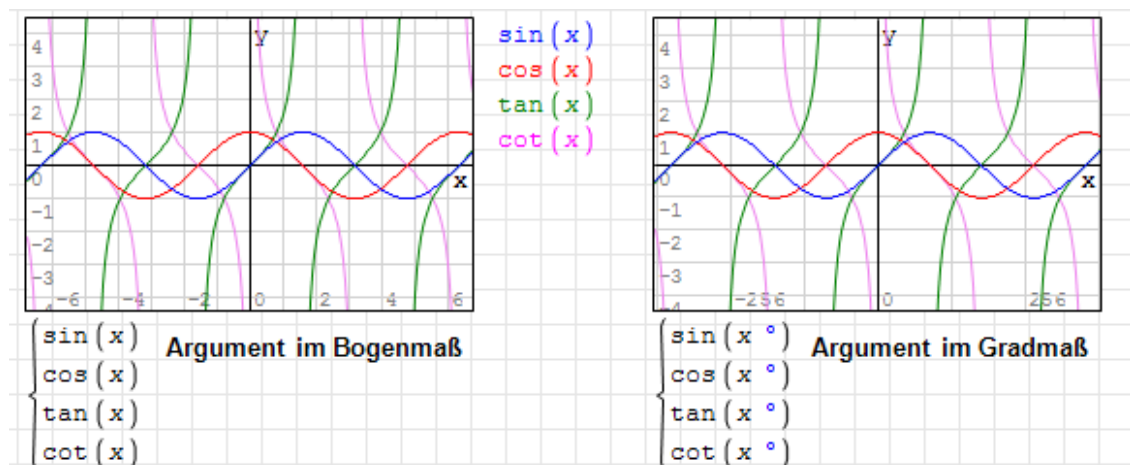
Die Standard-Winkeinheit ist Radiant (Bogenmaß). Für die Eingabe oder Anzeige von Winkeln im Gradmaß benutzen Sie die Einheit  $^\circ$  (Grad).

$$x * \left[ \leftarrow \right] \left[ x \right] \left[ \circ \right]$$

21

|                          |                             |                                     |
|--------------------------|-----------------------------|-------------------------------------|
| $\varphi := 2 \cdot \pi$ | $\varphi = 6,28$            | Radiant                             |
|                          | $\varphi = 360^\circ$       | Grad                                |
|                          | $\varphi = 400 \text{ Gon}$ | Neugrad, Gon (% vom rechten Winkel) |

Die Namen der Winkelfunktionen können unter **Extras** > **Einstellungen** > **Interface** > **Funktionsstil**: auf Europa oder Weltstandard gestellt werden. Im vorliegenden Handbuch wird die Europa-Einstellung verwendet.

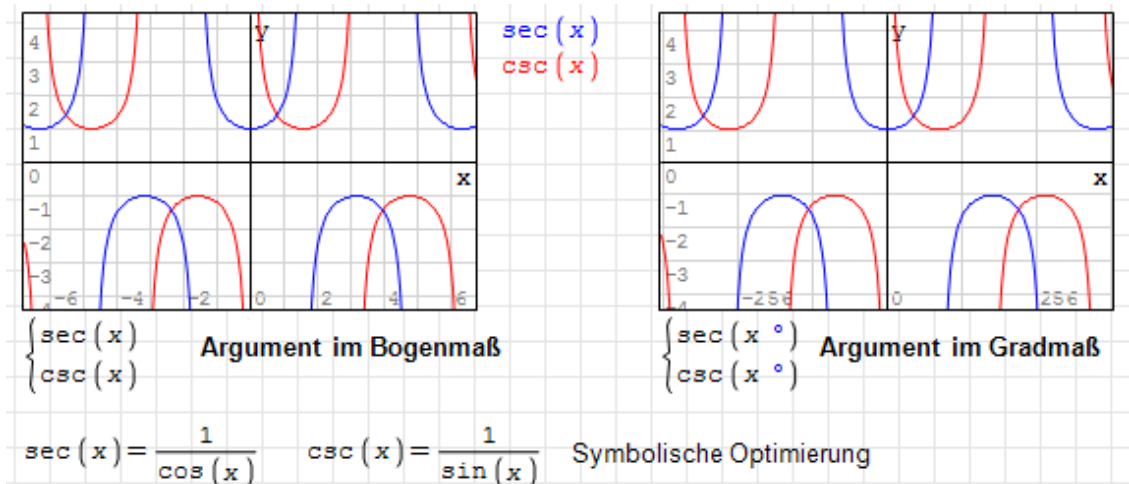


Einige Werte sind symbolisch bekannt (symbolische Auswertung).

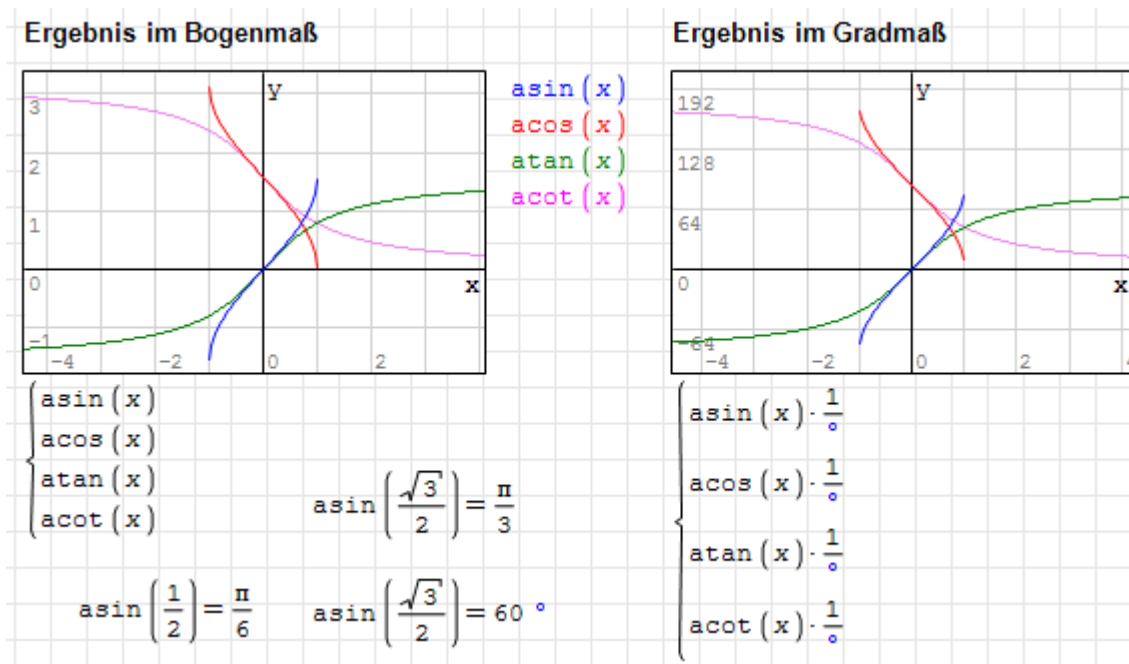
|                                |                                       |                                       |                                       |                                 |
|--------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------|
| $\sin(0^\circ) = 0$            | $\sin(30^\circ) = \frac{1}{2}$        | $\sin(45^\circ) = \frac{1}{\sqrt{2}}$ | $\sin(60^\circ) = \frac{\sqrt{3}}{2}$ | $\sin(90^\circ) = 1$            |
| $\cos(0^\circ) = 1$            | $\cos(30^\circ) = \frac{\sqrt{3}}{2}$ | $\cos(45^\circ) = \frac{1}{\sqrt{2}}$ | $\cos(60^\circ) = \frac{1}{2}$        | $\cos(90^\circ) = 0$            |
| $\tan(0^\circ) = 0$            | $\tan(30^\circ) = \frac{1}{\sqrt{3}}$ | $\tan(45^\circ) = 1$                  | $\tan(60^\circ) = \sqrt{3}$           | $\tan(90^\circ) = \blacksquare$ |
| $\cot(0^\circ) = \blacksquare$ | $\cot(30^\circ) = \sqrt{3}$           | $\cot(45^\circ) = 1$                  | $\cot(60^\circ) = \frac{1}{\sqrt{3}}$ | $\cot(90^\circ) = 0$            |

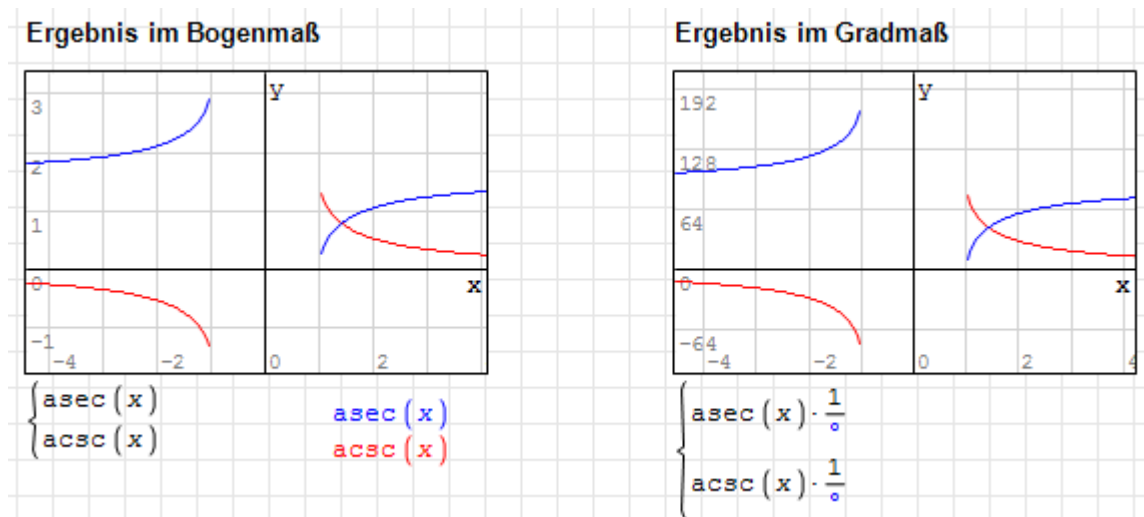


Die Sekans- und Kosekans-Funktion werden nicht direkt bereitgestellt, sondern intern auf den Sinus und Kosinus zurückgeführt.



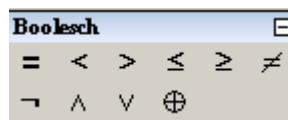
Umkehrfunktionen der Winkelfunktionen. Das Ergebnis der Umkehrfunktionen ist ein Winkel. Um ihn im Gradmaß anzuzeigen, wird wieder die Maßeinheit ° benutzt.





### 6.5.2. Logische und Vergleichsoperationen

SMath verfügt über die üblichen logischen und Vergleichsoperationen. Diese liefern den Zahlenwert 1 wenn sie wahr sind und 0 wenn sie falsch sind. Operanden gelten als falsch, wenn der Wert 0 ist, und als wahr, wenn der Wert ungleich Null ist. Die Operationen sind über **Seitenleiste** > **Boolesch** erreichbar, zudem gibt es einige Tastaturkürzel.



| Funktion          |                       | Taste                      | Priorität gegenüber := |
|-------------------|-----------------------|----------------------------|------------------------|
| $\sim$            | Logische Negation     |                            |                        |
| $\mathbf{=}$      | Logisches Gleich      | $\text{Strg} + \mathbf{+}$ | niedriger              |
| $\mathbf{<}$      | Kleiner als           | $\mathbf{<}$               | höher                  |
| $\mathbf{>}$      | Größer als            | $\mathbf{>}$               | höher                  |
| $\mathbf{\leq}$   | Kleiner oder gleich   | $\text{Strg} - \mathbf{0}$ | höher                  |
| $\mathbf{\geq}$   | Größer oder gleich    | $\text{Strg} - \mathbf{0}$ | höher                  |
| $\mathbf{\neq}$   | Ungleich              | $\text{Strg} - \mathbf{3}$ | niedriger              |
| $\mathbf{\wedge}$ | Logisches Und         | $\mathbf{\&}$              | niedriger              |
| $\mathbf{\vee}$   | Logisches Oder        | $\mathbf{ }$               | niedriger              |
| $\mathbf{\oplus}$ | Exklusives Oder (XOR) |                            | höher                  |

Tabelle 6.3.: Binäre logische Operatoren und Negation (unär)

Aufgrund der niedrigen Priorität einiger Operatoren muss die rechte Seite von Zuweisungen in solchen Fällen in Klammern gesetzt werden.

```
(a := b) & c      [Keyboard] a:b& [Arrow] c   erst Zuweisung, dann UND
a := (b & c)     [Keyboard] a:(b&c         erst UND, dann Zuweisung
```

Das Verhalten der Operatoren kann man anhand einer Wahrheitstabelle prüfen:

```
a := [0 0 1 1]  b := [0 1 0 1]
for i ∈ [1..4]
  T 1 i := (a i = b i)
  T 2 i := (a i ≠ b i)
  T 3 i := (a i ∧ b i)
  T 4 i := (a i ∨ b i)
  T 5 i := (a i ⊕ b i)
  t := [
    "a"
    "b"
    "S:a=b"
    "S:a≠b"
    "S:a&b"
    "S:a|b"
    "S:a⊕b"
  ]
  stack(a; b; T)
```

|     |   |   |   |   |
|-----|---|---|---|---|
| a   | 0 | 0 | 1 | 1 |
| b   | 0 | 1 | 0 | 1 |
| a=b | 1 | 0 | 0 | 1 |
| a≠b | 0 | 1 | 1 | 0 |
| a∧b | 0 | 0 | 0 | 1 |
| a∨b | 0 | 1 | 1 | 1 |
| a⊕b | 0 | 1 | 1 | 0 |

### Ternäre Operatoren

Ternäre Operatoren sind solche mit drei Operanden, also Ausdrücke wie  $a < x < b$ . Versucht man solche in SMath Studio mit den Zeichen  $>$  oder  $<$  einzugeben, dann wird deutlich, dass dies nicht das gewünschte Ergebnis bringt:

```
(a < x) < b      [Keyboard] a<x< [Arrow] b
```

Der Vergleich  $a < x$  liefert 0 oder 1 und das wird dann mit  $b$  verglichen.

Man muss solche Intervalle also umständlich mit Ausdrücken wie  $(a < x) \wedge (x < b)$  darstellen. Als lesefreundlichere Alternative stellt das Plugin *Custom Functions* ternäre logische Operatoren bereit. Sie werden intern durch Funktionen dargestellt, deren Namen sich aus den Operatoren ableiten lassen.

```
a < x < b      [Keyboard] ltlt [Tab] x [Arrow] a [Arrow] b
```

| Operator               | Funktion      |                            |
|------------------------|---------------|----------------------------|
| <b>■ ≤ ■ ≤ ■</b>       | lele(##;##;#) |                            |
| <b>■ ≤ ■ &lt; ■</b>    | lelt(##;##;#) |                            |
| <b>■ &lt; ■ ≤ ■</b>    | ltle(##;##;#) | le ≤ less than or equal    |
| <b>■ &lt; ■ &lt; ■</b> | ltlt(##;##;#) | lt < less than             |
| <b>■ ≥ ■ ≥ ■</b>       | gege(##;##;#) | ge ≥ greater than or equal |
| <b>■ ≥ ■ &gt; ■</b>    | gegt(##;##;#) | gt > greater than          |
| <b>■ &gt; ■ ≥ ■</b>    | gtge(##;##;#) |                            |
| <b>■ &gt; ■ &gt; ■</b> | gtgt(##;##;#) |                            |

Tabelle 6.4.: Ternäre logische Operatoren aus dem Plug-in *Custom Functions*

### 6.5.3. Stückweise stetige Funktionen

Stückweise stetige Funktionen sind solche, bei denen in verschiedenen Argumentbereichen unterschiedliche Ausdrücke für den Funktionswert gelten. Dadurch sind Sprünge und Knicke im Funktionsverlauf möglich.

Zu empfehlen ist die Verwendung der `cases()`-Funktion (Fallunterscheidung), da dies im Rechenblatt leicht lesbar ist und robust funktioniert. Entscheidungsalternativen werden hinzugefügt durch zweimalige Eingabe des Argumenttrennzeichens (Semikolon):

```

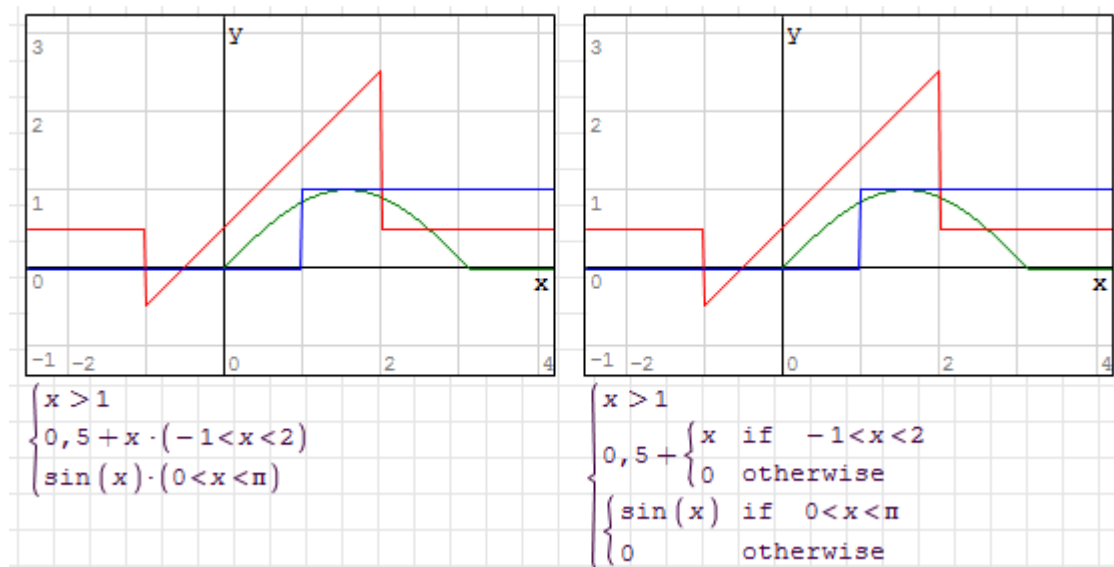
{
  if
  otherwise
}
{
  if
  if
  otherwise
}

```

cas Tab

cas Tab ;;

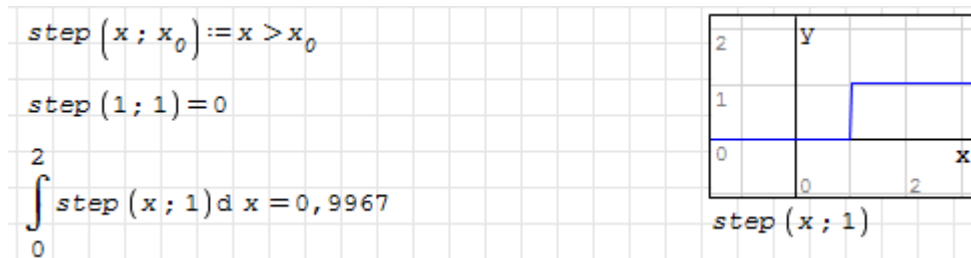
Logische Ausdrücke funktionieren ebenfalls und führen mitunter auf eine kompaktere Darstellung.



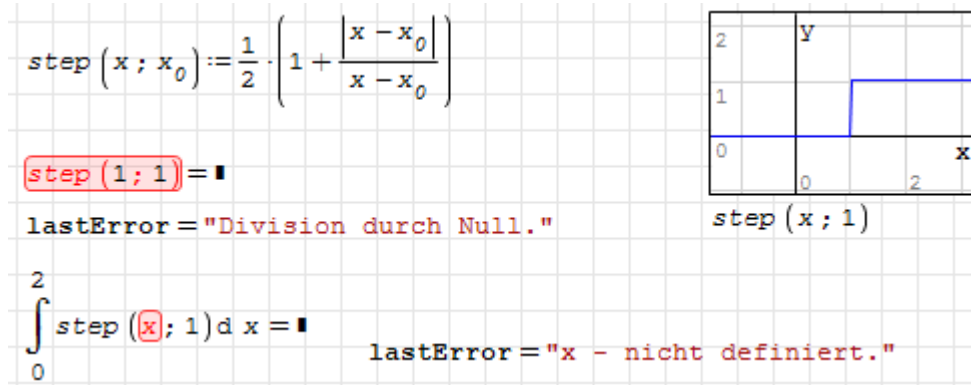
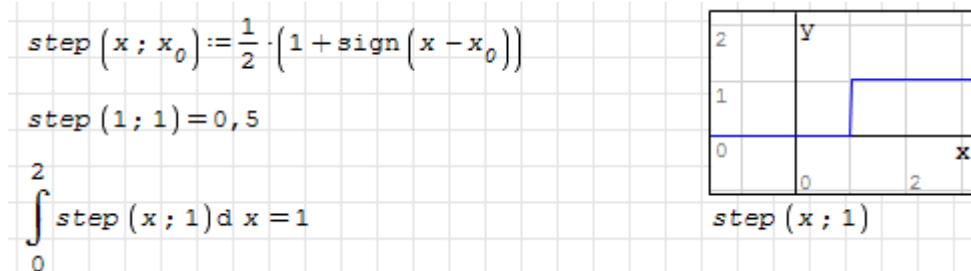
Im Folgenden demonstrieren wir verschiedene Darstellungen der Sprungfunktion  $\text{step}(x, x_0)$ , die an der Stelle  $x = x_0$  von Null auf den Wert 1 wechselt. Die Varianten unterscheiden sich

- im Wert, der an der Sprungstelle  $x_0$  geliefert wird. Das ist eine Definitionsfrage.
- ob die numerische Integration möglich ist (bei einigen Varianten wird die sinnlose Fehlermeldung „x nicht definiert“ geliefert).
- im Wert des numerisch bestimmten Integrals. Die Unterschiede ergeben sich primär durch die unterschiedlichen Werte bei  $x = x_0$ . Der dadurch entstehende Fehler ist maximal die halbe Intervallbreite (bei Sprunghöhe 1), also 0,005 (StandardEinstellung 100 Intervalle).

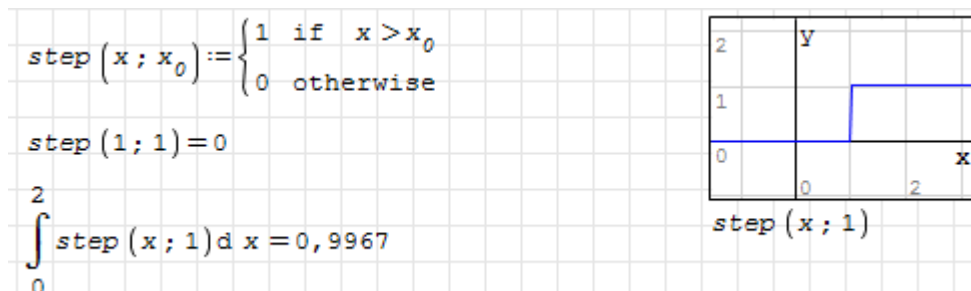
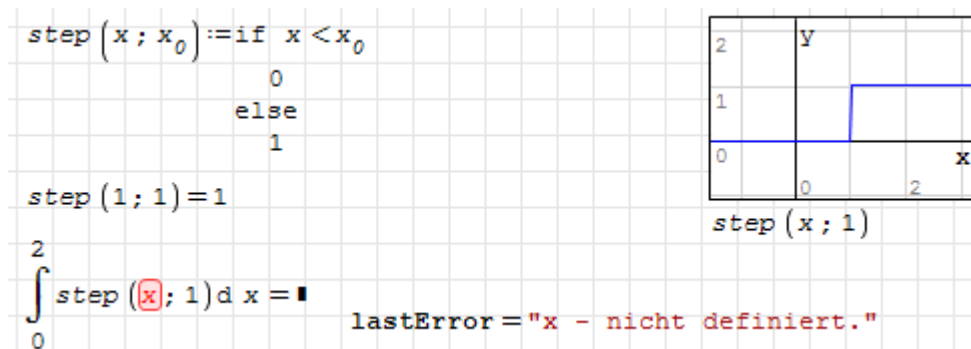
**Logische Ausdrücke**, die abhängig von der Erfüllung der Bedingung 0 oder 1 liefern



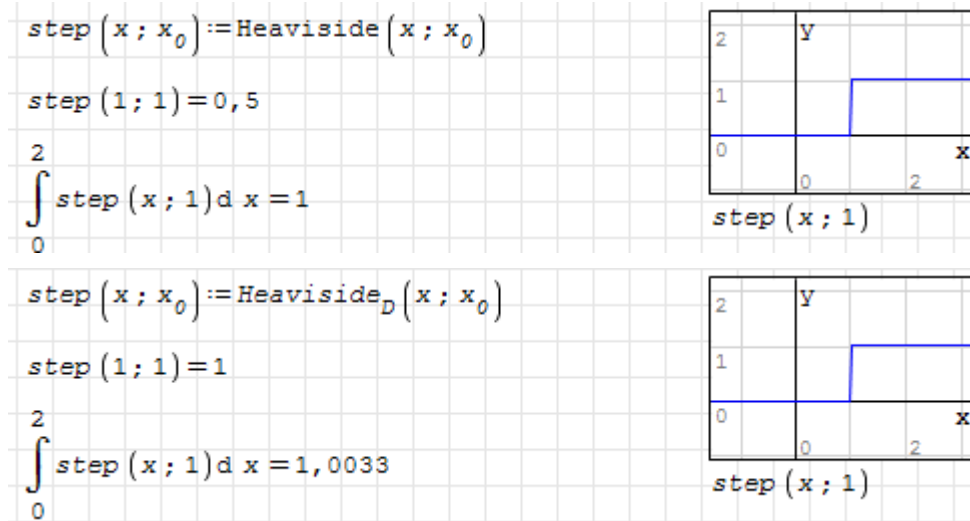
## Vorzeichenfunktion sign() und Betragsfunktion abs()



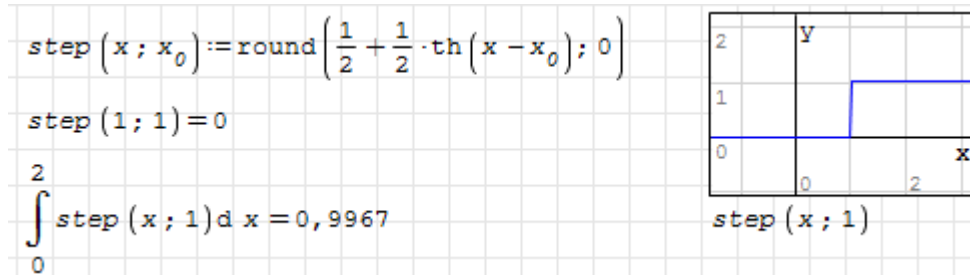
## Bedingte Anweisungen if() und cases()



**Sprungfunktionen** Heaviside() und Heaviside.D() aus dem Plugin *Statistical Tools*



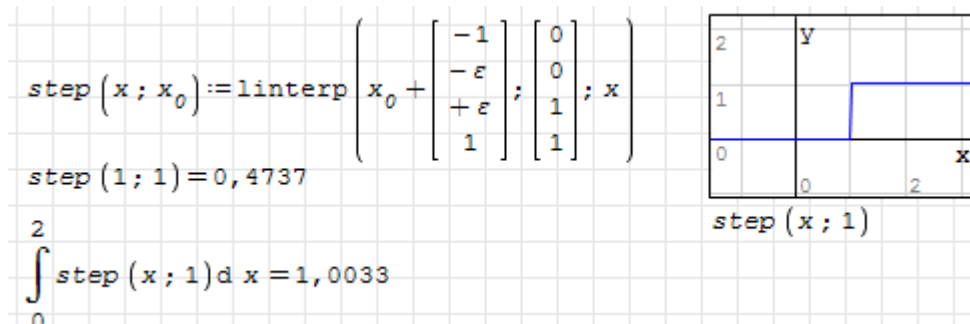
**Gerundeter Hyperbeltangens** (nur eine Spielerei)



**Lineare Interpolation** geeigneter Stützstellen. Da die Stützstellen in  $x$  streng monoton sein müssen, wird zunächst der kleinste als von Null verschieden akzeptierte Abstand ermittelt (Störung  $\varepsilon$ , Maschinengenauigkeit).

```
n := -10 while (1 + 10n) ≠ 1 ε := 10n+1 = 1 · 10-15
n := n - 1
```

Damit weiß man, wie eng die Stützstellen an der Sprungstelle beieinander liegen dürfen:



## 6.6. Rechnen mit Variablen

Variable sind Namen, für die ein Wert oder Ausdruck gespeichert wurde. Die Speicherung des Ausdrucks heißt Zuweisung oder Definition. Dafür wird der Zuweisungsoperator  $:=$

verwendet (einzutippen als :)

```
ρ := 7,85  $\frac{\text{g}}{\text{cm}^3}$    r   Strg+G   :7,85g   Tab   /cm   Tab   ^3
V := 1 l           V:1l   Tab
```

Die dynamische Hilfe kennt alle so definierten Variablen und zeigt deren Belegung an

Mit diesen Variablen kann man Ausdrücke bilden und das Ergebnis anzeigen lassen:

```
ρ · V = 7,85 kg   r   Strg+G   *V=
```

Ausdrücke mit Variablen können auch wieder Variablen zugewiesen werden. Dabei passiert dann folgendes:

- Gespeicherte Ausdrücke für die verwendeten Variablen werden eingesetzt.
- Der entstehende Ausdruck wird vereinfacht und das vereinfachte Ergebnis wird dem Namen zugewiesen.

Im Beispiel werden die Werte von  $\rho$  und  $V$  eingesetzt und das Ergebnis als  $m$  gespeichert. Spätere Änderungen von  $\rho$  und  $V$  haben keinen Einfluss mehr auf den Wert von  $m$

```
m := ρ · V   m = 7,85 kg   m
V := 2 l     m = 7,85 kg
```

**Mehrfachzuweisungen:** Man kann mit einer Zuweisung mehrere Variablen definieren. Im folgenden Beispiel wird das benutzt, um die Werte zweier Variablen zu vertauschen. Dabei werden erst die Ausdrücke im Vektor auf der rechten Seite ausgewertet und dann auf die Variablen im Vektor auf der linken Seite zugewiesen.

```
a := 1 m   b := 2 m   [ a ] := [ b ] [ a ] = [ 2 m ]
                [ b ] := [ a ] [ b ] = [ 1 m ]
```

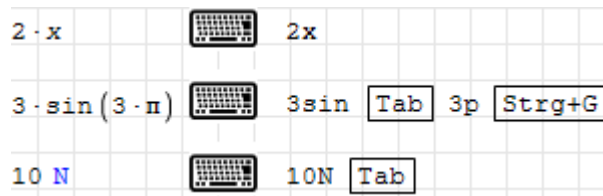
Mehrere Zuweisungen können auch in Matrizen, Listen oder Anweisungsblöcken gruppiert werden. Es gilt aber:

- In jedem Fall werden die Zuweisungen nacheinander ausgeführt, das obige Vertauschungsbeispiel funktioniert so nicht.
- Nur im Fall des Anweisungsblocks kann man sich auf die Reihenfolge der Ausführung verlassen.

```
a := 1 m   b := 2 m   | a := b   [ a ] = [ 2 m ]
                    | b := a   [ b ] = [ 2 m ]
```

Mehrfachzuweisungen können sinnvoll sein, wenn in einem Rechenblatt mehrere Eingangswerte geändert werden müssen, bevor eine Neuberechnung sinnvoll ist. Bei Einzelzuweisungen würde jeweils nach Verlassen des Formelbereichs die Neuberechnung gestartet.

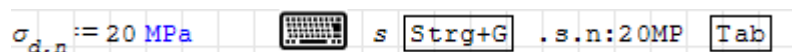
Wird nach einer Zahl ein Buchstabe eingegeben, versteht SMath das automatisch als Multiplikation und fügt einen Punkt ein. Zwischen Zahlen und Einheiten wird kein Punkt eingefügt.



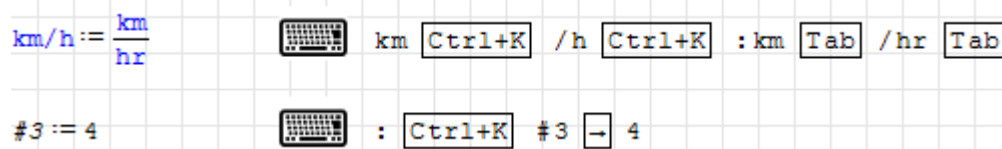
### 6.6.1. Namen

Namen für eigene Funktionen, Variablen und Maßeinheiten bestehen aus Zeichenfolgen, deren **erstes Zeichen keine Ziffer** sein darf. Ansonsten sind fast beliebige Sonderzeichen erlaubt.

Mit einem Punkt oder Komma werden Textindizes eingeleitet (Tiefstellungssymbol). Weitere Punkte oder Kommas werden dann als solche im Namen angezeigt.



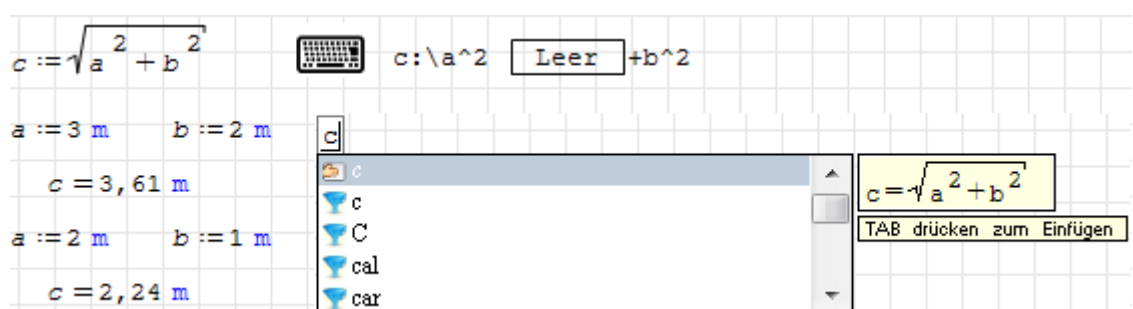
Mit `Strg-K` kann man vorübergehend die Wirkung der Operatorsymbole ausschalten, so dass auch Zeichen wie `*`, `/`, `+`, `-` oder sogar Leerzeichen in Namen eingegeben werden können. In diesem Modus wird die Einfügemarke rot.



### 6.6.2. Definition symbolischer Ausdrücke

Möchte man einen Ausdruck für unterschiedliche Werte der darin enthaltenen Variablen auswerten, muss der Ausdruck als solcher gespeichert werden. Dafür gibt es drei Wege:

- Formel definieren, bevor die Variablen darin einen Wert haben. Beim Auswerten werden dann jedes Mal die aktuellen Werte der Variablen im Rechenblatt eingesetzt.





Will man die Definition der Variablen im Rechenblatt vermeiden, kann man zur Auswertung auch die Substitutionsfunktion `at()` einsetzen:

The screenshot shows a Maple worksheet with the following content:

```

Clear(a; b; c) = 1
c := sqrt(a^2 + b^2)
c | { a = 3 m, b = 2 m } = 3,61 m
c | { a = 2 m, b = 1 m } = 2,24 m

```

Annotations in the image include: `c := \a^2` and `+b^2` for the first line; `at(c)`, `sys`, and `Tab` for the second line.

- Formel in einen Anweisungsblock packen, der gegebenenfalls nur eine einzige Zeile enthält. Alles wird dann gespeichert, wie es ist, das funktioniert auch, wenn die verwendeten Variablen bereits einen Wert haben. Anweisungsblöcke werden mit der Funktion `line()` gebildet.

The screenshot shows a Maple worksheet with the following content:

```

m := rho * V
m := 15,7 kg
V := 2 * V
m = 31,4 kg

```

Annotations in the image include: `m:lin`, `Tab`, `X`, `X`, `r`, and `Strg+G *V` for the first line; `m = rho * V` and `TAB drücken zum Einfügen` for the second line.

- Formel als Funktion mit Argumenten schreiben. Die Argumente bilden einen Eingabekanal, über den die Variablen erst beim Aufruf ihre Werte erhalten:

The screenshot shows a Maple worksheet with the following content:

```

m(rho; V) := rho * V
m(rho; V) = 15,7 kg
m(rho; V/2) = 7,85 kg

```

Annotations in the image include: `m(rho; V) = rho * V` and `TAB drücken zum Einfügen` for the second line.

Mehr zur Definition eigener Funktionen finden Sie im Kapitel Programmierung, Abschnitt 10.1.

Prüfen Sie stets mit der dynamischen Hilfe, ob die Formel tatsächlich symbolisch abgespeichert wurde (und nicht unbeabsichtigt bereits bekannte Werte eingesetzt wurden).

### 6.6.3. Variablen löschen

Zum Löschen von Definitionen dient die Funktion `Clear()`. Auch leere oder ungültige Definitionen entsprechen einer Löschung. Die numerische Auswertung undefinierter Variablen führt zu einer Fehlermeldung.

```

A := 1      B := 1      C := 1
Clear(A) = 1  A = ■      lastError = "A - nicht definiert."
B := ■      B = ■      lastError = "B - nicht definiert."
C :=  $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$   C = ■      lastError = "C - nicht definiert."

```

### 6.6.4. Matrizen und Vektoren

Im Abschnitt 6.8 werden Matrizen und Vektoren ausführlich behandelt. Für die symbolische Vereinfachung gilt, dass SMath Variablen erst durch Zuweisung entsprechender Werte als Matrizen erkennt. Die Zuweisung eines Typs ohne Wert ist nicht möglich.

Daher führt SMath symbolische Vereinfachungen so aus, als wären die Variablen Skalare. Das kann zu Fehlern führen, denn die dabei angewendeten Regeln, wie zum Beispiel die Vertauschbarkeit von Faktoren, gelten nicht für Vektoren und Matrizen.

Die folgenden Ausdrücke sind alle gleichwertig, wenn die Variablen Skalare sind, ihre symbolische Vereinfachung ergibt auch immer das gleiche Ergebnis:

$$(a \cdot b) \cdot c = a \cdot b \cdot c \quad a \cdot (b \cdot c) = a \cdot b \cdot c \quad b \cdot (a \cdot c) = b \cdot a \cdot c$$

Für vektorwertige Variablen gilt dies jedoch nicht:

$$a := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad b := \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \quad c := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$b \cdot c = 10 \quad a \cdot c = 6 \quad a \cdot b = 6$$

Bei der symbolischen Auswertung wird zunächst der Ausdruck vereinfacht, dann werden bekannte Werte eingesetzt. Diese Reihenfolge sorgt dafür, dass die symbolische Vereinfachung nichts davon weiß, falls Vektoren zu multiplizieren sind (falsche Ergebnisse rot):

Optimierung> Symbolisch

$$a \cdot (b \cdot c) = \begin{bmatrix} 6 \\ 12 \\ 18 \end{bmatrix} \quad b \cdot (a \cdot c) = \begin{bmatrix} 6 \\ 12 \\ 18 \end{bmatrix} \quad (a \cdot b) \cdot c = \begin{bmatrix} 6 \\ 12 \\ 18 \end{bmatrix}$$

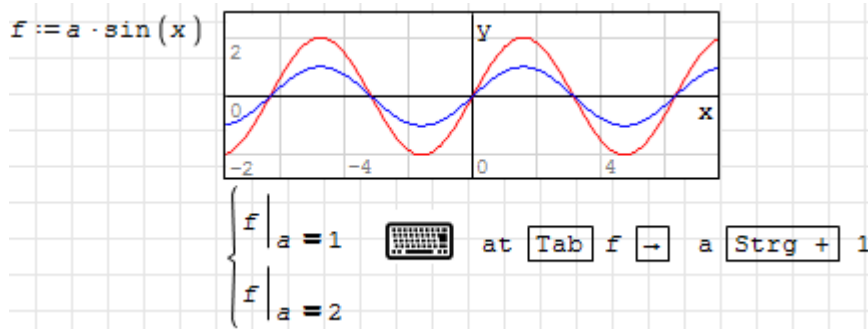
Wird numerische Auswertung angefordert, dann werden bekannte Werte eingesetzt und dann das Ergebnis berechnet. In diesem Fall werden die obigen Ausdrücke auch für vektorwertige Variablen richtig berechnet:

Optimierung> Numerisch

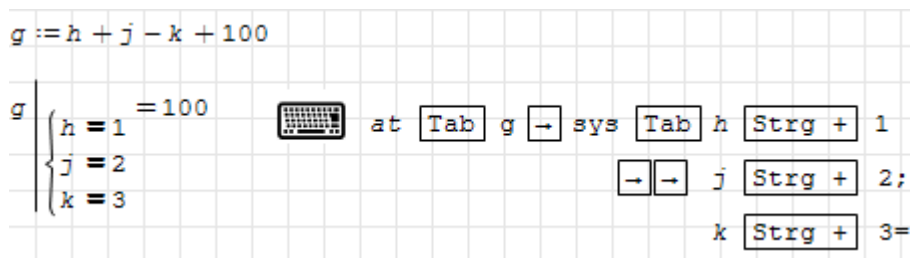
$$a \cdot (b \cdot c) = \begin{bmatrix} 6 \\ 12 \\ 18 \end{bmatrix} \quad b \cdot (a \cdot c) = \begin{bmatrix} 18 \\ 12 \\ 6 \end{bmatrix} \quad (a \cdot b) \cdot c = \begin{bmatrix} 6 \\ 12 \\ 18 \end{bmatrix}$$

### 6.6.5. Die Substitutionsfunktion at()

Die Substitutionsfunktion erlaubt die Ersetzung von Variablen in Ausdrücken. Damit kann der Ausdruck wie eine Funktion mit Argumenten verwendet werden, ohne dass die Variablen im Rechenblatt definiert sein müssen (im Gegenteil, sie dürfen dabei nicht definiert sein).



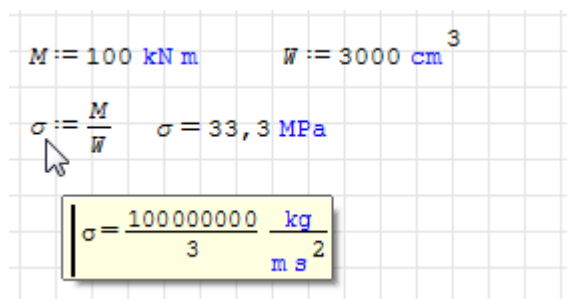
Man kann auch mehrere Werte gleichzeitig ersetzen, diese werden zu einer Liste zusammengefasst:



### 6.6.6. Parametrische Rechnungen

Bei der Definition einer Größe durch einen mathematischen Ausdruck, wird dieser Ausdruck in vereinfachter Form gespeichert. Beispielsweise werden bereits bekannte Variablenwerte oder symbolische Definitionen eingesetzt.

Wenn Sie im Rechenblatt zunächst die gegebenen Größen definieren und dann die abhängigen Größen, werden in den Formeln für die abhängigen Größen nur die Zahlenwerte, nicht aber die Variablennamen gespeichert:



Wenn Sie die Ergebnisse für andere Werte benötigen, dann ändern Sie die Eingabewerte und aktualisieren Ihr Rechenblatt.

Werden die Eingabewerte weiter unten geändert, hat das keine Auswirkungen:

|                             |                             |
|-----------------------------|-----------------------------|
| $M := 100 \text{ kN m}$     | $W := 3000 \text{ cm}^3$    |
| $\sigma := \frac{M}{W}$     | $\sigma = 33,3 \text{ MPa}$ |
| $M := 50 \text{ kN m}$      | Geänderter Eingabewert      |
| $\sigma = 33,3 \text{ MPa}$ | Ergebnis bleibt gleich      |

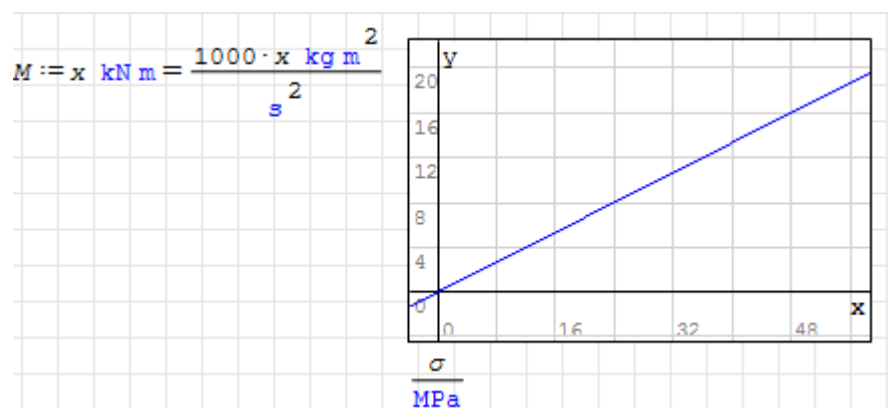
Mitunter wollen Sie weiter unten im Rechenblatt die abhängigen Größen für verschiedene Eingabedaten nebeneinander stellen oder Einflüsse von Eingabedaten in Diagrammen darstellen.

Dann müssen Sie dafür sorgen, dass die Definitionen der abhängigen Größen als symbolische Ausdrücke gespeichert werden. Wie das geht, wird in Abschnitt 6.6.2 beschrieben.

Eine Möglichkeit ist, erst die Formel, dann die Daten zu definieren. Dann wird die Formel nur mit den Variablenamen, nicht aber mit den Werten gespeichert.

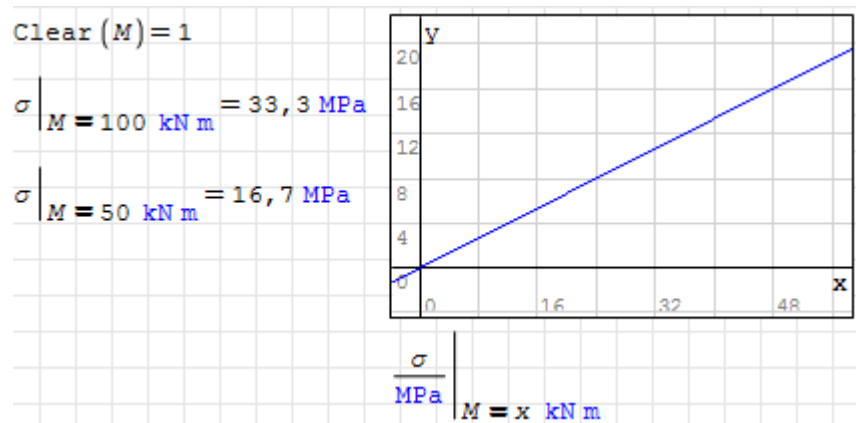
|                         |                          |                             |
|-------------------------|--------------------------|-----------------------------|
| $\sigma := \frac{M}{W}$ |                          |                             |
| $M := 100 \text{ kN m}$ | $W := 3000 \text{ cm}^3$ | $\sigma = 33,3 \text{ MPa}$ |
| $M := 50 \text{ kN m}$  |                          | $\sigma = 16,7 \text{ MPa}$ |

Möchte man ein Diagramm erstellen, benötigt man einen von der dimensionslosen Variable  $x$  abhängigen Ausdruck, der selbst dimensionslos ist. Dazu kann man einen Parameter in der Definition der abhängigen Größe durch  $x$  ersetzen.

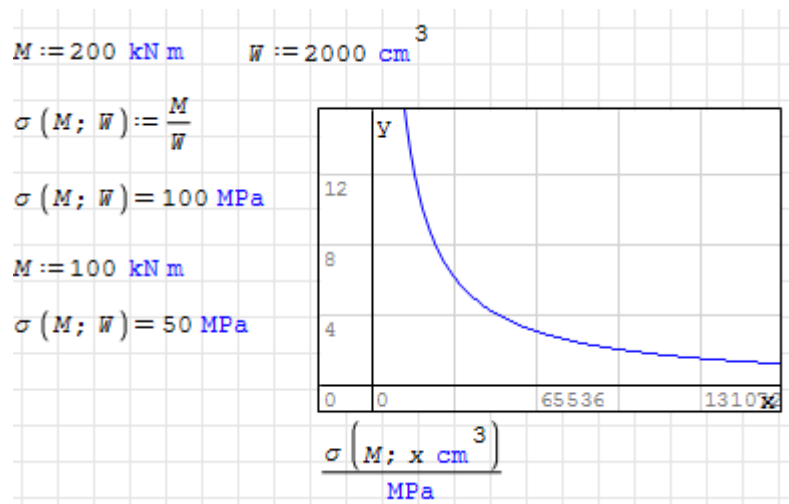


Diese Variante ist allerdings nicht sehr transparent, da dem Ausdruck im Plot nicht anzusehen ist, welche Größe eigentlich auf der  $x$ -Achse variiert wird.

Die Substitutionsfunktion `at()` kann hier helfen, sie funktioniert aber nur für Variablen, die noch keinen Wert haben. Also gegebenenfalls mit `Clear()` löschen.



Alternativ kann man den gleichen Effekt auch erreichen, indem man die abhängigen Größen als explizite Funktionen der zu variierenden Parameter definiert. Dabei ist es dann auch egal, ob die Parameter schon definiert sind.



### 6.6.7. Zwischenspeichern von Variablen in Dateien

Hinweis: Weitere Funktionen zum Lesen und Schreiben von Daten werden im Abschnitt 6.14 erläutert.

Mit den Funktionen `wfile(Ausdruck, Name)` und `rfile(Name)` kann man Zahlenwerte oder symbolische Ausdrücke in eine Datei schreiben und wieder einlesen. Dabei wird das gleiche interne Textformat wie in der Zwischenablage benutzt. De facto handelt es sich tatsächlich um eine Art Zwischenablage mit benannten Einträgen, die von Rechenblättern aus beschrieben und gelesen werden können.

Anwendungen sind beispielsweise:

- Weitergabe von Daten aus einem Rechenblatt in ein anderes,
- vorübergehendes Abspeichern von Werten innerhalb einer Rechnung,
- Aufbewahrung von Werten nach dem Schließen von SMath-Dokumenten.

Beachten Sie, dass das Argument `Name` buchstäblich genommen wird. Es wird nicht ausgewertet und kann keinen Pfad oder Endung enthalten.

Die Datei wird in das Einstellungsverzeichnis (Abschnitt 4.5) geschrieben, welches man mit der Funktion `SettingsDirectory()` anzeigen lassen kann.

Das folgende Beispiel erzeugt eine Datei namens  $t$  mit dem Inhalt 'm/1000

```
t := 1 mm          t = 0,001 m
wfile(t; t) = 1    t = 0,001 m
Clear(t) = 1       t = ■
rfile(t) = 0,001 m t = ■      rfile(■) definiert die Variable nicht
t := rfile(t)      t = 0,001 m neu, sondern liest nur den Wert
```

Das Plugin Mathcad File Access stellt etwas flexiblere Varianten des Funktionspaars bereit. Dabei können Dateiname und Pfad als Zeichenketten angegeben werden:

```
wfile(Ausdruck; Verzeichnis; Dateiname)
rfile(Verzeichnis, Dateiname)
```

Im folgenden Beispiel wird das Verhalten der Originalfunktion nachgebaut, indem das Einstellungsverzeichnis als Pfad und ein als Variablenname geeigneter Dateiname angegeben werden.

```
Plugin customFunctions Dir := SettingsDirectory(1)
Dir := concat(Dir; "user\")
Dir = "C:\FHB\Software\SMath\SMath 99.7219\user\"
FileName := "myfile1"

Plugin MathcadFileAccess wfile(⎡ 4 4 ⎤; Dir; FileName) = 1
Plugin MathcadFileAccess rfile(Dir; FileName) = ⎡ 4 4 ⎤
interne Funktion rfile(myfile1) = ⎡ 4 4 ⎤
```

## 6.7. Komplexe Zahlen

Zur Darstellung komplexer Zahlen wird die imaginäre Einheit  $i$  verwendet. Das ist ein gewöhnlicher Name, der mit  $\sqrt{-1}$  vorbelegt ist. Er wird auch dann aufrecht geschrieben, wenn er vom Anwender überschrieben wurde.

Beachten Sie, dass Real- und Imaginärteil einer komplexen Zahl Maßeinheiten der gleichen Art haben müssen.

```
z1 := 3 + 1 · i    z1 = 3 + i
z2 := 3 m + i s   z2 = ■      lastError = "Einheiten stimmen nicht überein."
z3 := 3 m + i · 100 cm z3 = (3 + i) m
```

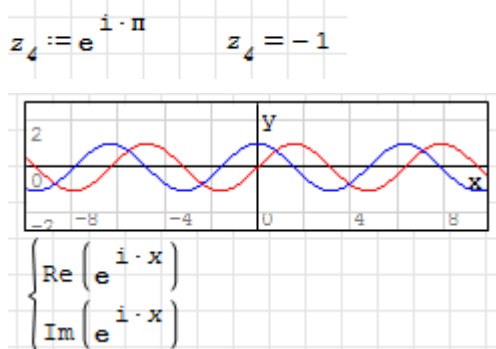
| Funktion     |                          | Operatordarstellung |
|--------------|--------------------------|---------------------|
| Re(#)        | Realteil                 |                     |
| Im(#)        | Imaginärteil             |                     |
| arg(#)       | Argument (Phase)         |                     |
| abs(#)       | Betrag (Radius)          | $ \cdot $           |
| Conjugate(#) | Komplex konjugierte Zahl | $\overline{\cdot}$  |

Tabelle 6.6.: Funktionen für komplexe Zahlen

Die Funktionen für komplexe Zahlen kommen mit Maßeinheiten zurecht:

|                                |                                      |
|--------------------------------|--------------------------------------|
| $z_1 = 3 + i$                  | $z_3 = (3 + i) \text{ m}$            |
| $\text{Re}(z_1) = 3$           | $\text{Re}(z_3) = 3 \text{ m}$       |
| $\text{Im}(z_1) = 1$           | $\text{Im}(z_3) = 1 \text{ m}$       |
| $\text{arg}(z_1) = 18,4^\circ$ | $\text{arg}(z_3) = 18,4^\circ$       |
| $ z_1  = 3,16$                 | $ z_3  = 3,16 \text{ m}$             |
| $\overline{z_1} = 3 - i$       | $\overline{z_3} = (3 - i) \text{ m}$ |

Bei der Definition von Zahlen kann auch die Polarform benutzt werden, eine Anzeige in der Polarform ist nicht direkt möglich.



Die Funktionen zur Koordinatenumrechnung `xy2pol()` und `pol2xy()` entsprechen der Berechnung von Radius und Phase aus Real- und Imaginärteil und umgekehrt:

|  |                |                           |                                |
|--|----------------|---------------------------|--------------------------------|
| $z_1 = 3 + i$  | $ z_1  = 3,16$ | $\text{arg}(z_1) = 0,322$ | $\text{arg}(z_1) = 18,4^\circ$ |
| $\text{xy2pol}(\text{Re}(z_1); \text{Im}(z_1)) = \begin{cases} 3,16 \\ 0,322 \end{cases}$  |                |                           | Radius<br>Phasenwinkel         |
| $\text{pol2xy}( z_1 ; \text{arg}(z_1)) = \begin{cases} 3 - 8,78 \cdot 10^{-17} \cdot i \\ 1 + 1,75 \cdot 10^{-16} \cdot i \end{cases}$ |                |                           | Realteil<br>Imaginärteil       |

Hinweis: Es ist sehr einfach, die imaginäre Einheit  $i$  versehentlich zu überschreiben, z.B. durch Verwendung als Schleifenzähler. Das kann bei der Definition von komplexen Zahlen zu Fehlern führen.

|                  |                                      |  |
|------------------|--------------------------------------|--|
| $i := 0$         | $a := 1 + 3 \cdot i = 1$             | $b := 1 + 3 \cdot \sqrt{-1} = 1 + 3 \cdot i$ |
| $i := 1$         | $a := 1 + 3 \cdot i = 4$             | $b := 1 + 3 \cdot \sqrt{-1} = 1 + 3 \cdot i$ |
| $i := \sqrt{-1}$ | $a := 1 + 3 \cdot i = 1 + 3 \cdot i$ | $b := 1 + 3 \cdot \sqrt{-1} = 1 + 3 \cdot i$ |

Sie sind auf der sicheren Seite, wenn Sie statt  $i$  immer  $\sqrt{-1}$  schreiben.

## 6.8. Matrizen und Vektoren

Matrizen sind rechteckige Anordnungen von mathematischen Ausdrücken. Eine Matrix mit  $m$  Zeilen und  $n$  Spalten heißt  $m \times n$ -Matrix. Vektoren im Sinne der linearen Algebra sind Matrizen mit einer einzigen Spalte.

|   |               |                          |   |
|---|---------------|--------------------------|---|
| $A := \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$  | $A_{22} := A$ | $A_{11} := \text{"A11"}$ | $A_{21} := 33 \text{ m}$                |
| $A = \begin{bmatrix} \text{"A11"} & 2 \\ 33 \text{ m} & \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \\ 5 & 6 \end{bmatrix}$ | $A_{22} = 4$  | $A[2;2]$                 | <input type="text" value="Leer"/> [2;2] |

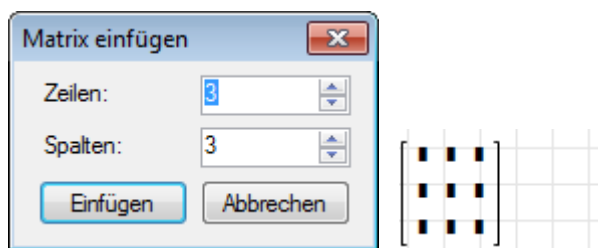
Matrizenelemente können durch Elementindizes angesprochen werden, die mit der eckigen Klammer  $[$  erzeugt werden. Ein zweiter Index wird mit dem Argumenttrennzeichen (Semikolon) erzeugt. Dann adressiert der erste Index die Zeile, der zweite Index die Spalte.

Anmerkung: Wird das Ergebnis der Definition eines einzelnen Matrixelements in der Definition selbst mit  $=$  angezeigt, so erscheint nicht das eben definierte Matrixelement, das der Leser erwarten würde, sondern die ganze Matrix. Das ist ein **Bug** in SMath.

$$K_{22} := 3 = \begin{bmatrix} 0 & 0 \\ 0 & 3 \end{bmatrix}$$

### 6.8.1. Interaktive Erzeugung von Matrizen

Der Dialog „Matrix einfügen“ erlaubt die Angabe der Zeilen- und Spaltenzahl und erzeugt mit  ein entsprechend großes Feld von Platzhaltern



Der Dialog wird aufgerufen mit

- - oder
- **Einfügen** > **Matrix** oder

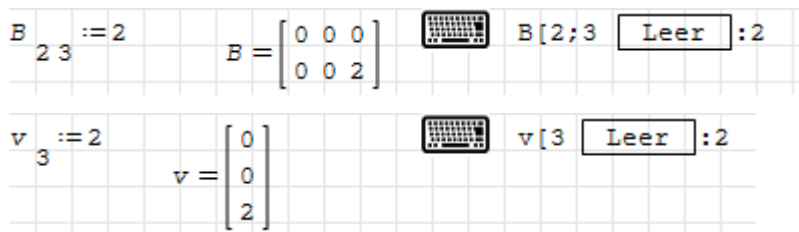


• **Seitenleiste Matrizen** 

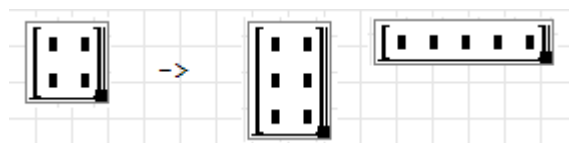
Ein  $2 \times 2$  Platzhalter wird erzeugt durch Eintippen von `mat`  oder `mat(`



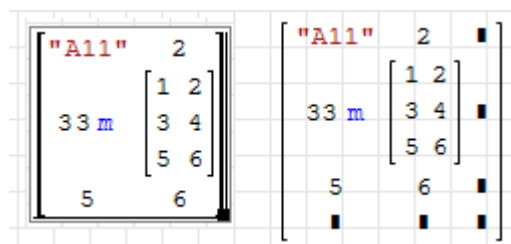
Mit der Zuweisung eines Matrixelements wird eine Matrix der erforderlichen Größe angelegt, wenn sie noch nicht existiert, oder vergrößert, wenn sie nicht groß genug war.



**Größe verändern:** linke oder rechte Klammer anklicken, dann erscheint eine Marke rechts unten, an der die Größe mit der Maus gezogen werden kann.

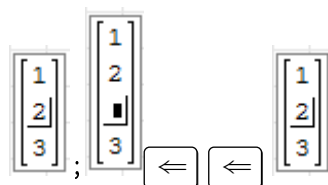


Beim Vergrößern einer Matrix entstehen unten neue Zeilen oder rechts neue Spalten.



Bei Vektoren (einspaltigen Matrizen) lassen sich Zeilen einfügen, indem man am Ende eines Eintrags das Argumenttrennzeichen ; (Semikolon) eingibt.

Entsprechend kann man auch leere Platzhalter löschen, indem man das (unsichtbare) Argumenttrennzeichen davor löscht.



Wegen der Beschränkungen beim Einfügen und Löschen von Zeilen und Spalten ist die interaktive Erzeugung von Datenmatrizen in SMath Studio unbequem.

Für die Datenerfassung können der externe Matrixeditor (siehe Abschnitt 6.8.3) oder die vielfältigen Importfunktionen für Text- und Excel-Dateien (Abschnitt 6.14) genutzt werden.

Zudem gibt es viele Funktionen für das automatische Erzeugen von Matrizen (Abschnitt 6.8.4).

## 6.8.2. Große Matrizen anzeigen

Bei der Ergebnisanzeige von Matrizen kann man an der rechten unteren Ecke mit der Maus ziehen, um die Anzahl der angezeigten Zeilen oder Spalten zu reduzieren. Es erscheinen dann Auslassungszeichen in der Matrix.

```
A := Random(10; 10)
```

$$A = \begin{bmatrix} 0,0562 & 0,924 & 0,935 & 0,224 & 0,164 & 0,913 & & & & \\ 0,503 & 0,874 & 0,874 & 0,668 & 0,593 & 0,0986 & & & & \\ 0,915 & 0,864 & 0,0702 & 0,137 & 0,772 & 0,618 & & & & \\ 1 & 0,407 & 0,152 & 0,837 & 0,0654 & 0,995 & \dots & & & \\ 0,432 & 0,676 & 0,62 & 0,958 & 0,328 & 0,506 & & & & \\ 0,171 & 0,869 & 0,623 & 0,958 & 0,208 & 0,192 & & & & \\ & & & \vdots & & & & & & \ddots \end{bmatrix}$$

Eine Alternative mit erweiterten Formatierungsmöglichkeiten ist der Tabellenbereich (siehe Abschnitt 5.9). Hier kann eingestellt werden, wie groß der anzuzeigende Bereich ist und man kann den gezeigten Bereich verschieben und so die gesamte Matrix sichten.

```
c := [1..cols(A)]^T   r := [1..rows(A)]
```

|     | ... | 2     | 3     | 4     | 5     | 6     | 7     | ... |
|-----|-----|-------|-------|-------|-------|-------|-------|-----|
| ... | ... |       |       |       |       |       |       | ... |
| 2   |     | 0.874 | 0.874 | 0.668 | 0.593 | 0.099 | 0.166 |     |
| 3   |     | 0.864 | 0.07  | 0.137 | 0.772 | 0.618 | 0.962 |     |
| 4   |     | 0.407 | 0.152 | 0.837 | 0.065 | 0.995 | 0.757 |     |
| 5   | ... | 0.676 | 0.62  | 0.958 | 0.328 | 0.506 | 0.319 | ... |
| 6   |     | 0.869 | 0.623 | 0.958 | 0.208 | 0.192 | 0.853 |     |
| 7   |     | 0.796 | 0.001 | 0.928 | 0.638 | 0.096 | 0.722 |     |
| 8   |     | 0.789 | 0.385 | 0.896 | 0.622 | 0.239 | 0.244 |     |
| ... | ... |       |       |       |       |       |       | ... |

Table 1

**Table settings**

Data | Data appearance | Main layout | Res

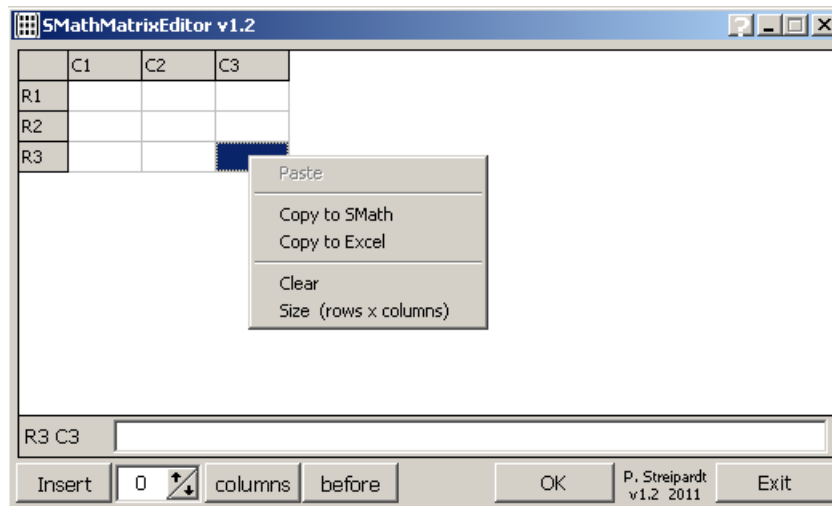
c   show header

r   show left stub

## 6.8.3. Der SMathMatrixEditor

Momentan ist es innerhalb von SMath Studio nicht möglich, Zeilen oder Spalten zu Matrizen hinzuzufügen, lediglich bei Vektoren funktionieren die gleichen Methoden wie bei Listen und Anweisungsblöcken.

Um diesen Mangel auszugleichen, hat Peter Streipardt ein separates Programm geschrieben, den *SMathMatrixEditor*. Er ist Bestandteil der THB SMath-Distribution.



Schritte für das Editieren einer Matrix:

1. Kopieren der Matrix in SMath
2. Einfügen im Matrixeditor
3. Markieren einer Spalte (column) oder Zeile (row) durch Klick auf den jeweiligen Titel
4. Auswahl der Aktion
  - a) Einfügen/Löschen
  - b) Anzahl
  - c) Zeilen oder Spalten
  - d) vor oder nach der aktuellen Zeile oder Spalten
5. Ausführen mit OK
6. Kopieren der Matrix (**Kontextmenü** > **Copy to SMath**)

### 6.8.4. Funktionen zur Erzeugung von Matrizen und Vektoren

**Einheitsmatrix** gegebener Dimension erzeugen:

$$\text{identity}(2) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{identity}(3) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

**Diagonalmatrix** aus einem Vektor oder einer Matrix (von der dann nur die Diagonalelemente übernommen werden):

$$\text{Diag} \left( \begin{bmatrix} 1 \text{ m} \\ 3 \text{ N} \\ 5 \text{ kg} \end{bmatrix} \right) = \begin{bmatrix} 1 \text{ m} & 0 & 0 \\ 0 & 3 \frac{\text{kg m}}{\text{s}^2} & 0 \\ 0 & 0 & 5 \text{ kg} \end{bmatrix} \quad \text{Diag} \left( \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix} \right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 9 \\ 0 & 0 & 0 \end{bmatrix}$$

**Nullmatrix** gegebener Dimension erzeugen:

$$\text{zeros}(2) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{zeros}(2; 3) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\text{matrix}(2; 4) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Man kann auch Matrizen mit Null Spalten oder Null Zeilen erzeugen, z.B. um dann in einer Programmschleife Spalten oder Zeilen hinzuzufügen.

$$\text{matrix}(0; 2) = \text{mat}(0; 2)$$

$$\text{matrix}(3; 0) = \text{mat}(3; 0)$$

**Einsmatrix** gegebener Dimension:

$$\text{ones}(2) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{ones}(2; 3) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

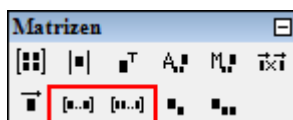
Die Funktion `range()` erzeugt **Vektoren mit auf- oder absteigenden Zahlenwerten**. Anzugeben sind der erste Wert und der Maximalwert. Optional kann der zweite Wert angegeben werden, der dann die Schrittweite definiert. Sie ist auf +1 oder -1 voreingestellt, je nach dem, ob der Maximalwert größer oder kleiner als der Endwert ist.

$$x := [1..5] \quad x = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} \quad [0,5..3,5] = \begin{bmatrix} 0,5 \\ 1,5 \\ 2,5 \\ 3,5 \end{bmatrix} \quad [2..-2] = \begin{bmatrix} 2 \\ 1 \\ 0 \\ -1 \\ -2 \end{bmatrix}$$

$$x_1 := 0 \quad x_2 := 10 \quad x_n := 50$$

$$\text{rang}(x.1..x.n) = \begin{bmatrix} 0 \\ 10 \\ 20 \\ 30 \\ 40 \\ 50 \end{bmatrix}$$

Die Funktion `range()` ist auch über das Panel „Matrizen“ in der Seitenleiste verfügbar.



Man kann Matrizen und Vektoren auch **elementweise erzeugen**. Darauf wird in den Abschnitten 6.8.6 und 6.8.7 eingegangen.

### 6.8.5. Matrixalgebra

Seien Sie extrem vorsichtig mit symbolischer Auswertung (Optimierung) bei Matrizen und Vektoroperationen. SMATH vereinfacht arithmetische Ausdrücke nach den Regeln für skalare Werte (Distributivität und Kommutativität der Multiplikation). Diese gelten im Allgemeinen nicht für Vektoren und Matrizen. **Schalten Sie also im Zweifelsfall auf numerische Auswertung.**

Siehe dazu auch das Beispiel in Abschnitt 6.6.4.

#### Multiplikation mit einer Zahl


$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 3 = \begin{bmatrix} 3 & 6 \\ 9 & 12 \end{bmatrix} \quad 5 \cdot \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 5 & 10 \\ 15 & 20 \end{bmatrix}$$

$$\frac{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}}{3} = \begin{bmatrix} 0,333 & 0,667 \\ 1 & 1,333 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \frac{1}{3} = \begin{bmatrix} 0,333 & 0,667 \\ 1 & 1,333 \end{bmatrix}$$

#### Addition, Subtraktion

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + 5 = \begin{bmatrix} 6 & 7 \\ 8 & 9 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} - 10 = \begin{bmatrix} -9 & -8 \\ -7 & -6 \end{bmatrix}$$


$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 3 & 6 \\ 9 & 12 \end{bmatrix} = \begin{bmatrix} 4 & 8 \\ 12 & 16 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} - \begin{bmatrix} 3 & 6 \\ 9 & 12 \end{bmatrix} = \begin{bmatrix} -2 & -4 \\ -6 & -8 \end{bmatrix}$$

**Transposition** (Eingabe als `transpose()` oder **Seitenleiste** > **Matrizen** > )

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

**Matrixmultiplikation.** Die Spaltenzahl der linken Matrix muss gleich der Zeilenzahl der rechten Matrix sein.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} = \begin{bmatrix} 5 & 11 & 17 \\ 11 & 25 & 39 \\ 17 & 39 & 61 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 7 & 10 \\ 15 & 22 \\ 23 & 34 \end{bmatrix}$$

**Determinante** (Eingabe als `det()` oder **Seitenleiste** > **Matrizen** > ) . Determinanten können grundsätzlich auch symbolisch berechnet werden, aber es gibt keine Möglichkeit, SMATH mitzuteilen, dass  $\lambda$  hier ein Skalar sein soll.

$$C := \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad |C| = a \cdot d - c \cdot b \quad |\lambda \cdot C| = \left| \lambda \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} \right| \quad |3 \cdot C| = 9 \cdot (a \cdot d - c \cdot b)$$

$$A := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad |A| = -1$$

#### Rang einer Matrix

$$A := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{rank}(A) = 2$$

$$D := \begin{bmatrix} a & b \\ 2 \cdot a & 2 \cdot b \end{bmatrix} \quad \text{rank}(D) = 1$$

SMath Studio bekommt schon bei moderat großen Matrizen Probleme mit der numerischen Berechnung der Determinante. Auch wenn die symbolische Auswertung noch Werte liefert, macht deren Darstellung Schwierigkeiten.

$$A := \begin{bmatrix} 8 & 0 & 4 & 8 & 8 & 2 & 8 & 3 & 3 & 2 \\ 5 & 3 & 0 & 7 & 4 & 7 & 5 & 4 & 6 & 7 \\ 5 & 8 & 4 & 1 & 8 & 3 & 1 & 8 & 6 & 9 \\ 6 & 3 & 5 & 0 & 0 & 3 & 8 & 9 & 8 & 0 \\ 5 & 8 & 3 & 5 & 8 & 3 & 4 & 6 & 6 & 5 \\ 9 & 6 & 7 & 2 & 8 & 9 & 6 & 3 & 9 & 5 \\ 1 & 9 & 6 & 4 & 5 & 1 & 8 & 0 & 6 & 1 \\ 9 & 5 & 5 & 5 & 1 & 7 & 5 & 8 & 2 & 4 \\ 8 & 4 & 2 & 1 & 9 & 9 & 5 & 2 & 1 & 1 \\ 6 & 5 & 9 & 3 & 9 & 5 & 1 & 8 & 0 & 7 \end{bmatrix}$$

Numerische Optimierung

$$\text{rank}(A) = \blacksquare$$

$$|A| = \blacksquare$$

symbolische Optimierung

$$\text{rank}(A) = 10$$

$$|A| = \frac{7644853240860760 \cdot (30056222782060300000000)}{\dots}$$

Inverse Matrix (Eingabe als  $\wedge^{-1}$ )

$$\begin{bmatrix} a & b \\ 2 \cdot a & 3 \cdot b \end{bmatrix}^{-1} = \begin{bmatrix} \frac{3}{a} & -\frac{1}{a} \\ -\frac{2}{b} & \frac{1}{b} \end{bmatrix}$$

$$A^{-1} = \begin{bmatrix} 0,1 & -0,14 & 0,12 & -0,079 & -0,025 & 0,075 & -0,052 & 0,15 \\ -0,043 & -0,063 & 0,038 & -0,052 & 0,058 & -0,018 & 0,052 & 0,09 \\ -0,0081 & -0,03 & -0,076 & 0,003 & -0,016 & 0,077 & 0,021 & -0,0046 \\ -0,01 & 0,025 & -0,18 & -0,035 & 0,21 & 0,032 & -0,042 & 0,016 \\ 0,013 & 0,0029 & -0,032 & 0,018 & 0,074 & -0,016 & -0,024 & -0,11 \\ -0,11 & 0,1 & -0,15 & 0,031 & 0,055 & 0,024 & -0,02 & -0,045 \dots \\ 0,065 & 0,076 & 0,17 & 0,05 & -0,29 & -0,14 & 0,14 & -0,0076 \\ -0,044 & 0,034 & -0,066 & 0,1 & 0,1 & -0,077 & -0,049 & -0,046 \\ -0,028 & -0,0001 & -0,094 & 0,033 & 0,16 & 0,12 & -0,068 & -0,064 \\ 0,063 & 0,073 & 0,25 & -0,046 & -0,3 & -0,052 & 0,079 & 0,034 \end{bmatrix}$$

Der **Minor** ist der Wert der Unterdeterminante, die sich durch Streichung der  $i$ -ten Zeile und der  $j$ -ten Spalte ergibt. Die Eingabe erfolgt als `minor()` oder mit  $M_{ij}$  aus dem Matrixpanel der Seitenleiste. Die Funktion `vminor()` liefert die entsprechende Matrix, leider ist das an der Operatordarstellung nicht zu unterscheiden.

$$\text{minor}() \quad M_{12} \left( \begin{bmatrix} 5 & 11 & 17 \\ 11 & 25 & 39 \\ 17 & 39 & 61 \end{bmatrix} \right) = 8$$

$$\text{vminor}() \quad M_{12} \left( \begin{bmatrix} 5 & 11 & 17 \\ 11 & 25 & 39 \\ 17 & 39 & 61 \end{bmatrix} \right) = \begin{bmatrix} 11 & 39 \\ 17 & 61 \end{bmatrix} \quad \left| \begin{bmatrix} 11 & 39 \\ 17 & 61 \end{bmatrix} \right| = 8$$

Das **algebraische Komplement** ist gleich dem Minor, multipliziert mit  $(-1)^{i+j}$ . Eingabe mit `alg()` oder **Seitenleiste** > **Matrizen** > **A<sub>i,j</sub>**.

$$\text{alg()} \quad A_{12} \left( \begin{pmatrix} 5 & 11 & 17 \\ 11 & 25 & 39 \\ 17 & 39 & 61 \end{pmatrix} \right) = -8$$

**Spur** (*trace*) einer Matrix (Summe der Diagonalelemente)

$$\text{tr} \left( \begin{pmatrix} 5 & 11 & 17 \\ 11 & 25 & 39 \\ 17 & 39 & 61 \end{pmatrix} \right) = 91$$

**Skalarprodukt** von Vektoren

$$\begin{aligned} \mathbf{a} &:= \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} & \mathbf{b} &:= \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} & \mathbf{a} \cdot \mathbf{b} &= a_1 \cdot b_1 + a_2 \cdot b_2 + a_3 \cdot b_3 \\ & & & & \mathbf{a} \cdot \mathbf{a} &= a_1^2 + a_2^2 + a_3^2 \\ & & & & \mathbf{a}^2 &= a_1^2 + a_2^2 + a_3^2 \end{aligned}$$

Das Skalarprodukt als Matrixprodukt geschrieben ergibt eine  $1 \times 1$ -Matrix.

$$\mathbf{a}^T \cdot \mathbf{b} = [a_1 \cdot b_1 + a_2 \cdot b_2 + a_3 \cdot b_3]$$

**Kreuzprodukt** zweier dreidimensionaler Vektoren

$$\mathbf{a} \times \mathbf{b} = \begin{pmatrix} a_2 \cdot b_3 - a_3 \cdot b_2 \\ a_3 \cdot b_1 - a_1 \cdot b_3 \\ a_1 \cdot b_2 - a_2 \cdot b_1 \end{pmatrix}$$

**Dyadisches oder Tensorprodukt**

$$\mathbf{a} \cdot \mathbf{b}^T = \begin{bmatrix} a_1 \cdot b_1 & a_1 \cdot b_2 & a_1 \cdot b_3 \\ a_2 \cdot b_1 & a_2 \cdot b_2 & a_2 \cdot b_3 \\ a_3 \cdot b_1 & a_3 \cdot b_2 & a_3 \cdot b_3 \end{bmatrix}$$

**Matrix- und Vektornormen**

SMath kennt mehrere Normen, hier zeigen wir sie zusammen mit anwenderdefinierten Nachbauten (man kann nie misstrauisch genug sein).

$$\mathbf{A} := \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \quad \mathbf{b} := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Die Euklidische Norm `norme()` ist der quadratische Mittelwert aller Matrixelemente. Links die eingebaute Funktion, rechts ein Nachbau.

$$\begin{array}{lll}
 \text{norme}(A) = 9,54 & \text{norm}_2(M\#) := \sqrt{\text{tr}(M\#^T \cdot M\#)} & \text{norm}_2(A) = 9,54 \\
 \text{norme}(A^T) = 9,54 & & \text{norm}_2(A^T) = 9,54 \\
 \text{norme}(b) = 3,74 & & \text{norm}_2(b) = 3,74
 \end{array}$$

Die 1-Norm oder Spaltensummennorm  $\text{norm1}()$  ist gleich der maximalen Spaltensumme der Matrix.

$$\begin{array}{lll}
 \text{norm1}(A) = 12 & \text{norm}_1(M\#) := \left| \begin{array}{l} \text{for } j \in [1..\text{cols}(M\#)] \\ \text{rows}(M\#) \\ a\#_j := \sum_{i=1}^{\text{rows}(M\#)} |M\#_{ij}| \\ \max(a\#) \end{array} \right. & \begin{array}{l} \text{norm}_1(A) = 12 \\ \text{norm}_1(A^T) = 11 \\ \text{norm}_1(b) = 6 \end{array}
 \end{array}$$

Die  $\infty$ -Norm oder Zeilensummennorm  $\text{normi}()$  ist gleich der maximalen Zeilensumme der Matrix.

$$\begin{array}{lll}
 \text{normi}(A) = 11 & \text{norm}_\infty(M\#) := \left| \begin{array}{l} \text{for } i \in [1..\text{rows}(M\#)] \\ \text{cols}(M\#) \\ a\#_i := \sum_{j=1}^{\text{cols}(M\#)} |M\#_{ij}| \\ \max(a\#) \end{array} \right. & \begin{array}{l} \text{norm}_\infty(A) = 11 \\ \text{norm}_\infty(A^T) = 12 \\ \text{norm}_\infty(b) = 3 \end{array}
 \end{array}$$

## Eigen- und Singulärwerte

SMath Studio verfügt nicht über eingebaute Funktionen für die Eigen- und Singulärwertberechnung. Es sind aber verschiedene Plug-ins verfügbar, die numerische Funktionsbibliotheken einbinden. Darunter das Plugin *DotNumerics*, welches einige Funktionen der [DotNumeric-Bibliothek](#)<sup>2</sup> einbindet.

**Eigenwerte und Eigenvektoren** berechnen:

$$\begin{array}{l}
 M := \begin{bmatrix} 100 & 50 \\ 50 & 0 \end{bmatrix} \\
 \lambda := \text{dn\_LinAlgEigenvalues}(M) \quad \lambda = \begin{bmatrix} 121 \\ -20,7 \end{bmatrix} \\
 \Phi := \text{dn\_LinAlgEigenvectors}(M) \quad \Phi = \begin{bmatrix} 0,924 & -0,383 \\ 0,383 & 0,924 \end{bmatrix} \quad |\Phi| = 1
 \end{array}$$

Die Eigenvektoren werden als Spalten der orthogonalen Spektralmatrix  $\Phi$  geliefert. Mit dieser kann man die gegebene Matrix auf Diagonalform transformieren. Auf der Diagonale stehen dann die Eigenwerte:

$$\Phi^T \cdot M \cdot \Phi = \begin{bmatrix} 121 & -1,89 \cdot 10^{-14} \\ -1,23 \cdot 10^{-14} & -20,7 \end{bmatrix}$$

<sup>2</sup><http://www.dotnumerics.com/>



Aufgrund von Rundungsfehlern sind die Nichtdiagonalelemente nicht exakt Null.

Ebenso kann aus den Eigenwerten und der Spektralmatrix wieder die ursprüngliche Matrix erzeugt werden (inverse Transformation mit  $\underline{\Phi}^{-1} = \underline{\Phi}^T$ )

$$\underline{\Phi} \cdot \text{diag}(\lambda) \cdot \underline{\Phi}^T = \begin{bmatrix} 100 & 50,0 & \\ 50,0 & 5,63 \cdot 10^{-15} & -15 \end{bmatrix}$$

**Singulärwertzerlegung.** Die Funktion `dn_LinAlgSVD()` liefert die einen Vektor aus den Komponenten der Singulärwertzerlegung.

$$X := \begin{bmatrix} 1 & 2 & 3 \\ 3,5 & 0,5 & 8 \\ -1 & 2 & -3 \\ 4 & 9 & 7 \end{bmatrix} \quad SUV := \text{dn\_LinAlgSVD}(X)$$

Der erste Eintrag ist die Matrix mit den Singulärwerten:

$$S := SUV_1 = \begin{bmatrix} 14,5 & 0 & 0 \\ 0 & 6,39 & 0 \\ 0 & 0 & 0,453 \\ 0 & 0 & 0 \end{bmatrix}$$

Die beiden folgenden Einträge sind die Matrizen  $\underline{U}$  und  $\underline{V}^T$ , deren Spalten die Rechts- bzw. die Linkssingulärvektoren enthalten:

$$U := SUV_2 = \begin{bmatrix} -0,257 & 0,00816 & 0,959 & -0,12 \\ -0,526 & 0,672 & -0,206 & -0,478 \\ 0,108 & -0,532 & -0,0708 & -0,837 \\ -0,803 & -0,515 & -0,181 & 0,239 \end{bmatrix} \quad |U| = 1$$

$$V := SUV_3^T = \begin{bmatrix} -0,374 & 0,131 & -0,918 \\ -0,538 & -0,837 & 0,1 \\ -0,755 & 0,532 & 0,384 \end{bmatrix} \quad |V| = 1$$

Die so zerlegte Matrix kann nun wieder als Produkt der Faktoren dargestellt werden:

$$U \cdot S \cdot V^T = \begin{bmatrix} 1 & 2 & 3 \\ 3,5 & 0,5 & 8 \\ -1 & 2 & -3 \\ 4 & 9 & 7 \end{bmatrix}$$

Die Singulärwerte von  $\underline{X}$  entsprechen den Quadratwurzeln der Eigenwerte von  $\underline{X}^T \underline{X}$ :

$$\lambda := \text{dn\_LinAlgEigenvalues}(\underline{X}^T \cdot \underline{X}) = \begin{bmatrix} 210 \\ 0,205 \\ 40,8 \end{bmatrix}$$

$$\sqrt{\lambda} = \begin{bmatrix} 14,5 \\ 0,453 \\ 6,39 \end{bmatrix}$$

### 6.8.6. Manipulation von Matrizen

Abmessungen von Matrizen ermitteln (**Zeilenzahl**, **Spaltenzahl**, **Elementanzahl**): `cols( $\underline{M}$ )` ist die Spaltenzahl, `rows( $\underline{M}$ )` ist die Zeilenzahl und `length( $\underline{M}$ )` ist die Elementanzahl einer Matrix  $\underline{M}$ .

$$A := \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \quad \text{cols}(A) = 2 \quad \text{rows}(A) = 3 \quad \text{length}(A) = 6$$

### Zugriff auf Zeilen und Spalten

$\text{col}(M; c)$  ist die  $c$ -te Spalte,  $\text{row}(M; r)$  ist die  $r$ -te Zeile einer Matrix  $M$ .

$$\text{col}(A; 1) = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} \quad \text{row}(A; 3) = [5 \ 6]$$

**Elementindizes** werden mit der Funktion `el()` oder mit dem Tastaturkürzel `[]` erzeugt. Dabei entsteht zunächst nur ein Index, ein zweiter kann durch Eingabe eines Argumenttrennzeichens (Semikolon) hinzugefügt werden.

$$A_{1 \ 2} = 2$$

Matrizelemente können einzeln definiert/überschrieben werden:

$$A_{1 \ 1} := 10 \quad A = \begin{bmatrix} 10 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

Überschreiten die Indexwerte bei der Zuweisung die Größe der Matrix, wird diese dynamisch erweitert.

$$A_{3 \ 4} := 1 \quad A = \begin{bmatrix} 10 & 2 & 0 & 0 \\ 3 & 4 & 0 & 0 \\ 5 & 6 & 0 & 1 \end{bmatrix}$$

Zugriff auf **Teilmatrizen**: Die Funktion `submatrix(M; r1; r2; c1; c2)` extrahiert die Teilmatrix von Zeile  $r_1$  bis  $r_2$  und Spalte  $c_1$  bis  $c_2$ :

$$\text{submatrix} \left( \begin{bmatrix} 5 & 11 & 17 \\ 11 & 25 & 39 \\ 17 & 39 & 61 \end{bmatrix}; 1; 2; 2; 3 \right) = \begin{bmatrix} 11 & 17 \\ 25 & 39 \end{bmatrix}$$

Alternativ können **Indexvektoren** verwendet werden. Das heißt, dass man bei der Auswertung für die Indizes von Vektoren und Matrizen selbst wieder Vektoren einsetzen kann.

$$\text{submatrix} \left( \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}; 1; 2; 2; 3 \right) = \begin{bmatrix} b & c \\ e & f \end{bmatrix}$$

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}_{[1..2][2..3]} = \begin{bmatrix} b & c \\ e & f \end{bmatrix} [1..2] = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad [2..3] = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

Die Erzeugung der Indexvektoren mit der Bereichsfunktion `range()` ist aufgrund der Operatordarstellung besonders gut lesbar.

Die Einträge in den Indexvektoren müssen nicht zusammenhängend oder aufsteigend sein:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{matrix} [1\ 3] \\ [1\ 3] \end{matrix} = \begin{bmatrix} a & c \\ g & i \end{bmatrix}$$

Verwendet man einen Indexvektor in umgekehrter Reihenfolge, kann man die Matrix horizontal oder vertikal spiegeln:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{matrix} [3..1] \\ [1..3] \end{matrix} = \begin{bmatrix} g & h & i \\ d & e & f \\ a & b & c \end{bmatrix} \quad \text{---}$$

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{matrix} [1..3] \\ [3..1] \end{matrix} = \begin{bmatrix} c & b & a \\ f & e & d \\ i & h & g \end{bmatrix} \quad \text{|}$$

Kombiniert man skalare und vektorielle Indizes, so kann man Zeilen oder Spalten herausgreifen.

$$M := \text{Random}_N(3; 4; 0; 10) = \begin{bmatrix} 0 & 6 & 6 & 9 \\ 9 & 5 & 6 & 0 \\ 1 & 9 & 9 & 4 \end{bmatrix}$$

$$\text{row}(M; 2) = [9\ 5\ 6\ 0] \quad M_{2[1..cols(M)]} = [9\ 5\ 6\ 0]$$

$$\text{col}(M; 3) = \begin{bmatrix} 6 \\ 6 \\ 9 \end{bmatrix} \quad M_{[1..rows(M)]3} = \begin{bmatrix} 6 \\ 6 \\ 9 \end{bmatrix}$$

**Implizite Schleifen** entstehen, wenn man auf der linken Seite von Zuweisungen vektorwertige Variable als Index verwendet. Dann wird die Zuweisung für alle Werte der Variable ausgeführt. Diese Variable wird daher auch Bereichsvariable (*range variable*, *ranged index*) genannt.

Im folgenden Beispiel wird eine ganze Zeile Null gesetzt:

$$M := \text{Random}_N(3; 4; 0; 10) = \begin{bmatrix} 2 & 6 & 3 & 8 \\ 6 & 5 & 5 & 7 \\ 6 & 1 & 10 & 7 \end{bmatrix}$$

$$M_{2[1..4]} := 0 \quad M = \begin{bmatrix} 2 & 6 & 3 & 8 \\ 0 & 0 & 0 & 0 \\ 6 & 1 & 10 & 7 \end{bmatrix}$$

Implizite Schleifen können auch sehr elegant zur Erzeugung von Matrizen und Vektoren verwendet werden:

$$\text{Clear}(M; N) = 1 \quad j := [1..3] \quad k := [1..3]$$

$$M_{j\ k} := j - k \quad N_{j\ k} := (-1)^{j+k}$$

$$M = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} \quad N = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}$$

**Lineare Adressierung:** Matrixelemente können auch über einen einzelnen Index angesprochen werden, der zeilenweise von 1 bis zur Länge der Matrix läuft. Das funktioniert sowohl bei der Anzeige als auch bei der Zuweisung.

$$A_{10} = 6$$

$$A_{10} := "" \quad A = \begin{bmatrix} 10 & 2 & 0 & 0 \\ 3 & 4 & 0 & 0 \\ 5 & "" & 0 & 1 \end{bmatrix}$$

Auch die lineare Adressierung kann mit Indexvektoren geschehen:

$$M = \begin{bmatrix} 0 & 6 & 6 & 9 \\ 9 & 5 & 6 & 0 \\ 1 & 9 & 9 & 4 \end{bmatrix} \quad M_{[5..8]} = \begin{bmatrix} 9 \\ 5 \\ 6 \\ 0 \end{bmatrix}$$

$$M_{[5..8]} := 0 \quad M = \begin{bmatrix} 0 & 6 & 6 & 9 \\ 0 & 0 & 0 & 0 \\ 1 & 9 & 9 & 4 \end{bmatrix}$$

**Verbinden von Matrizen** passender Abmessungen. Nebeneinanderstellen erfolgt mit der Funktion `augment()` (wörtlich: Erweitern):

$$\text{augment} \left( \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}; \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} \right) = \begin{bmatrix} 1 & 2 & 1 \\ 3 & 4 & 3 \\ 5 & 6 & 5 \end{bmatrix}$$

Skalare Argumente werden passend erweitert:

$$\text{augment} ([1..5]; "x"; 10) = \begin{bmatrix} 1 & "x" & 10 \\ 2 & "x" & 10 \\ 3 & "x" & 10 \\ 4 & "x" & 10 \\ 5 & "x" & 10 \end{bmatrix}$$

Hier wird die Funktion benutzt, um in einer expliziten Schleife eine zunächst leere zwei-zeilige Matrix um Spalten zu erweitern.

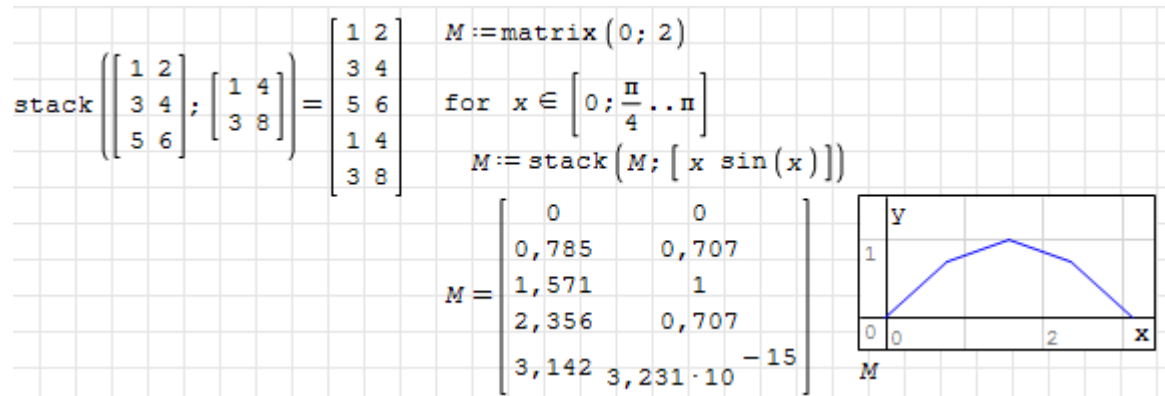
$$M := \text{matrix}(2; 0)$$

$$\text{for } x \in \left[ 0; \frac{\pi}{4} .. \pi \right]$$

$$M := \text{augment} \left( M; [x \sin(x)]^T \right)$$

$$M = \begin{bmatrix} 0 & 0,785 & 1,571 & 2,356 & 3,142 \\ 0 & 0,707 & 1 & 0,707 & 3,231 \cdot 10^{-15} \end{bmatrix}$$

Übereinanderstellen leistet die Funktion `stack()` (Stapeln). Auch hier ein Beispiel für die Verwendung beim Aufbau einer zunächst leeren zwispaltigen Matrix, die in der Schleife um Zeilen erweitert wird:



Hier haben wir die zweispaltige Matrix als Linienzug dargestellt.

### Suchen und Sortieren

Die Funktion `findrows(Matrix; Ausdruck; i)` greift aus einer Matrix alle Zeilen heraus, deren Wert in Spalte `i` gleich dem `Ausdruck` ist. Wird kein passender Eintrag gefunden, liefert die Funktion den Wert Null.

$$\text{findrows} \left( \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}; 4; 2 \right) = \begin{bmatrix} 3 & 4 \\ 3 & 4 \end{bmatrix}$$

$$\text{findrows} \left( \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}; 5; 1 \right) = [5 \ 6]$$

$$\text{findrows} \left( \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}; 7; 1 \right) = 0$$

Mit `csort(Matrix; i)` werden die Zeilen einer Matrix anhand der `i`-ten Spalte sortiert:

$$\text{csort} \left( \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 1 & 7 \\ 3 & 8 \\ 5 & 6 \end{bmatrix}; 1 \right) = \begin{bmatrix} 1 & 2 \\ 1 & 7 \\ 3 & 4 \\ 3 & 8 \\ 5 & 6 \end{bmatrix}$$

Entsprechend sortiert `rsort(Matrix; i)` die Spalten einer Matrix anhand der `i`-ten Zeile:

$$\text{rsort} \left( \begin{bmatrix} 1 & 3 & 1 & 3 & 5 \\ 2 & 4 & 7 & 8 & 6 \end{bmatrix}; 1 \right) = \begin{bmatrix} 1 & 1 & 3 & 3 & 5 \\ 2 & 7 & 4 & 8 & 6 \end{bmatrix}$$

Die Funktion `sort()` sortiert die Elemente einer Spaltenmatrix (eines Vektors) der Größe nach.

$$\text{sort} \left( \begin{bmatrix} 1 \\ 3 \\ 1 \\ 3 \\ 5 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 1 \\ 3 \\ 3 \\ 5 \end{bmatrix}$$

`reverse()` kehrt die Reihenfolge der Zeilen in einer Matrix um:

$$\text{reverse} \left( \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \right) = \begin{bmatrix} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{bmatrix}$$

Man kann dies benutzen, um absteigend zu sortieren:

$$\text{reverse} \left( \text{csort} \left( \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 1 & 7 \\ 3 & 8 \\ 5 & 6 \end{bmatrix}; 2 \right) \right) = \begin{bmatrix} 3 & 8 \\ 1 & 7 \\ 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{bmatrix}$$

Maximum und Minimum einer Matrix bestimmt man mit folgenden Funktionen:

$$\text{max} \left( \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \right) = 6 \quad \text{min} \left( \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \right) = 1$$

Das Plug-in *Custom Functions* enthält erweiterte Versionen der Maximum- und Minimum-Funktionen, die auch mit mehreren Argumenten unterschiedlicher Formen (Listen, Matrizen, Zahlen) zurechtkommen.

$$\text{Min} \left( 8; \begin{bmatrix} \pi \\ 7 \\ 5 \\ 1 & 25 & 44 \\ -10 \\ 3 \end{bmatrix}; \frac{14}{6}; \begin{bmatrix} \sqrt{2} & 12,7 \\ \pi & e^4 \\ 2 & -\frac{6}{7} \\ 5 & -\frac{6}{7} \\ -4 & -\frac{6}{7} \\ 7 \end{bmatrix} \right) = -10$$

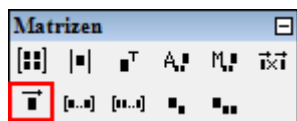
$$\text{Max} \left( 8; \begin{bmatrix} \pi \\ 7 \\ 5 \\ 1 & 25 & 44 \\ -10 \\ 3 \end{bmatrix}; \frac{14}{6}; \begin{bmatrix} \sqrt{2} & 12,7 \\ \pi & e^4 \\ 2 & -\frac{6}{7} \\ 5 & -\frac{6}{7} \\ -4 & -\frac{6}{7} \\ 7 \end{bmatrix} \right) = 54,6$$

### 6.8.7. Funktionen auf Vektor- oder Matrixelemente anwenden

Mitunter muss man beliebige Funktionen auf alle Elemente einer Matrix oder eines Vektors anwenden. Dafür gibt es mehrere Optionen:

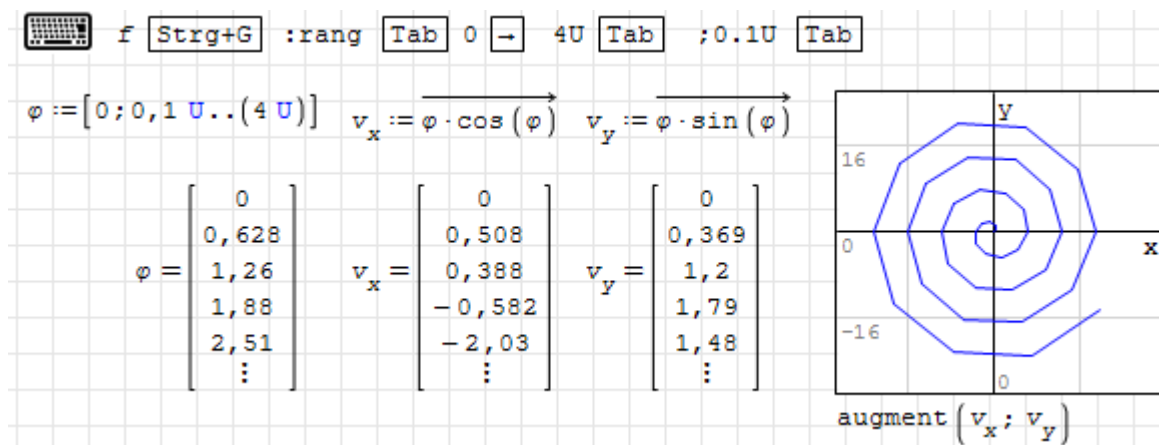
- Die Funktion `vectorize()`, die einen gegebenen Ausdruck auf alle Elemente der darin vorkommenden Variablen anwendet. Das entspricht dem Rechnen mit Tabellenspalten in Programmen wie Excel.
- Explizite Programmschleifen.
- Implizite Schleifen mit Bereichsvariablen.

**Vektorisierung von Ausdrücken:** Die Funktion `vectorize()` kann über die Tastatur mit der dynamischen Hilfe oder über das Matrix-Panel der Seitenleiste eingegeben werden.



$$\begin{aligned}
 x &:= [0..5] \\
 x &= \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} \xrightarrow{x^2} \begin{bmatrix} 0 \\ 1 \\ 4 \\ 9 \\ 16 \\ 25 \end{bmatrix} \xrightarrow{x^2 + x} \begin{bmatrix} 0 \\ 2 \\ 6 \\ 12 \\ 20 \\ 30 \end{bmatrix} \xrightarrow{\sin(x)} \begin{bmatrix} 0 \\ 0,841 \\ 0,909 \\ 0,141 \\ -0,757 \\ -0,959 \end{bmatrix}
 \end{aligned}$$

Hier ein Anwendungsbeispiel: Darstellung einer Spirale. „U“ ist eine Winkeleinheit, die Umdrehung (360°).



**Explizite Schleifen:** In Schleifen, deren Zähler von 1 bis zur Länge eines Vektors läuft, kann man elementweise beliebige Berechnungen anstellen. Das Beispiel oben sieht dann so aus:

$$\begin{aligned}
 \varphi &:= [0; 0,1 U..(4 U)] \\
 \text{for } i \in [1..length(\varphi)] \\
 \left. \begin{aligned} v_{x_i} &:= \varphi_i \cdot \cos(\varphi_i) \\ v_{y_i} &:= \varphi_i \cdot \sin(\varphi_i) \end{aligned} \right\} \begin{aligned} v_x &= \begin{bmatrix} 0 \\ 0,508 \\ 0,388 \\ -0,582 \\ -2,03 \\ \vdots \end{bmatrix} & v_y &= \begin{bmatrix} 0 \\ 0,369 \\ 1,2 \\ 1,79 \\ 1,48 \\ \vdots \end{bmatrix} \end{aligned}
 \end{aligned}$$

Solche Schleifen werden ausführlich im Abschnitt 10.2.2 behandelt.

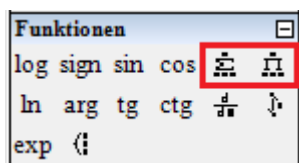
**Implizite Schleifen:** Werden bei Zuweisungen auf Vektor- oder Matrixelemente vektorwertige Indizes verwendet, dann wird die Zuweisung für alle Indexwerte nacheinander ausgeführt, ohne dass man diese Schleife explizit schreiben muss:

$$\begin{aligned} \varphi &:= [0; 0,1 \cup \dots (4 \cup)] \\ i &:= [1..length(\varphi)] \quad v_{x_i} := \varphi_i \cdot \cos(\varphi_i) \quad v_{y_i} := \varphi_i \cdot \sin(\varphi_i) \\ \varphi &= \begin{bmatrix} 0 \\ 0,1 \\ 0,2 \\ 0,3 \\ 0,4 \\ \vdots \end{bmatrix} \cup i = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ \vdots \end{bmatrix} \quad v_x = \begin{bmatrix} 0 \\ 0,508 \\ 0,388 \\ -0,582 \\ -2,03 \\ \vdots \end{bmatrix} \quad v_y = \begin{bmatrix} 0 \\ 0,369 \\ 1,2 \\ 1,79 \\ 1,48 \\ \vdots \end{bmatrix} \end{aligned}$$

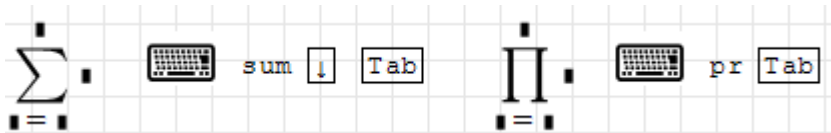
### 6.8.8. Summen und Produkte

Summen- und Produktzeichen können auf drei Wegen in das Rechenblatt eingefügt werden:

- über das Funktionen-Panel der Seitenleiste



- über das Menü **Einfügen** > **Funktionen** > **Alle** > **sum(4)** oder **product**
- über die Tastatur



Die Summe aller Zahlen von 1 bis 100 berechnet sich so:

$$\sum_{i=1}^{100} i = 5050 \quad \text{sum} \downarrow \text{Tab} \quad i \rightarrow i \rightarrow 1 \rightarrow 100 =$$

Größen, die man in einer Summe mit ihrem Index ansprechen will, müssen als Vektor oder Matrix (dann wird zeilenweise lineare Adressierung verwendet) definiert werden. Achten Sie darauf, Elementindizes (statt Textindizes) zu verwenden. Man erzeugt sie mit der Funktion `e1()` oder durch das Tastaturzeichen `[` (eckige Klammer auf).

Im folgenden Beispiel wird der Gesamtschwerpunkt dreier Flächen  $A_i$  mit den Teilschwerpunkten  $x_i$  berechnet.

$$\begin{aligned} & \text{A}[1] \text{ Leer } 20\text{cm} \text{ Tab} \\ A_1 &:= 20 \text{ cm}^2 & A_2 &:= 30 \text{ cm}^2 & A_3 &:= 20 \text{ cm}^2 \\ x_1 &:= 2 \text{ cm} & x_2 &:= 4 \text{ cm} & x_3 &:= -2 \text{ cm} \\ A &= \begin{bmatrix} 20 \\ 30 \\ 20 \end{bmatrix} \text{ cm}^2 & x &= \begin{bmatrix} 20 \\ 40 \\ -20 \end{bmatrix} \text{ mm} & n &:= \text{length}(A) \end{aligned}$$



$$x_s := \frac{\sum_{j=1}^n (A_j \cdot x_j)}{\sum A} = 17,1 \text{ mm}$$

Im Nenner steht eine verkürzte Form des Summenzeichens, diese summiert über alle Elemente eines Vektors und kann über den Dialog „Funktion einfügen“ oder die Tastatur eingegeben werden, dabei ist `sum(1)` auszuwählen.

Die Summe im Zähler kann auch als Skalarprodukt geschrieben werden, das gibt eine kompaktere Schreibweise, die aber weniger verständlich ist und daher vermieden werden sollte:

$$x_s := \frac{A \cdot x}{\sum A} = 17,1 \text{ mm}$$

Nun noch ein Beispiel für ein Produkt. Die Fakultät einer Zahl ist gleich dem Produkt aller natürlichen Zahlen von 1 bis zu dieser Zahl:

$$\prod_{i=1}^{10} i = 3628800 \quad 10! = 3628800$$

### 6.8.9. Vektorrechnung in kartesischen Koordinaten

Vektoren werden durch Spaltenmatrizen ihrer kartesischen Koordinaten dargestellt. Entsprechend werden Tensoren als quadratische Koordinatenmatrizen dargestellt.

Vektorrechnung mit Zahlenwerten und numerischer Auswertung funktioniert, symbolische Rechnungen stoßen aber auf Probleme:

- Es ist nicht möglich, symbolische Variablen als Vektoren zu deklarieren.
- Symbolische Vereinfachungen erfolgen nach den Regeln für skalare Variablen. **Das kann zu Fehlern führen, die unbemerkt bleiben**, ein Beispiel finden Sie beim Skalarprodukt, Seite [139](#).
- Manche Funktionen (Determinante, Kreuzprodukt) funktionieren symbolisch nur eingeschränkt, siehe Seite [140](#).

Zudem fehlen die üblichen Markierungen für Vektorsymbole (Fettschrift, Unterstreichung, Pfeil).

#### Basis, Vektorkomponenten und Vektorkoordinaten

Basisvektoren im kartesischen Koordinatensystem:

$$e_x := \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad e_y := \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad e_z := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Man kann **Basisvektoren** auch als Funktion definieren. Der  $i$ -te Basisvektor ist die  $i$ -te Spalte in einer Einheitsmatrix passender Dimension (hier 3):

$$e(k) := \text{col}(\text{identity}(3), k) \quad e(1) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad e(2) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad e(3) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Ein physikalischer Vektor kann dargestellt werden als Summe aus den Produkten einheitenbehafteter Koordinaten und den zugehörigen Basisvektoren.  $F_x, F_y, F_z$  seien beispielsweise die **Koordinaten** (oder skalare Komponenten) eines Kraftvektors  $\mathbf{F}$ :

$$F_x := 1 \text{ N} \quad F_y := 3 \text{ N} \quad F_z := 2 \text{ N}$$

Dann sind die Produkte dieser Koordinaten mit den zugehörigen Basisvektoren die sogenannten (vektoriellen) **Komponenten** des Kraftvektors in Richtung der Koordinatenachsen. Merke: *Koordinaten* sind (gegebenenfalls einheitenbehaftete) Zahlen, *Komponenten* sind Vektoren, haben also eine Richtung.

$$F_x \cdot e_x = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \text{ N} \quad F_y \cdot e_y = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} \text{ N} \quad F_z \cdot e_z = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} \text{ N}$$

Die Summe der Komponenten ergibt den kompletten Kraftvektor:

$$\mathbf{F} := F_x \cdot e_x + F_y \cdot e_y + F_z \cdot e_z \quad \mathbf{F} = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} \text{ N}$$

Man kann auch direkt die Koordinaten in eine Spaltenmatrix schreiben:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} \text{ N}$$

Mit SMath kann man die Koordinaten eines Vektors mit der Indexfunktion ansprechen:

$$F_1 = 1 \text{ N} \quad F_2 = 3 \text{ N} \quad F_3 = 2 \text{ N}$$

## Skalarprodukt

Das Skalarprodukt zweier Vektoren  $\mathbf{a}$  und  $\mathbf{b}$  ist eine Zahl, die gleich dem Produkt aus dem Betrag  $|\mathbf{a}|$  des einen, dem Betrag  $|\mathbf{b}|$  des anderen und dem Kosinus des eingeschlossenen Winkels ist. In kartesischen Koordinaten ist dies gleich der Summe über die Produkte der Koordinaten beider Vektoren:

$$\mathbf{a} := \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad \mathbf{b} := \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \quad \mathbf{a} \cdot \mathbf{b} = a_x \cdot b_x + a_y \cdot b_y + a_z \cdot b_z$$

Die Basisvektoren kartesischer Koordinatensysteme haben die Länge 1 und stehen senkrecht aufeinander. Daher hat das Skalarprodukt eines Basisvektors mit sich selbst den

Wert 1 und mit einem anderen Basisvektor den Wert 0. Solch eine Basis nennt man orthonormiert (rechtwinklig und Länge 1).

$$j := [1..3] \quad k := [1..3]$$

$$M_{j,k} := e(j) \cdot e(k) \quad M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Die Multiplikation des Skalarprodukts zweier Vektoren mit einem dritten Vektor liefert wieder einen Vektor. Dabei kommt es entscheidend auf die Klammersetzung an.

Die Vektoren  $\mathbf{a}$ ,  $\mathbf{b}$  und  $\mathbf{c}$  sind hier noch nicht definiert (SMath weiß überhaupt noch nicht, dass die Variablen mal Vektoren werden sollen). Symbolische Auswertung liefert:

$$(\mathbf{a} \cdot \mathbf{b}) \cdot \mathbf{c} = \mathbf{a} \cdot \mathbf{b} \cdot \mathbf{c} \quad \mathbf{a} \cdot (\mathbf{b} \cdot \mathbf{c}) = \mathbf{a} \cdot \mathbf{b} \cdot \mathbf{c} \quad \mathbf{b} \cdot (\mathbf{a} \cdot \mathbf{c}) = \mathbf{b} \cdot \mathbf{a} \cdot \mathbf{c}$$

Wie man sieht, werden die Klammern ignoriert, was für Skalare ja auch in Ordnung ist. Auch wenn die Vektoren bekannt sind, führt symbolische Auswertung auf falsche Ergebnisse, denn es wird erst symbolisch vereinfacht, dann eingesetzt:

$$\mathbf{a} := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{b} := \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \quad \mathbf{c} := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad \begin{array}{l} \mathbf{a} \cdot \mathbf{b} = 6 \\ \mathbf{a} \cdot \mathbf{c} = 6 \\ \mathbf{b} \cdot \mathbf{c} = 10 \end{array}$$

Auswertung symbolisch

$$(\mathbf{a} \cdot \mathbf{b}) \cdot \mathbf{c} = \begin{bmatrix} 6 \\ 12 \\ 18 \end{bmatrix} \quad \mathbf{a} \cdot (\mathbf{b} \cdot \mathbf{c}) = \begin{bmatrix} 6 \\ 12 \\ 18 \end{bmatrix} \quad \mathbf{b} \cdot (\mathbf{a} \cdot \mathbf{c}) = \begin{bmatrix} 6 \\ 12 \\ 18 \end{bmatrix} \quad \text{falsch!}$$

Bei numerischer Auswertung wird direkt eingesetzt und berechnet, dann stimmen die Ergebnisse:

$$(\mathbf{a} \cdot \mathbf{b}) \cdot \mathbf{c} = \begin{bmatrix} 6 \\ 12 \\ 18 \end{bmatrix} \quad \mathbf{a} \cdot (\mathbf{b} \cdot \mathbf{c}) = \begin{bmatrix} 10 \\ 10 \\ 10 \end{bmatrix} \quad \mathbf{b} \cdot (\mathbf{a} \cdot \mathbf{c}) = \begin{bmatrix} 18 \\ 12 \\ 6 \end{bmatrix}$$

## Betrag



Der Betrag eines Vektors ist gleich der Wurzel aus dem Skalarprodukt mit sich selbst. Die Funktion `norme()` ist die euklidische Vektornorm und liefert ebenfalls den Betrag eines Vektors.

$$\mathbf{F} = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} \text{ N} \quad \mathbf{F} \cdot \mathbf{F} = 14 \text{ N}^2 \quad \sqrt{\mathbf{F} \cdot \mathbf{F}} = 3,74 \text{ N} \quad \text{norme}(\mathbf{F}) = 3,74 \text{ N}$$

$$\sqrt{\mathbf{a} \cdot \mathbf{a}} = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (\text{symbolische Auswertung})$$

## Kreuzprodukt

Das Kreuzprodukt zweier Vektoren  $\mathbf{a}$  und  $\mathbf{b}$  ist gleich einem Vektor, der auf beiden Vektoren senkrecht steht und dessen Betrag gleich der von  $\mathbf{a}$  und  $\mathbf{b}$  aufgespannten Trapezfläche


ist. Diese Fläche wiederum ist gleich dem Produkt aus den Beträgen von  $\mathbf{a}$  und  $\mathbf{b}$  und dem Sinus des eingeschlossenen Winkels. Das Kreuzprodukt kann sehr einfach aus den kartesischen Koordinaten der beiden Vektoren berechnet werden. In SMATH steht dafür eine separate Funktion bereit, erreichbar über **Seitenleiste** > **Matrizen** >  oder **Strg**  **8** .

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} a_y \cdot b_z - a_z \cdot b_y \\ a_z \cdot b_x - a_x \cdot b_z \\ a_x \cdot b_y - a_y \cdot b_x \end{bmatrix}$$

Die Kreuzprodukte zweier Basisvektoren liefern jeweils den senkrecht auf beiden stehenden, also den dritten Basisvektor:

$$\mathbf{e}_x \times \mathbf{e}_y = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{e}_y \times \mathbf{e}_z = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{e}_z \times \mathbf{e}_x = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

SMATH hat bei symbolischer Auswertung unter Umständen Schwierigkeiten mit Faktoren, die nicht auf ersten Blick als Vektoren zu erkennen sind.

```
(λ · a) × b = 
lastError = "Vektorprodukt ist nur für Vektoren mit drei Komponenten zulässig."
(2 · a) × b = 
$$\begin{bmatrix} 2 \cdot (a_y \cdot b_z - a_z \cdot b_y) \\ 2 \cdot (a_z \cdot b_x - a_x \cdot b_z) \\ 2 \cdot (a_x \cdot b_y - a_y \cdot b_x) \end{bmatrix}$$

```

## Spatprodukt

Das Spatprodukt dreier Vektoren ist das Volumen des von diesen drei Vektoren aufgespannten Parallelepipeds (Spat).

$$\mathbf{c} := \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} \quad \mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = a_x \cdot (b_y \cdot c_z - b_z \cdot c_y) + a_y \cdot (b_z \cdot c_x - b_x \cdot c_z) + a_z \cdot (b_x \cdot c_y - b_y \cdot c_x)$$

(symbolische Auswertung)

## 6.9. Listen

Listen sind eindimensionale Felder, die in SMATH als Spalte mit einer geschweiften Klammer links angezeigt werden:

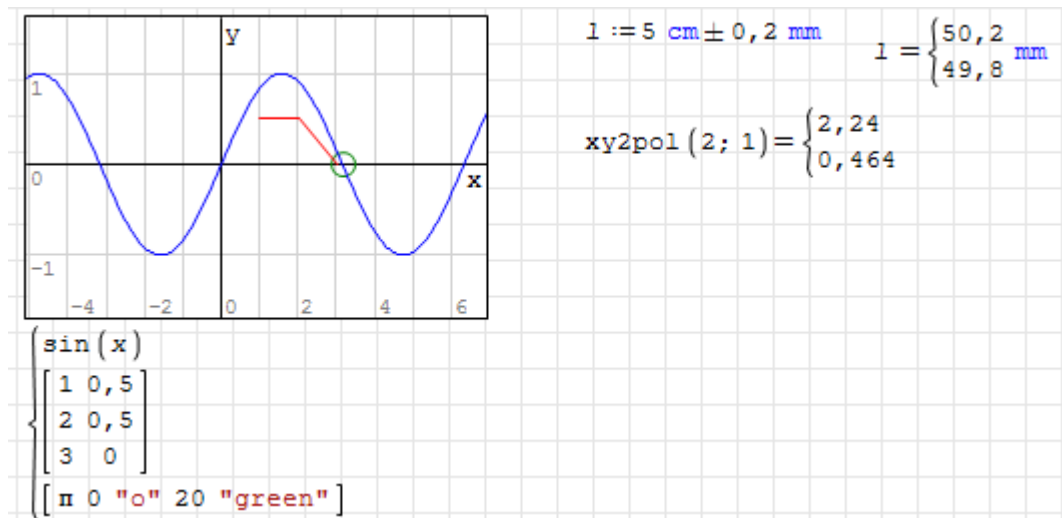
$$\{ \begin{matrix} \blacksquare \\ \blacksquare \end{matrix} \}$$

Listen werden in unterschiedlichen Zusammenhängen verwendet:

- Bei vielen Maxima-Funktionen treten Argumente oder Ergebnisse als Listen auf.

- Gruppierung mehrerer Objekte in einem Diagramm
- Gruppierung von Ergebnissen mit mehreren Werten (bei manchen Funktionen)

Im Englischen werden diese Objekte auch *multiple values objects* oder *systems* genannt.




### 6.9.1. Listenobjekte erzeugen

Listen erzeugt man mit **Seitenleiste** > **Funktionen** >  oder durch Eingabe von

sys 



Elemente können eingefügt werden, indem ein Semikolon vor oder hinter einem Ausdruck eingegeben wird. Leere Platzhalter werden durch  (Rücktaste) gelöscht.

Die Länge von Listen kann wie bei Matrizen durch Anklicken der Klammer und Ziehen an der rechten unteren Ecke vergrößert oder verkleinert werden.



Manche Funktionen oder Rechenoperationen liefern Listen als Ergebnisse, so zum Beispiel der Plus/Minus-Operator (**Seitenleiste** > **Arithmetik** >  $\pm$ ) und der Minus/Plus-Operator (**Einfügen** > **Operator** > **minus/plus**). Das Beispiel mit undefinierten Variablen und symbolischer Auswertung zeigt deutlich den Unterschied zwischen beiden Operatoren.

$$a \pm b = \begin{cases} a + b \\ a - b \end{cases} \quad a \mp b = \begin{cases} a - b \\ a + b \end{cases}$$

Auch die Koordinatenumrechnungsfunktionen liefern Listen:

$$\text{xy2pol}(-2; 0) = \begin{cases} 2 \\ 3,14 \end{cases}$$

$$\text{pol2xy}(1; 45^\circ) = \begin{cases} 0,707 - 4,37 \cdot 10^{-16} \cdot i \\ 0,707 + 4,44 \cdot 10^{-16} \cdot i \end{cases}$$

Die Listeneinträge können für Anzeige oder Definition mit Elementindizes angesprochen werden. Falls erforderlich, wird die Liste dabei verlängert. Im folgenden Beispiel wird

- die Liste  $s$  mit den Elementen 1 und 2,5 definiert,
- das Element Nummer 2 angezeigt,
- das Element Nummer 4 auf den Wert 3 gesetzt
- die ganze Liste angezeigt.

$$s := \begin{cases} 1 \\ 2,5 \end{cases} \quad s_2 = 2,5 \quad s_4 := 3 \quad s = \begin{cases} 1 \\ 2,5 \\ 0 \\ 3 \end{cases}$$

Wie eben gezeigt, funktioniert die dynamische Erweiterung durch Zuweisung auf ein Element, das noch nicht existiert, auch bei Listen. Die Variable muss allerdings schon als Liste existieren (rechts), sonst wird ein Vektor erzeugt (links).

$$L_2 := 2 \quad L = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \quad L := \{0\} \quad L_3 := 3 \quad L = \begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix}$$

Speziell für Listen gibt es die Anzeigeeoption, ob mehrfach auftretende Werte zusammengefasst werden. Die Einstellung erfolgt in **Extras** > **Einstellungen** > **Berechnung** > **Ergebnisse in Listen**. Das kann Einfluss auf die angezeigte Länge der Liste haben. Mit „Alle Einträge anzeigen“ werden alle erzeugten Listenelemente angezeigt:

Ergebnisse in Listen Alle Einträge anzeigen ▼

$$1 \pm 1 \pm 1 = \begin{cases} 3 \\ 1 \\ 1 \\ -1 \end{cases} \quad \begin{cases} 1 \\ 2 \\ 2 \\ 3 \end{cases} = \begin{cases} 1 \\ 2 \\ 2 \\ 3 \end{cases}$$

Mit „Wiederholte Werte nur einmal“ werden Wertwiederholungen unterdrückt, so dass die Liste keine gleichen Werte mehrfach enthält:

Ergebnisse in Listen Wiederholte Werte nur ei ▼

$$1 \pm 1 \pm 1 = \begin{cases} 3 \\ 1 \\ -1 \end{cases} \quad \begin{cases} 1 \\ 2 \\ 2 \\ 3 \end{cases} = \begin{cases} 1 \\ 2 \\ 3 \end{cases}$$

Standardmäßig werden im vorliegenden Handbuch „alle Einträge“ angezeigt.

## 6.9.2. Rechnen mit Listen

Rechenoperationen werden gleichzeitig auf alle Elemente einer Liste angewendet.

$$s := \begin{cases} 1 \\ 2, 5 \end{cases} \quad s + 2 = \begin{cases} 3 \\ 4, 5 \end{cases} \quad s^2 = \begin{cases} 1 \\ 6, 25 \end{cases} \quad \cos(s) = \begin{cases} 0, 54 \\ -0, 801 \end{cases}$$

Treten in einem Ausdruck zwei oder mehrere Listen auf, so wird der Ausdruck für alle möglichen Elementkombinationen ausgewertet:

$$s + s = \begin{cases} 2 \\ 3, 5 \\ 3, 5 \\ 5 \end{cases} \quad s \cdot s = \begin{cases} 1 \\ 2, 5 \\ 2, 5 \\ 6, 25 \end{cases} \quad s \cdot s = \begin{cases} 1 \\ 2, 5 \\ 9, 88 \end{cases} \quad s \cdot s + s = \begin{cases} 2 \\ 3, 5 \\ 2 \\ 10, 9 \\ 3, 5 \\ 5 \\ 3, 5 \\ 12, 4 \end{cases}$$

**Beispiel: Toleranzanalyse.** In welchen Grenzen schwankt das Volumen eines Quaders, wenn die Kantenlängen mit ihren Toleranzen bekannt sind? Zunächst werden die Abmessungen mit ihren Toleranzen definiert:

$$l := 10 \text{ cm} + \begin{cases} 0 \text{ mm} \\ 0, 5 \text{ mm} \end{cases} \quad b := 5 \text{ cm} \pm 0, 2 \text{ mm} \quad \text{absolute Toleranz}$$

$$h := 3 \text{ cm} \cdot \begin{cases} 0, 9 \\ 1, 1 \end{cases} \quad \text{relative Toleranz, } \pm 10\%$$

Die Listen enthalten nun die extremal möglichen Abmessungswerte:

$$l = \begin{cases} 100 \\ 100, 5 \end{cases} \text{ mm} \quad b = \begin{cases} 50, 2 \\ 49, 8 \end{cases} \text{ mm} \quad h = \begin{cases} 27 \\ 33 \end{cases} \text{ mm}$$

(Signifikante Stellen auf 4 gesetzt)

Das Volumen ist das Produkt der drei Kantenlängen. Berechnet wird das Volumen jeder möglichen Kombination von Einzelwerten. Da es hier um kleine Unterschiede zwischen den Zahlen geht, wird die Darstellung auf 3 Nachkommastellen umgestellt.

$$V := l \cdot b \cdot h$$

$$V = \begin{cases} 135, 540 \\ 136, 218 \\ 134, 460 \\ 135, 132 \\ 165, 660 \\ 166, 488 \\ 164, 340 \\ 165, 162 \end{cases} \text{ cm}^3$$

Format:  
3 Nachkommastellen

Nun sind noch die Extremwerte zu ermitteln. Die Standardfunktionen `min()` und `max()` sind dafür ungeeignet, denn sie werden auf Listen elementweise angewendet, was nutzlos ist. Das Plug-in *Custom Functions* stellt unter anderem die Funktionen `Min()` und `Max()` bereit, die mit beliebig strukturierten Kombinationen aus Matrizen und Listen zurechtkommen.

$$\min(V) = \begin{cases} 135,540 \\ 136,218 \\ 134,460 \\ 135,132 \\ 165,660 \\ 166,488 \\ 164,340 \\ 165,162 \end{cases} \text{ cm}^3$$

$\text{Min}(V) = 134,460 \text{ cm}^3$   
 $\text{Max}(V) = 166,488 \text{ cm}^3$   
 Format:  
 3 Nachkommastellen

### 6.9.3. Umwandlung Liste $\longleftrightarrow$ Vektor

Das Plug-in *Custom Functions* bietet unter anderem auch Umwandlungsfunktionen für Matrizen in Listen und umgekehrt.

**Liste zu Vektor:** Die Funktionen `sys2mat()` und `sys2mat.1()` wandeln entweder alle oder nur die äußerste Listenebene einer verschachtelten Struktur in Vektoren um:

$$\text{sys2mat} \left( \left( \left[ \begin{array}{c} 1 \left\{ \begin{array}{c} 10 \\ 20 \end{array} \right\} \\ 3 \quad 4 \\ 5 \\ 6 \end{array} \right] \right) \right) = \left[ \begin{array}{c} 1 \left[ \begin{array}{c} 10 \\ 20 \end{array} \right] \\ 3 \quad 4 \\ 5 \\ 6 \end{array} \right]$$

$$\text{sys2mat}_1 \left( \left( \left[ \begin{array}{c} 1 \left\{ \begin{array}{c} 10 \\ 20 \end{array} \right\} \\ 3 \quad 4 \\ 5 \\ 6 \end{array} \right] \right) \right) = \left[ \begin{array}{c} 1 \left\{ \begin{array}{c} 10 \\ 20 \end{array} \right\} \\ 3 \quad 4 \\ 5 \\ 6 \end{array} \right]$$

Beachten Sie, dass direkt verschachtelte Listen grundsätzlich intern aufgelöst werden:

$$\left\{ \begin{array}{c} 1 \\ 2 \\ 5 \\ 6 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 2 \\ 5 \\ 6 \end{array} \right\}$$

**Vektor zu Liste:** `mat2sys()` packt alle Elemente einer Struktur in eine einzige Liste. `mat2sys.1()` wandelt nur die äußere Vektorebene in eine Liste um.

$$\text{mat2sys} \left( \left( \left[ \begin{array}{c} 1 \left\{ \begin{array}{c} 10 \\ 20 \end{array} \right\} \\ 3 \quad 4 \\ 5 \\ 6 \end{array} \right] \right) \right) = \left[ \begin{array}{c} 1 \\ 10 \\ 20 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \right]$$

$$\text{mat2sys}_1 \left( \left( \left[ \begin{array}{c} 1 \left\{ \begin{array}{c} 10 \\ 20 \end{array} \right\} \\ 3 \quad 4 \\ 5 \\ 6 \end{array} \right] \right) \right) = \left[ \begin{array}{c} 1 \left\{ \begin{array}{c} 10 \\ 20 \end{array} \right\} \\ 3 \quad 4 \\ 5 \\ 6 \end{array} \right]$$

### 6.9.4. Maxima-Funktionen für Listen

Maxima ist in Lisp (*list processing language*) geschrieben, also sind Listen im Umgang mit Maxima wichtig. Einige Funktionen, die die Arbeit mit Listen vereinfachen, seien hier genannt.

`args()` liefert die Parameter einer Funktion als Liste. Das kann nützlich sein, wenn Maxima Funktionen mit Argumenten als Ergebnis liefert und man diese Argumente extrahieren möchte:



$$\text{args}(f(x;y)) = \begin{cases} x \\ y \end{cases}$$

Andersrum kann man mit `apply()` die Elemente einer Liste zu Argumenten einer Funktion machen:

$$\text{apply}(f; \begin{cases} a \\ b \end{cases}) = f(a; b)$$

Mit `append()` können Listen aneinandergelagert werden:

$$\text{append}\left(\begin{cases} a \\ b \end{cases}; \begin{cases} c \\ d \end{cases}\right) = \begin{cases} a \\ b \\ c \\ d \end{cases}$$

Um eine Matrix in eine Liste von Listen (mit den Matrixzeilen) umzuwandeln, benutzen Sie ebenfalls die Funktion `args()`:

$$\text{args}\left(\begin{bmatrix} 1 & 2 \\ a & b \end{bmatrix}\right) = \begin{cases} \begin{cases} 1 \\ 2 \end{cases} \\ \begin{cases} a \\ b \end{cases} \end{cases}$$

## 6.10. Analysis

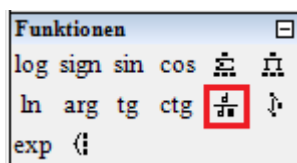
SMath Studio kann von Haus aus Ausdrücke symbolisch differenzieren, aber nur numerisch integrieren (mit einem sehr einfachen Algorithmus). Mit Maxima kann man auch symbolisch integrieren und Grenzwerte und Summen bilden.

| Darstellung   | Eingabe              | Bedeutung                               |
|---------------|----------------------|---|
| '             | <code>diff(</code>   | Differentiation (Maxima)                |
| $\frac{d}{d}$ | <code>dif</code>     | Differentiation (symbolisch)            |
| $\frac{d}{d}$ | <code>dif</code> ;   | Differentiation (symbolisch)            |
| $\nabla$      | <code>gra</code>     | Gradient (inaktiv)                      |
| $\nabla$      | <code>gra</code> ;   | Gradient an einer Stelle (inaktiv)      |
| $\int$        | <code>int(;</code>   | unbestimmtes Integral (Maxima)          |
| $\int$        | <code>int</code> ; ; | bestimmtes Integral (SMath oder Maxima) |

|                         |                  |                           |
|-------------------------|------------------|---------------------------|
| <code>lim</code>        | <code>lim</code> | Grenzwert (Maxima)        |
| <code>Hessian()</code>  |                  | Hessematrix (symbolisch)  |
| <code>Jacobian()</code> |                  | Jacobimatrix (symbolisch) |
| <code>Taylor()</code>   |                  | Taylorreihe (symbolisch)  |

### 6.10.1. Differenzialrechnung

**Ableitung berechnen.** Dazu dient die Funktion `diff()`, welche die erste oder höhere Ableitungen symbolisch berechnen kann. Sie ist auch über das Funktionen-Panel der Seitenleiste zugänglich:



Hier einige Beispiele:

$$\frac{d}{dx} u(x) \cdot v(x) = \frac{d}{dx} u(x) \cdot v(x) + u(x) \cdot \frac{d}{dx} v(x)$$

$$\frac{d}{dx} \sin(x) = \cos(x) \quad \frac{d^2}{dx^2} \sin(x) = -\sin(x)$$

$$f(x) := a \cdot x^3 + b \cdot x^2 + c \cdot x + d$$

$$\frac{d}{dx} f(x) = c + x \cdot (b + a \cdot x + b + 2 \cdot a \cdot x)$$

$$\frac{d^2}{dx^2} f(x) = 2 \cdot (b + a \cdot x + 2 \cdot a \cdot x)$$

Soll die Ableitung einer Funktion  $f(x)$  an einer Stelle  $x_0$  ausgewertet werden, muss zunächst die Ableitung gebildet und dann der Wert eingesetzt werden (mit der Substitutionsfunktion):

$$\frac{d^2}{dx^2} f(x_0) = 0 \quad \left. \frac{d^2}{dx^2} f(x) \right|_{x=x_0} = 2 \cdot (b + a \cdot x_0 + 2 \cdot a \cdot x_0)$$

Die Funktion `Jacobian()` bildet die **Jacobimatrix**, also die Matrix aus den ersten Ableitungen eines Funktionsvektors nach den unabhängigen Variablen:

$$\text{Jacobian} \left( \begin{matrix} f_1(x; y) \\ f_2(x; y) \end{matrix}; \begin{matrix} x \\ y \end{matrix} \right) = \begin{bmatrix} \frac{d}{dx} f_1(x; y) & \frac{d}{dy} f_1(x; y) \\ \frac{d}{dx} f_2(x; y) & \frac{d}{dy} f_2(x; y) \end{bmatrix}$$

$$\text{Jacobian} \left( \begin{matrix} e^{2 \cdot x + 2} \\ x \cdot \cos(3 \cdot y) \end{matrix}; \begin{matrix} x \\ y \end{matrix} \right) = \begin{bmatrix} 2 \cdot e^{2 \cdot (1+x)} & 0 \\ \cos(3 \cdot y) & -3 \cdot \sin(3 \cdot y) \cdot x \end{bmatrix}$$

Die unabhängigen Variablen müssen nicht unterschiedliche Namen haben, es können auch mit Elementindizes adressierte Elemente der gleichen Variable sein:

$$v(x) := \begin{bmatrix} e^{2 \cdot x_1 + 2} \\ x_1 \cdot \cos(3 \cdot x_2) \end{bmatrix}$$

$$\text{Jacobian} \left( v(x); \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right) = \begin{bmatrix} 2 \cdot e^{2 \cdot (1+x_1)} & 0 \\ \cos(3 \cdot x_2) & -3 \cdot \sin(3 \cdot x_2) \cdot x_1 \end{bmatrix}$$

Die Funktion  $\text{Hessian}(f(x); x)$  berechnet die **Hesse-Matrix**, also die Matrix der zweiten Ableitungen einer skalaren Funktion mehrerer Variablen. Die unabhängigen Variablen können als Vektor oder Liste angegeben werden:

$$\text{Hessian} \left( x \cdot \cos(3 \cdot y); \begin{matrix} x \\ y \end{matrix} \right) = \begin{bmatrix} 0 & -3 \cdot \sin(3 \cdot y) \\ -3 \cdot \sin(3 \cdot y) & -9 \cdot \cos(3 \cdot y) \cdot x \end{bmatrix}$$

$$\text{Hessian} \left( f(x; y); \begin{matrix} x \\ y \end{matrix} \right) = \begin{bmatrix} \frac{d}{dx} \frac{d}{dx} f(x; y) & \frac{d}{dx} \frac{d}{dy} f(x; y) \\ \frac{d}{dy} \frac{d}{dx} f(x; y) & \frac{d}{dy} \frac{d}{dy} f(x; y) \end{bmatrix}$$

Eine **Taylor-Reihe** approximiert eine gegebene Funktion  $f(x)$  in einem Punkt  $x_0$  (Entwicklungspunkt) mit einem Polynom  $n$ -ter Ordnung. Dies leistet in SMATH die Funktion  $\text{Taylor}(f(x); x_0; n)$ .

$$\text{Taylor}(F(x); a; 0) = F(a)$$

$$\text{Taylor}(F(x); a; 1) = \frac{d}{da} F(a) \cdot (x - a) + F(a)$$

$$\text{Taylor}(F(x); a; 2) = \frac{(x - a) \cdot \left( 2 \cdot \frac{d}{da} F(a) + \frac{d}{da} \frac{d}{da} F(a) \cdot (x - a) \right) + 2 \cdot F(a)}{2}$$

Besonders beliebt sind Taylor-Reihen erster Ordnung, sie approximieren die Funktion durch eine Gerade durch  $f(x_0)$  mit dem Anstieg  $df(x)/dx$  an der Stelle  $x = x_0$  (Linearisierung).

$$\text{Taylor}(\sin(x); 0; 1) = x$$

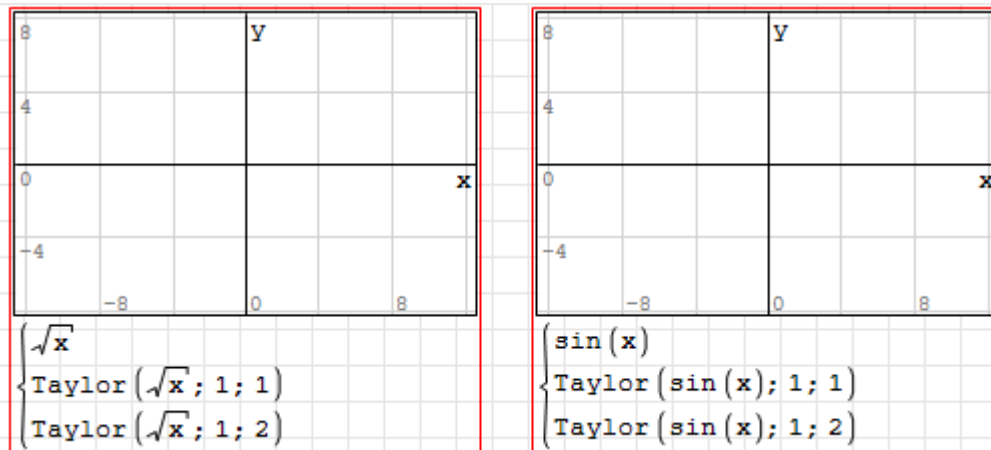
$$\text{Taylor}(\sqrt{x}; a; 1) = \frac{x + a}{2 \cdot \sqrt{a}}$$

Das zweite Argument (Entwicklungspunkt) muss in der Form  $x = x_0$  geschrieben werden, wenn der zu approximierende Ausdruck mehr als eine Variable enthält:

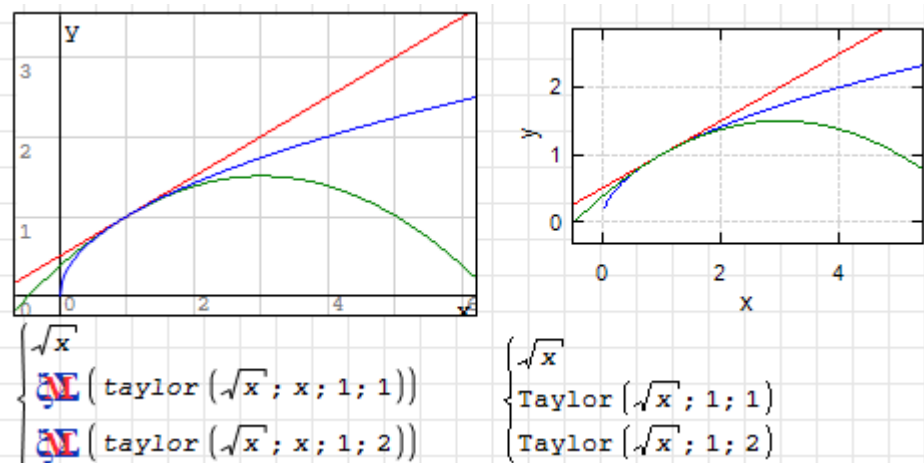
$$\text{Taylor}(\sqrt{a^2 + x}; x = 0; 1) = \frac{2 + x}{2}$$

$$\text{Taylor}(x^n; x = a; 1) = 1 + n \cdot (-1 + x)$$

In den folgenden Bildern sollen eigentlich die Wurzelfunktion und die Sinusfunktion mit ihren **linearen** und quadratischen Approximationen gezeigt werden, aufgrund eines Fehlers (SS-4631)<sup>3</sup> in der Funktion Taylor() gelingt dies jedoch nicht im Standard-Diagrammbereich:



Alternativ kann man auf die Maxima-Funktion taylor() oder auf den X-Y-Plotbereich oder Maxima-Plots ausweichen:



### 6.10.2. Symbolische Integration mit Maxima

Maxima kann unbestimmte Integrale symbolisch berechnen, dies muss mit MaximaTakeover() aktiviert werden.

$$\text{MaximaTakeover}(\text{"int()"}) = \text{"int() handled by Maxima"}$$

$$\int x^2 dx = \frac{x^3}{3}$$

<sup>3</sup>[smath.com/bts/Issues/IssueDetail.aspx?id=4631](http://smath.com/bts/Issues/IssueDetail.aspx?id=4631)

$$\int \ln(x)^3 dx = x \cdot (\ln(x)^3 - 3 \cdot \ln(x)^2 + 6 \cdot \ln(x) - 6)$$

$$\int \frac{x^3 + 2 \cdot x^2 - x + 1}{x^3 + 3 \cdot x^2 + x + 3} dx = -\frac{\ln(x^2 + 1)}{4} - \frac{\ln(x + 3)}{2} - \frac{\operatorname{atan}(x)}{2} + x$$

Die Maxima-Integrationsfunktion kann mit Maßeinheiten umgehen. Integrand, Integrationsvariable und Integrationsgrenzen können Maßeinheiten haben.

$$q(x) := q_0 + m \cdot x \quad q_0 := 10 \frac{\text{N}}{\text{mm}} \quad m := 0,2 \frac{\text{N}}{\text{mm}^2} \quad q(2 \text{ m}) = 410 \frac{\text{N}}{\text{mm}}$$

$$x_1 := 1 \text{ m} \quad x_2 := 3 \text{ m} \quad \int_{x_1}^{x_2} q(x) dx = 820 \text{ kN}$$

Maxima geht beim symbolischen Integrieren nicht automatisch davon aus, dass die Untergrenze kleiner als die Obergrenze ist und fragt zurück. Da der Anwender normalerweise keinen Grund hat, die Untergrenze größer als die Obergrenze zu setzen, wird diese Abfrage von der Maxima-SMath-Schnittstelle abgefangen. Dem Ergebnis wird dann die Meldung über die getroffene Annahme hinzugefügt:

$$\varepsilon(x) := \varepsilon_0 \cdot \left(\frac{x}{x_0}\right) \quad y(x) := y_0 \cdot \left(\frac{x}{x_0}\right)^2$$

$$\int_0^{x_0} \varepsilon(x) \cdot \sqrt{1 + \left(\frac{d}{dx} y(x)\right)^2} dx = \left[ \frac{\left(2 \sqrt{x_0^2 \cdot (4 \cdot y_0^2 + x_0^2)}^3 - x_0^6\right) \cdot \varepsilon_0}{12 \cdot y_0^2 \cdot x_0^3} \right]$$

"x\_0 is assumed to be positive."

Meldung und eigentliches Ergebnis werden in einer Liste zurückgegeben. Möchte man die Frage vermeiden, muss man vorher Maxima eine Annahme mitteilen. Das macht man mit der Maxima-Funktion `assume()`.

$$\mathcal{M}(\text{assume}(x_0 > 0)) = \{x_0 > 0\}$$

$$\int_0^{x_0} \varepsilon(x) \cdot \sqrt{1 + \left(\frac{d}{dx} y(x)\right)^2} dx = \frac{\left(2 \sqrt{x_0^2 \cdot (4 \cdot y_0^2 + x_0^2)}^3 - x_0^6\right) \cdot \varepsilon_0}{12 \cdot y_0^2 \cdot x_0^3}$$

Macht man mehrmals die gleiche Annahme, kommt die Meldung **redundant** als Ergebnis der `assume()`-Funktion.

Möchte man eine Annahme ändern, muss man sie vorher mit `forget()` löschen, sonst kommt die Meldung **inconsistent**.

$$\mathcal{M}(\text{assume}(x_0 < 0)) = \{\text{inconsistent}\}$$

$$\mathcal{M}(\text{forget}(x_0 > 0)) = \{x_0 > 0\}$$

$$\mathcal{M}(\text{assume}(x_0 < 0)) = \{x_0 < 0\}$$

$$\int_0^{x_0} \varepsilon(x) \cdot \sqrt{1 + \left(\frac{d}{d x} y(x)\right)^2} dx = - \frac{\left(x_0^6 - 2\sqrt{x_0^2 \cdot (4 \cdot y_0^2 + x_0^2)}^3\right) \cdot \varepsilon_0}{12 \cdot y_0^2 \cdot x_0^3}$$

Bei Maßeinheiten weiß die Integrationsfunktion in Maxima, dass sie positiv sind. Man kann also auch eine Variable als Maßeinheit kennzeichnen (vorangestelltes Hochkomma), um Vorzeichenfragen zu vermeiden:

$$x_0 := x_0 \quad y_0 := y_0$$

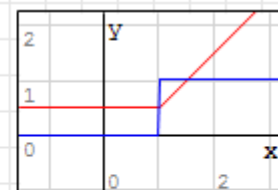
$$\int_0^{x_0} \varepsilon(x) \cdot \sqrt{1 + \left(\frac{d}{d x} y(x)\right)^2} dx = \frac{\varepsilon_0 \cdot \left(2\sqrt{4 y_0^2 + x_0^2}^3 - x_0^3\right)}{12 y_0^2}$$

Stückweise stetige Funktionen müssen mit `sign()` definiert werden, da die Übersetzung boolscher Ausdrücke noch nicht implementiert ist.

$$\text{step}(x; x_0) := \frac{1}{2} \cdot (1 + \text{sign}(x - x_0)) \quad \text{step}(1; 1) = 0,5$$

$$\int_0^2 \text{step}(x; 1) dx = 1$$

$$\int \text{step}(x; 1) dx = \frac{x + |-1 + x|}{2}$$



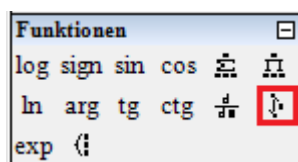
$$\left\{ \begin{array}{l} \text{step}(x; 1) \\ \int \text{step}(x; 1) dx \end{array} \right.$$

### 6.10.3. Numerische Integration

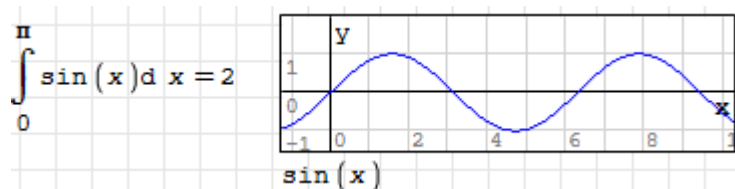
Zum Nachvollziehen der Beispiele in diesem Abschnitt verwenden Sie ein neues Rechenblatt oder stellen Sie sicher, dass nicht Maxima, sondern SMath die Integration ausführt:

```
MaximaTakeover("none") = "All functions handled by SMath"
```

SMath kann nur bestimmte Integrale (mit Integrationsgrenzen) und diese auch nur numerisch berechnen. Dazu dient die Funktion `int(f;x;a;b)`. Sie ist neben Tastatur und **Einfügen** > **Funktion** auch über **Seitenleiste** > **Funktionen** zugänglich:



Ein Beispiel:



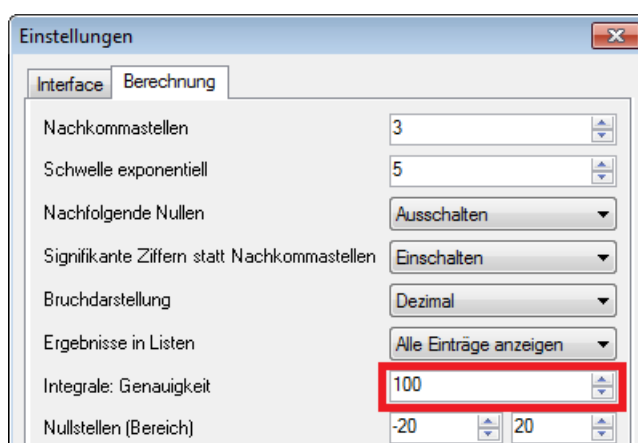
### Einschränkungen und Empfehlungen

- Maßeinheiten an den Integrationsgrenzen werden ignoriert. Die Integrationsvariable wird daher als dimensionslos betrachtet.
- Die Funktionen müssen an allen Stellen im Integrationsintervall auswertbar sein, dürfen also keine undefinierten Stellen enthalten.
- Sie dürfen keine Fallunterscheidungen mit `if()` enthalten. Verwenden Sie statt dessen `cases()` oder logische Ausdrücke für stückweise stetige Funktionen (siehe Abschnitt 6.5.3).
- Die Integrationsgenauigkeit hängt von der eingestellten Zahl der Intervalle ab, das wird im folgenden Abschnitt erläutert.
- Im Interesse der Genauigkeit ist es vorteilhaft, das Integrationsgebiet an Unstetigkeitsstellen aufzuteilen.

Die numerische Berechnung sollten Sie nur dann verwenden, wenn Maxima kein symbolisches Ergebnis findet.

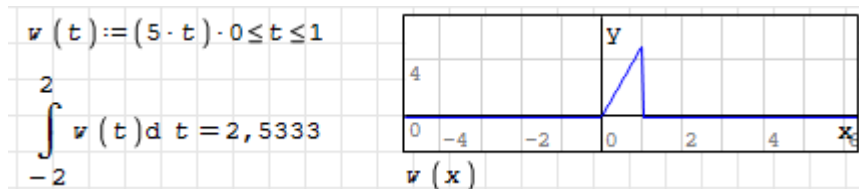
### Genauigkeit

Integrale werden intern numerisch berechnet. Die Genauigkeit hängt von der Anzahl der Unterteilungen des Integrationsbereichs ab. Diese ist einzustellen unter **Extras > Einstellungen > Berechnung > Integrale: Genauigkeit**



Je glatter eine Funktion verläuft, umso weniger Unterteilungen braucht man. Bei Funktionen, die sehr unstetig sind, muss man den Wert eventuell höher setzen. An Polstellen kann man überhaupt keine sinnvollen Werte erwarten.

Die folgende Funktion ist ein Dreiecksimpuls, der für  $0 \leq x \leq 1$  von 0 auf 5 ansteigt und ansonsten den Wert 0 hat. Die Fläche unter dem Impuls ist gleich 2,5. Zur Verdeutlichung der Unterschiede wird das Ergebnis mit vier Nachkommastellen angezeigt.

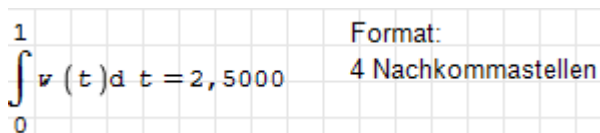


Eine Konvergenzanalyse zeigt folgende Abhängigkeit des Ergebnisses von der eingestellten Genauigkeit:

| Genauigkeitseinstellung | Wert   |
|-------------------------|--------|
| exakt                   | 2,5000 |
| 100 (Voreinstellung)    | 2,5333 |
| 1000                    | 2,5033 |
| 10000                   | 2,5003 |

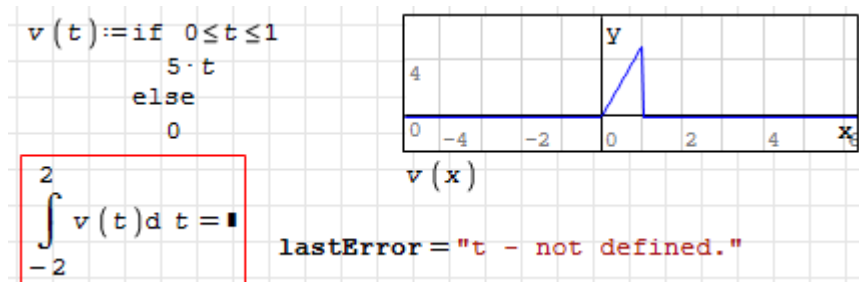
Aufgrund der Unstetigkeit bei  $t = 1$  kommt es bei der Standardeinstellung (Genauigkeit 100) zu einem Fehler bereits in der dritten Mantissenstelle.

Wird das Intervall so beschränkt, dass die Funktion darin stetig ist, bekommt man einen deutlich genaueren Wert

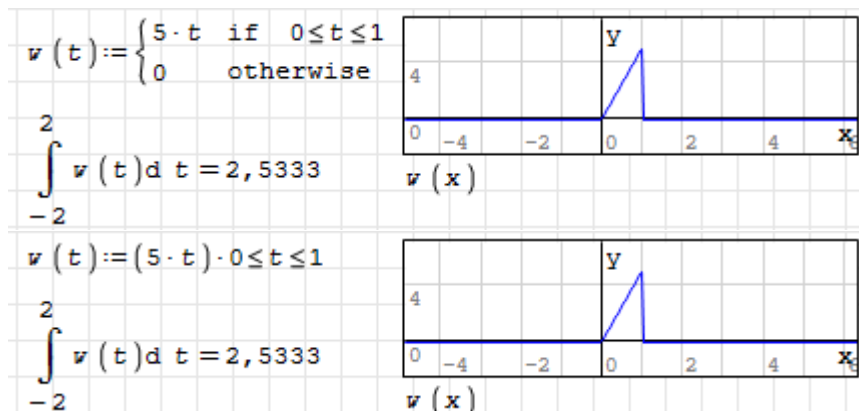


### Integrale stückweise stetiger Funktionen

Die Integrationsfunktion `int()` versagt, wenn Funktionen unter Verwendung der Fallunterscheidungen `if()` definiert werden. Es kommt dann die unsinnige Fehlermeldung, die Integrationsvariable sei nicht definiert.



Alternativ kann man die `cases()`-Anweisung benutzen oder direkt mit logischen Ausdrücken multiplizieren:





Innerlich besteht die `cases()`-Anweisung auch nur aus logischen Ausdrücken.

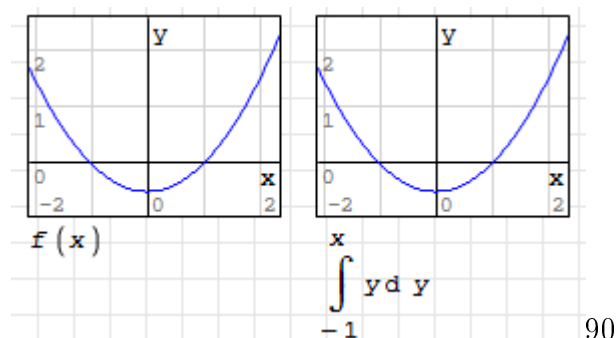
### Integrale mit variablen Grenzen

Man kann Funktionen definieren, die Integrationen mit variablen Grenzen enthalten.

Achten Sie darauf, unterschiedliche Variablen für Grenzen (hier  $x$ ) und Integration (hier  $y$ ) zu verwenden!

$$f(x) := \int_{-1}^x y \, dy \quad x := [-2..2] = \begin{bmatrix} -2 \\ -1 \\ 0 \\ 1 \\ 2 \end{bmatrix} \quad f := f(x) = \begin{bmatrix} 1,5 \\ 0 \\ -0,5 \\ 0 \\ 1,5 \end{bmatrix}$$

Derartige Ausdrücke können auch direkt geplottet werden:



### Integrale über Größen mit Maßeinheiten

SMath Studio kann die numerische Integration nur mit dimensionsloser Integrationsvariable ausführen. Maßeinheiten an den Integrationsgrenzen werden ignoriert. Maßeinheiten des Integranden werden allerdings korrekt ins Ergebnis aufgenommen.

Es ist kein grundsätzliches Problem, eine einheitenfeste numerische Integrationsfunktion zu definieren, wie weiter unten gezeigt wird. Sie ist aber leider noch nicht in SMath Studio implementiert. Der entsprechende Entwicklungsvorschlag findet sich als [SS-85<sup>4</sup>](#) im Ticket-System von SMath Studio.

Vorerst muss man die **Transformation auf eine dimensionslose Integrationsvariable** selbst erledigen. Es sei ein Integral der Form

$$\int_{x_1}^{x_2} f(x) dx$$

zu berechnen.

- Die Maßeinheit  $M$  der Integrationsvariable  $x$  wird aus den Integrationsgrenzen  $x_1$  und  $x_2$  ermittelt.
- Die Integrationsvariable  $x$  wird durch eine dimensionslose Variable  $\xi = x/M$  ersetzt.

<sup>4</sup><http://smath.info/bts/Issues/IssueDetail.aspx?id=85>

- Die Integrationsgrenzen werden ebenfalls durch die Maßeinheit  $M$  geteilt.
- Die Variable  $x$  wird im Integranden durch  $M \xi$  ersetzt
- Das Ergebnis wird mit  $M$  multipliziert.

Es gilt also

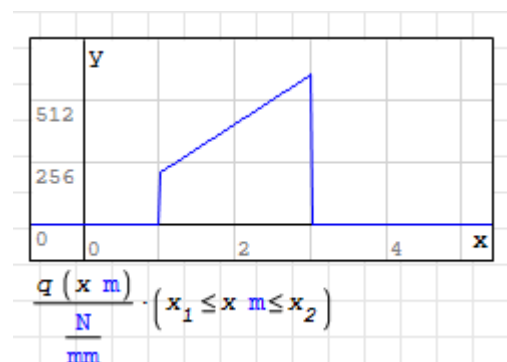
$$\int_{x_1}^{x_2} f(x) dx = M \int_{x_1/M}^{x_2/M} f(\xi M) d\xi$$

Im folgenden Beispiel wird eine Streckenlast (Dimension Kraft pro Länge) über eine Länge integriert. Das Ergebnis ist also eine Kraft.

$$q(x) := q_0 + m \cdot x \quad q_0 := 10 \frac{\text{N}}{\text{mm}}$$

$$m := 0,2 \frac{\text{N}}{\text{mm}} \quad q(2 \text{ m}) = 410 \frac{\text{N}}{\text{mm}} \quad x_1 := 1 \text{ m} \quad x_2 := 3 \text{ m}$$

Bei der grafischen Darstellung müssen Funktion und unabhängige Variable auch dimensionslos gemacht werden. Dadurch kann man aber die tatsächlichen Einheiten im Plot erkennen. Der Faktor bei  $x$  ist die Einheit der  $x$ -Achse, die Einheit, durch die das Ergebnis geteilt wird, ist die Einheit der  $y$ -Achse.



Versucht man einfach das Integral zu berechnen, so erhält man bei symbolischer Auswertung das Integral mit vereinfachtem Integranden. Die Summe  $20x + m$  kann nicht funktionieren wenn  $x$  dimensionslos ist.

Numerische Auswertung bringt eine sinnlose Fehlermeldung.

|  |   |           |
|--|---|-----------|
| symbolisch   |   | numerisch |
| $\int_{x_1}^{x_2} q(x) dx = \int_m^{3 \text{ m}} \frac{10000 \text{ kg} \cdot (m + 20 \cdot x)}{2} dx$ | $\int_{x_1}^{x_2} q(x) dx = \blacksquare$ |           |
| lastError = "x - not defined."   |   |           |

Nun die richtige Vorgehensweise:

Zunächst wird die Maßeinheit der Integrationsvariable aus den Grenzen mit der Funktion `UnitsOf()` ermittelt. Mit der Substitutionsfunktion `at()` wird die Variablentransformation im Integranden vorgenommen. Dann wird das transformierte Integral wie oben beschrieben ausgewertet:

$$\text{ME} := \text{UnitsOf} \left( \frac{x_2 - x_1}{\text{ME}} \right) \quad q(x) \Big|_{x = \text{ME} \cdot \xi} = \frac{10000 \text{ kg} \cdot (1 + 20 \cdot \xi)}{\text{s}^2}$$

$$\text{ME} \cdot \int_{\frac{x_1}{\text{ME}}}^{\frac{x_2}{\text{ME}}} q(x) \Big|_{x = \text{ME} \cdot \xi} d\xi = 820 \text{ kN}$$

### Referenzimplementierung für eine einheitensichere Integrationsfunktion

Die beschriebene Vorgehensweise kann auch in eine Funktion gegossen werden, die an die Stelle der bisherigen Integrationsfunktion treten könnte:

$$\text{Int}(f\#; x\#; a\#; b\#) := \left| \begin{array}{l} u\# := \text{UnitsOf}(b\# - a\#) \\ \frac{b\#}{u\#} \\ u\# \cdot \int_{\frac{a\#}{u\#}}^{\frac{b\#}{u\#}} f\# \Big|_{x\# = xt\# \cdot u\#} dxt\# \\ \frac{a\#}{u\#} \end{array} \right.$$

$$\text{Int}(q(x); x; x_1; x_2) = 820 \text{ kN}$$

$$\text{Int}(g_e \cdot t; t; 0; 1 \text{ s}) = 4,90 \text{ m}$$

## 6.11. Gleichungslösung

Für die Lösung von linearen und nichtlinearen Gleichungssystemen ist stets die Funktion `Solve()` aus dem Maxima-Plug-in (siehe Kapitel 7) zu empfehlen.

- Lineare und nichtlineare Gleichungssysteme werden symbolisch gelöst.
- Startwerte oder Suchbereiche werden nicht benötigt.
- Das Ergebnis wird in Form von Listen mit Gleichungen `Variable=Wert` zurückgegeben.
- Die Funktion kann mit Maßeinheiten umgehen.

In manchen Fällen werden die symbolischen Ausdrücke zu kompliziert und `Solve()` findet keine Lösung. Dann kann man die Funktion `FindRoot()` aus dem Plug-in *Nonlinear Solvers* benutzen. Das ist ein numerischer Löser, der Startwerte für die Variablen benötigt. Er kann ebenfalls mit Maßeinheiten umgehen. Darauf wird in Abschnitt 6.11.2 eingegangen.

Hat man nur eine Gleichung für eine Variable und kennt den Bereich, in dem die Lösung liegen muss, bietet sich auch das Halbteilungsverfahren an (Funktion `Bisection()`, siehe Abschnitt 6.11.3).

SMATH Studio selbst bringt die Funktionen `solve()` und `roots()` mit, die allerdings weniger leistungsfähig sind und für den Standardanwender nicht empfohlen werden. Ihre Beschreibung wurde daher in den Anhang ausgelagert (Abschnitt G.2).

### 6.11.1. Symbolische Lösungen mit Solve()

Diese Funktion verwendet die Maxima-Funktion `solve()`. Das Gleichungssystem wird als Liste oder Vektor von Gleichungen vorgegeben, dabei wird das boolesche (fette) Gleichheitszeichen verwendet. Eine weitere Liste oder ein weiterer Vektor geben die Variablen an, nach denen aufzulösen ist.

Die Lösung kommt als Liste von Gleichungen der Form `Name=Wert`. Wenn es mehrere Lösungen gibt, werden diese als Matrix von Listen (bei mehreren Variablen) oder als Liste (bei einer Variablen) zurückgegeben.

Die Zuweisung der Lösung auf die Variablen ist mit der Funktion `Assign()` oder mit der Substitutionsfunktion `at()` möglich. `Assign()` interpretiert die Gleichungen `Name=Wert` als Zuweisungen und führt diese aus. Bei mehreren Lösungen muss man sich dabei für eine Lösung entscheiden, die man über einen Elementindex an der Ergebnismatrix auswählen kann.

Hinweis: Verwenden Sie stets die symbolische Auswertung mit `Strg-]`. Das Ergebnis enthält undefinierte Variablen und die kann SMATH Studio sonst nicht anzeigen.

#### Eine Gleichung, eine Unbekannte, eine Lösung

Einzelne Gleichungen oder Variablen können auch skalar (also nicht als Liste oder Matrix) angegeben werden.

$$\text{Solve}(a \cdot x + b = y; a) = a = \frac{y - b}{x}$$

Die Lösung wird der Variable mit Funktion `Assign()` aus dem Maxima-Plugin zugewiesen:

$$\begin{aligned} L := \text{Solve}(a \cdot x + b = y; a) & & L = a = \frac{y - b}{x} \\ \text{Assign}(L) = \frac{y - b}{x} & & a = \frac{y - b}{x} \end{aligned}$$

#### Zwei Gleichungen, zwei Unbekannte, eine Lösung

$$\begin{aligned} \text{Clear}(a) = 1 \\ L := \text{Solve}\left(\left[\begin{array}{l} 3 \cdot x + 4 \cdot y = 7 \\ 2 \cdot x + a \cdot y = 13 \end{array}\right]; \left[\begin{array}{l} x \\ y \end{array}\right]\right) & & L = \begin{cases} x = \frac{-52 + 7 \cdot a}{-8 + 3 \cdot a} \\ y = \frac{25}{-8 + 3 \cdot a} \end{cases} \\ \text{Assign}(L) = \begin{cases} \frac{-52 + 7 \cdot a}{-8 + 3 \cdot a} \\ 25 \\ \frac{-8 + 3 \cdot a}{-8 + 3 \cdot a} \end{cases} & & x = \frac{-52 + 7 \cdot a}{-8 + 3 \cdot a} \quad y = \frac{25}{-8 + 3 \cdot a} \end{aligned}$$

### Eine Gleichung, eine Unbekannte, drei Lösungen

$$\text{Clear}(x; y) = 1$$

$$L := \left( \text{Solve} \left( x^3 - 3 \cdot x^2 + 5 \cdot x = 15; x \right) \right) \quad L = \begin{cases} x = -i \cdot \sqrt{5} \\ x = i \cdot \sqrt{5} \\ x = 3 \end{cases}$$

Mehrere Lösungen für eine einzelne Variable werden als Listen angezeigt. Man kann per Elementindex eine davon auswählen.

$$\begin{array}{lll} \text{Assign}(L_1) = -i \cdot \sqrt{5} & x = -2,24 \cdot i & \text{Clear}(x) = 1 \\ \text{Assign}(L_2) = i \cdot \sqrt{5} & x = 2,24 \cdot i & \text{Clear}(x) = 1 \\ \text{Assign}(L_3) = 3 & x = 3 & \text{Clear}(x) = 1 \end{array}$$

Bei der Lösung und der Auswahl der Lösung dürfen die verwendeten Variablen noch keinen Wert haben. Gegebenenfalls muss ein vorhandener Wert mit `Clear()` gelöscht werden.

### Zwei Gleichungen, zwei Unbekannte, zwei Lösungen

Hier werden die Gleichungen einzeln zugewiesen. `G1` ist ein Vektor, der alle Gleichungen des Systems enthält. Das kann sinnvoll sein, wenn Sie die Bedeutung oder Herleitung der Gleichungen einzeln erläutern wollen. Bei der Zuweisung muss die Gleichung in Klammern stehen, da der Zuweisungsoperator eine höhere Priorität als das Boolesche Gleichheitszeichen hat.

$$G1_1 := (x^2 + 3 \cdot x \cdot y + y^2 = 0) \quad G1_2 := (3 \cdot x + y = 1)$$

$$L := \text{Solve} \left( G1; \begin{bmatrix} x \\ y \end{bmatrix} \right) \quad L = \begin{cases} x = -\frac{-3 + \sqrt{5}}{2} \\ y = \frac{-7 + 3 \cdot \sqrt{5}}{2} \end{cases} \quad \begin{cases} x = \frac{3 + \sqrt{5}}{2} \\ y = -\frac{7 + 3 \cdot \sqrt{5}}{2} \end{cases}$$

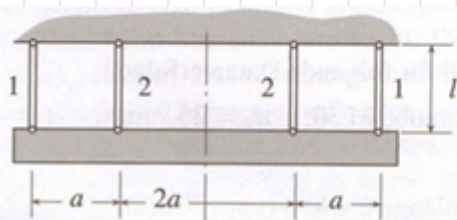
$$\text{Assign}(L_1) = \begin{cases} -\frac{-3 + \sqrt{5}}{2} \\ -\frac{7 + 3 \cdot \sqrt{5}}{2} \end{cases} \quad x = -\frac{-3 + \sqrt{5}}{2} \quad y = \frac{-7 + 3 \cdot \sqrt{5}}{2} \quad \text{Clear}(x; y) = 1$$

$$\text{Assign}(L_2) = \begin{cases} \frac{3 + \sqrt{5}}{2} \\ -\frac{7 + 3 \cdot \sqrt{5}}{2} \end{cases} \quad x = \frac{3 + \sqrt{5}}{2} \quad y = -\frac{7 + 3 \cdot \sqrt{5}}{2} \quad \text{Clear}(x; y) = 1$$

#### 6.11.1.1. Anwendungsbeispiel: Lineares System

Die Aufgabenstellung wurde aus Assmann/Selke 2009, Aufgabe 3-26, gescannt und als Bildbereich eingefügt.

3-26 Eine homogene, starre Masse ist an vier Stäben aufgehängt. Diese sind gleich, jedoch sind die Stäbe (2) um  $s$  zu kurz gefertigt. In allgemeiner Form und für  $m = 400 \text{ kg}$ ;  $l = 2,0 \text{ m}$ ;  $A = 20 \text{ mm}^2$ ;  $s = 0,50 \text{ mm}$  sind die Stabkräfte zu bestimmen. Die Ergebnisse sind zu diskutieren.



Die gegebenen Größen werden mit den Maßeinheiten aus der Aufgabenstellung übernommen.

**Gegeben**  $m := 400 \text{ kg}$   $l := 2 \text{ m}$   $A := 20 \text{ mm}^2$   $s := 0,5 \text{ mm}$   
 $E := 210000 \text{ MPa}$  (Annahme, hat keine Auswirkung auf das Ergebnis)  
**Gesucht**  $S_1, S_2$

Aus dem elastischen Gesetz für die Stäbe, der Geometriebedingung (alle Stäbe müssen im verformten Zustand gleich lang sein) und der Gleichgewichtsbedingung ergeben sich vier Gleichungen, die das mathematische Modell des Problems bilden. Die Gleichungen werden in die Variable G1 an die durch den jeweiligen Elementindex bezeichnete Stelle geschrieben. Wenn diese Variable vorher nicht existiert, wird sie durch die Zuweisung von Elementen automatisch zu einem Vektor.

#### Modellbildung

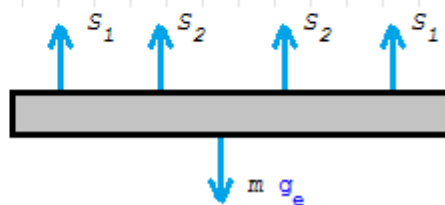
Elastisches Gesetz für die Stäbe

$$G1_1 := \left( \Delta l_1 = \frac{1}{E \cdot A} \cdot S_1 \right) \quad G1_2 := \left( \Delta l_2 = \frac{1}{E \cdot A} \cdot S_2 \right)$$

Geometrie  $G1_3 := (\Delta l_2 = \Delta l_1 + s)$

Gleichgewicht an der Masse in vertikaler Richtung

$$G1_4 := (m \cdot g_e = 2 \cdot S_1 + 2 \cdot S_2)$$



Mit der Funktion `Unknowns()` kann man prüfen, welche Variablen im Gleichungssystem unbekannt sind, also noch keinen Wert haben. Die Funktion `Solve()` löst das System und liefert das Ergebnis als Liste von logischen Gleichungen `Variable=Wert` zurück. Das ist hier nicht zu sehen, da das Ergebnis direkt in `Assign()` als Variablendefinition weiter verwendet wird.

Berechnung

$$\text{Unknowns (Gl)} = \begin{bmatrix} S_1 \\ S_2 \\ \Delta l_1 \\ \Delta l_2 \end{bmatrix} \text{ Assign } \left( \text{Solve } \left( \text{Gl}; \begin{bmatrix} S_1 \\ S_2 \\ \Delta l_1 \\ \Delta l_2 \end{bmatrix} \right) \right) = \begin{cases} 456 \frac{\text{kg m}}{\text{s}^2} \\ 1510 \frac{\text{kg m}}{\text{s}^2} \\ 0,000217 \text{ m} \\ 0,000717 \text{ m} \end{cases}$$

$$S_1 = 0,456 \text{ kN} \quad S_2 = 1,51 \text{ kN}$$

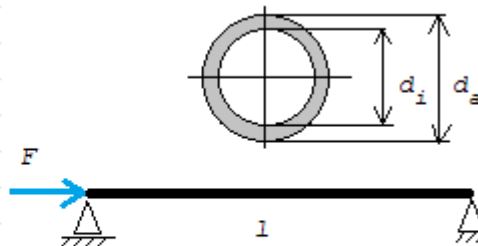
**6.11.1.2. Anwendungsbeispiel: Nichtlineares System**

Die Aufgabenstellung ist wiederum Assmann/Selke 2009, Aufgabe 7-7, entnommen.

Eine Druckspindel,  $l = 800 \text{ mm}$  lang, mit  $F = 20 \text{ kN}$  belastet, soll als Hohlwelle mit  $d_i = 25 \text{ mm}$  ausgeführt werden. Für  $S_K > 3,5$  ist der Außendurchmesser auf volle mm gerundet zu bestimmen.

Gegebene Größen

|                           |  |
|---------------------------|--|
| $l := 800 \text{ mm}$     | Spindellänge                           |
| $F := 20 \text{ kN}$      | Druckkraft                             |
| $d_i := 25 \text{ mm}$    | Innendurchmesser                       |
| $S_K := 3,5$              | Knicksicherheit                        |
| $E := 210000 \text{ MPa}$ | E-Modul Stahl (Annahme)                |
| $l_K := 1$                | beidseitig gelenkig gelagert (Annahme) |

Gleichungen

|  |   |
|--|---|
| Knicklast  | Flächenträgheitsmoment der Hohlwelle                              |
| $Gl_1 := \left( F \cdot S_K = \frac{\pi^2 \cdot E \cdot I}{l_K^2} \right)$ | $Gl_2 := \left( I = \frac{\pi}{64} \cdot (d_a^4 - d_i^4) \right)$ |

$$\text{Unknowns (Gl)} = \begin{bmatrix} d_a \\ I \end{bmatrix}$$

Gleichung 2 ist nichtlinear, es können also mehrere Lösungen auftreten. Die korrekte Lösung muss dann aus dem Zusammenhang der Aufgabenstellung ermittelt werden. Im vorliegenden Fall gibt es vier Lösungen (vier mögliche Werte für die vierte Wurzel).

Lösung

$$L := \text{Solve} \left( \text{Gl}; \begin{bmatrix} d_a \\ I \end{bmatrix} \right)$$

$$L_1 = \begin{cases} d_a = \frac{i \cdot m \cdot \sqrt[4]{65536 + 1875 \cdot \pi^3}}{200 \cdot \sqrt[4]{3} \cdot \sqrt[4]{\pi^3}} \\ I = \frac{m^4}{4687500 \cdot \pi^2} \end{cases} \quad L_2 = \begin{cases} d_a = -\frac{m \cdot \sqrt[4]{65536 + 1875 \cdot \pi^3}}{200 \cdot \sqrt[4]{3} \cdot \sqrt[4]{\pi^3}} \\ I = \frac{m^4}{4687500 \cdot \pi^2} \end{cases}$$

$$L_3 = \begin{cases} d_a = -\frac{i \cdot m \cdot \sqrt[4]{65536 + 1875 \cdot \pi^3}}{200 \cdot \sqrt[4]{3} \cdot \sqrt[4]{\pi^3}} \\ I = \frac{m^4}{4687500 \cdot \pi^2} \end{cases} \quad L_4 = \begin{cases} d_a = \frac{m \cdot \sqrt[4]{65536 + 1875 \cdot \pi^3}}{200 \cdot \sqrt[4]{3} \cdot \sqrt[4]{\pi^3}} \\ I = \frac{m^4}{4687500 \cdot \pi^2} \end{cases}$$

Da der Durchmesser  $d_a$  positiv und reell sein muss, kommt nur Lösung 4 in Frage. Diese wird mit `Assign()` zugewiesen und weiterverarbeitet. Der gewählte Durchmesser wird als der nächstgrößere ganzzahlige mm-Wert mit Hilfe der Funktion `Ceil()` bestimmt, die in der Operatorform  $\lceil x \rceil$  dargestellt wird.

$$\text{Assign}(L_4) = \begin{cases} 0,0302 \text{ m} \\ 2,16 \cdot 10^{-8} \text{ m}^4 \end{cases} \quad d_a = 30,2 \text{ mm}$$

$$d_{a, \text{gew}} := \left\lceil \frac{d_a}{\text{mm}} \right\rceil \text{ mm} \quad \boxed{d_{a, \text{gew}} = 31 \text{ mm}}$$

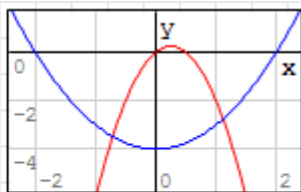
### 6.11.1.3. Beispiel: Schnittpunkt von Kurven

Dieses Beispiel ist zu finden unter [?](#) > Beispiele > Beispiel: Schnittpunkte von Kurven mit `Solve()`.

Die Schnittpunkte zweier Kurven  $f(x)$  und  $g(x)$  sind zu bestimmen:

$$f(x) := x^2 - 4 \quad g(x) := 2 \cdot x - 4 \cdot x^2$$

$$L := \text{Solve}(f(x) - g(x); x)$$

$$L = \begin{cases} x = -\frac{-1 + \sqrt{21}}{5} \\ x = \frac{1 + \sqrt{21}}{5} \end{cases}$$


Die Lösungsgleichungen sollten direkt mit `Assign()` in Definitionen verwendet werden können. Leider scheitert dies und man muss umständlich eine Zuweisung auf die Lösungsvariable zwischenschalten.



```

Assign (■) versagt, wenn es selbst Teil einer Definition ist.

x1 := Assign(L1)    x1 = x = - (-1 + sqrt(21)) / 5
Umgehung des Problems
Assign(L1) = - (-1 + sqrt(21)) / 5    Definition von x anhand von L1
x1 := x = - (-1 + sqrt(21)) / 5    Definition von x1 anhand von x
Clear(x) = 1
Assign(L2) = (1 + sqrt(21)) / 5    Definition von x anhand von L2
x2 := x = (1 + sqrt(21)) / 5    Definition von x2 anhand von x
Clear(x) = 1

```

Man kann auch die Lösungsvariable mit der Substitutionsfunktion umbenennen und dann `Assign()` ausführen. Das ist aber ebenfalls nicht sehr naheliegend:

```

Clear(x; x1; x2) = 1
Assign(L1 | x = x1) = - (-1 + sqrt(21)) / 5    x1 = - (-1 + sqrt(21)) / 5
Assign(L2 | x = x2) = (1 + sqrt(21)) / 5    x2 = (1 + sqrt(21)) / 5

```

Die Variablen  $x_1$  und  $x_2$  werden nun für die grafische Darstellung der Lösung benutzt:

```

(
  explicit (f(x); x; x1 - 1; x2 + 1)
  color = red
  explicit (g(x); x; x1 - 1; x2 + 1)
  p := (
    color = black
    point_type = circle
    points (
      (
        x1 f(x1)
      )
      (
        x2 f(x2)
      )
    )
  )
)

```

Das Diagramm ist übrigens ein `Maxima-Draw2D()`-Bereich (siehe Abschnitt 8.1.4).

### 6.11.2. Numerische Lösung mit Startwerten: `FindRoot()`

Das Plug-in *Non Linear Solvers* stellt eine Reihe von Funktionen zur Gleichungslösung bereit. Die wichtigste davon ist `FindRoot()`, die intern verschiedene spezielle Funktionen ausprobiert, um eine Lösung zu finden. Das Besondere an den Funktionen des Plug-ins ist, dass sie alle mit Maßeinheiten umgehen können, obwohl sie rein numerische Verfahren enthalten.

Die Funktion `FindRoot()` nimmt zwei, drei oder vier Argumente:

1. Eine skalare Gleichung oder ein Gleichungssystem, gegeben als Vektor oder Matrix,
2. eine skalare Variable mit Anfangswert, gegeben als `Variable=Anfangswert` (logisches Gleichheitszeichen) oder ein Vektor oder eine Liste solcher Gleichungen.
3. Option: eine Liste mit Toleranzen für die Erfüllung der Gleichungen. Dabei ist zu beachten, dass die Toleranzen die für die jeweilige Gleichung richtigen Maßeinheiten haben.
4. Zusatzoption: eine Liste mit Toleranzen für die Variablenwerte. Auch hier sind die Werte mit passenden Maßeinheiten anzugeben.

Da abhängig von den Anfangswerten immer nur eine Lösung ermittelt wird, kann diese auch gleich auf die Variablen zugewiesen werden, so dass der Zusatzschritt mit `Assign()` nicht erforderlich ist.

Die Lösung des Beispiels in Abschnitt 6.11.1.2 kann mit `FindRoot()` so aussehen:

Gleichungen

Knicklast Flächenträgheitsmoment der Hohlwelle

$$G1_1 := \left( F \cdot S_K = \frac{\pi^2 \cdot E \cdot I}{l_K^2} \right) \quad G1_2 := \left( I = \frac{\pi}{64} \cdot (d_a^4 - d_i^4) \right)$$

Unknowns (G1) =  $\begin{bmatrix} d_a \\ I \end{bmatrix}$

Clear (d\_a; I) = 1

$$\text{FindRoot} \left( G1; \left\{ \begin{array}{l} d_a = d_i \\ I = d_i^4 \end{array} \right\}; \left\{ \begin{array}{l} 1 \text{ N} \\ 1 \text{ mm}^4 \end{array} \right\} \right) = \left\{ \begin{array}{l} 0,0302 \text{ m} \\ 2,16 \cdot 10^{-8} \text{ m}^4 \end{array} \right.$$

$d_a = 30,2 \text{ mm}$        $d_{a, \text{gew}} := \left\lfloor \frac{d_a}{\text{mm}} \right\rfloor \text{ mm}$        $d_{a, \text{gew}} = 31 \text{ mm}$

Gleichung 1 hat die Dimension einer Kraft, daher wird als Toleranz hier ein Newton vorgegeben. Gleichung 2 hat die Dimension eines Flächenträgheitsmoments, daher ist auch eine entsprechende Toleranz erforderlich.

### 6.11.3. Numerische Lösung mit vorgegebenem Intervall:

#### `Bisection()`

Wenn man sicher ist, dass es in einem bestimmten Intervall genau eine Lösung gibt, kann man auch die Funktion `Bisection()` aus dem Plug-in *Nonlinear Solvers* verwenden. Dabei wird das Halbteilungsverfahren verwendet. Dieses teilt das Intervall in zwei Teile und ermittelt das Vorzeichen, wenn die rechte Gleichungsseite von der linken abgezogen wird. Wechselt das Vorzeichen, muss die Lösung in dem betreffenden Intervall liegen. Auf diese Weise kann man die Lösung immer weiter eingrenzen, bis eine vorgegebene Toleranz erreicht ist.

Die Funktion `Bisection()` kann nur skalare Gleichungen nach einer Variablen auflösen, Gleichungssysteme sind hier also nicht lösbar. Die Argumente sind:

1. Die zu lösende Gleichung,
2. Die Variable, nach der aufzulösen ist und deren Untergrenze in der Form  $\text{Variable}=\text{Untergrenze}$
3. Die Obergrenze der Variable
4. Optional: Toleranz für die Erfüllung der Gleichung.

Das soll wieder am Beispiel aus Abschnitt 6.11.1.2 demonstriert werden. Man muss jetzt allerdings die Gleichungen nacheinander auflösen, da `Bisection()` ja immer nur eine Gleichung lösen kann.

Wir bestimmen zunächst das erforderliche Flächenträgheitsmoment. Als Untergrenze setzen wir Null an, als Obergrenze einen Wert, von dem wir glauben, dass er größer als die tatsächliche Lösung ist.

$$\text{Clear}(d_a; I) = 1$$

$$\text{Bisection}\left(F \cdot S_K = \frac{\pi^2 \cdot E \cdot I}{l_K^2}; I = 0; 10 \text{ cm}^4\right) = 2,16 \text{ cm}^4$$

Wir haben hier auf die Angabe einer Toleranz verzichtet, die Voreinstellung ist  $10^{-4}$  bei Verwendung der Basiseinheit, hier also  $10^{-4}$  N und liefert einen guten Wert für  $I$ .

Bei der Ermittlung des Durchmessers aus dem Trägheitsmoment funktioniert das nicht, der ermittelte Wert ist völlig falsch.

$$\text{Bisection}\left(\left(I = \frac{\pi}{64} \cdot (d_a^4 - d_i^4)\right); d_a = d_i; 10 \cdot d_i\right) = 138 \text{ mm}$$

Mit einer Toleranz von  $10^{-4}I$  bekommen wir ein brauchbares Ergebnis:

$$\text{Clear}(d_a) = 1$$

$$\text{Bisection}\left(\left(I = \frac{\pi}{64} \cdot (d_a^4 - d_i^4)\right); d_a = d_i; 10 \cdot d_i; 0,0001 \cdot I\right) = 30,2 \text{ mm}$$

$$d_a = 30,2 \text{ mm} \quad d_{a, \text{gev}} := \left\lfloor \frac{d_a}{\text{mm}} \right\rfloor \text{ mm} \quad \boxed{d_{a, \text{gev}} = 31 \text{ mm}}$$

## 6.12. Gewöhnliche Differenzialgleichungen

SMath verfügt über keinerlei eingebaute Funktionen zur Lösung von gewöhnlichen Differenzialgleichungen. Allerdings gibt es einige Plug-ins, die numerische Verfahren bereitstellen. Zudem bieten die Plug-ins *MapleWrapper* und *Maxima* symbolische Lösungsfunktionen.

| Funktion                              | Erläuterung  |
|---------------------------------------|--|
| Plug-in <i>AlgLib</i>                 |  |
| <code>al_rkckadapt()</code>           | Adaptives Runge-Kutta-Cash-Karp-Verfahren 4. Ordnung             |
| Plug-in <i>DotNumerics</i>            |  |
| <code>dn_AdamsMoulton()</code>        | Adams-Moulton-Verfahren für steife Systeme                       |
| <code>dn_ExplicitRK45()</code>        | Explizites Runge-Kutta-Verfahren                                 |
| <code>dn_GearsBDF()</code>            | Gears BDF-Verfahren  |
| <code>dn_ImplicitRK45()</code>        | Implizites Runge-Kutta-Verfahren für steife Systeme              |
| Plug-in <i>GNU Scientific Library</i> |  |
| <code>gslrk2()</code>                 | Explizites eingebettetes Runge-Kutta-Verfahren der Ordnung (2,3) |
| <code>gslrk4()</code>                 | Klassisches explizites Runge-Kutta-Verfahren 4. Ordnung.         |
| <code>gslrk8pd()</code>               | Explizites eingebettetes Runge-Kutta Prince-Dormand (8, 9) Verf. |
| <code>glsrkck()</code>                | Explizites eingebettetes Runge-Kutta Cash-Karp (4, 5) Verf.      |
| <code>glsrkf45</code>                 | Explizites eingebettetes Runge-Kutta-Fehlberg (4, 5) Verf.       |
| Plug-in <i>ODE Solver</i>             |  |
| <code>Rkadapt()</code>                | Runge-Kutta-Verfahren mit adaptiver Schrittweite                 |
| <code>rkfixed()</code>                | Runge-Kutta-Verfahren mit fester Schrittweite                    |
| Plug-in <i>Maxima</i>                 |  |
| <code>ODE.2()</code>                  | Symbolischer Löser   |

Tabelle 6.8.: Funktionen zur Lösung gewöhnlicher Differenzialgleichungen

### 6.12.1. Numerische Löser

Alle numerischen Lösungsverfahren lösen Differenzialgleichungssysteme erster Ordnung. Differenzialgleichungen beliebiger Ordnung können immer auf solche Systeme zurückgeführt werden. Dazu führt man für die Ableitungen der abhängigen Variable jeweils extra Variablen ein. Aus der inhomogenen Differenzialgleichung zweiter Ordnung

$$m \frac{d^2 y}{dx^2} + c \frac{dy}{dx} + ky = f(x)$$

wird mit den Ersetzungen  $y_1 = y$  und  $y_2 = dy/dx$  ein System von DGL erster Ordnung:

$$\begin{aligned} \frac{dy_1}{dx} &= y_2 \\ \frac{dy_2}{dx} &= \frac{1}{m} (f(x) - k y_1 - c y_2). \end{aligned}$$

Um solche Systeme zu lösen, sind anzugeben:

- Integrationsbereich für die unabhängige Variable  $x$ , also deren Anfangswert  $x_a$  und Endwert  $x_e$ ,
- Anfangswerte für die abhängigen Variablen  $y_j(x_a)$ ,
- Rechte Seite des Gleichungssystems, also die Ableitung der abhängigen Variablen nach der unabhängigen Variablen, ausgedrückt durch die Werte der abhängigen Variablen und der unabhängigen Variablen

$$\frac{dy_j}{dx} = D_j(x, y_k)$$

- Schrittweite für die Ergebnisberechnung.

Die Ausgabe erfolgt als Matrix, in der jede Zeile für einen Wert der unabhängigen Variable steht und die zugehörigen Werte der abhängigen Variable enthält:

$$L = \begin{bmatrix} x_1 & y_1(x_1) & y_2(x_1) & \dots & y_m(x_1) \\ x_2 & y_1(x_2) & y_2(x_2) & \dots & y_m(x_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n & y_1(x_n) & y_2(x_n) & \dots & y_m(x_n) \end{bmatrix}$$

### Beispiel: Harmonische Schwingungen

Zu lösen ist eine homogene lineare Differenzialgleichung zweiter Ordnung. Sie wird zunächst in ein Gleichungssystem erster Ordnung überführt.

$$u'' + u = 0 \quad \rightarrow \quad \begin{cases} u' = v \\ v' + u = 0 \end{cases}$$

$$y = \begin{bmatrix} u \\ v \end{bmatrix} \quad y' = \begin{bmatrix} v \\ -u \end{bmatrix}$$

Die Ableitungsfunktion hat mindestens zwei Argumente, das erste ist die unabhängige Variable, das zweite ist der Vektor der abhängigen Variablen. Funktionswert ist ein Vektor mit den Ableitungen der abhängigen Variablen.

$$D(x; y) := \begin{bmatrix} y_2 \\ -y_1 \end{bmatrix}$$

Anfangs- und Endwert der unabhängigen Variable sowie Zahl der Schritte

$$x_0 := 0 \quad x_e := 2 \cdot \pi \quad m := 30$$

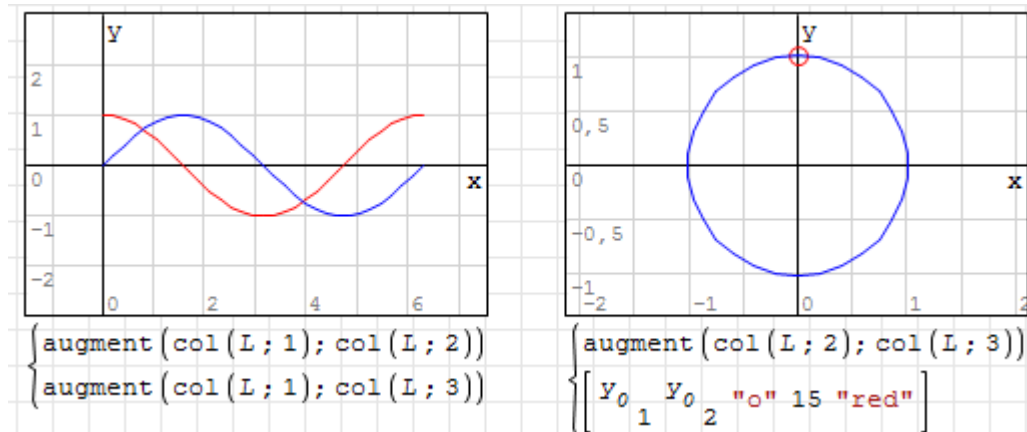
Anfangswerte der abhängigen Variablen:

$$y_0 := \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Aufruf des Runge-Kutta-Lösers mit fester vorgegebener Schrittweite. Einzelne Zeilen der Lösung werden angezeigt.

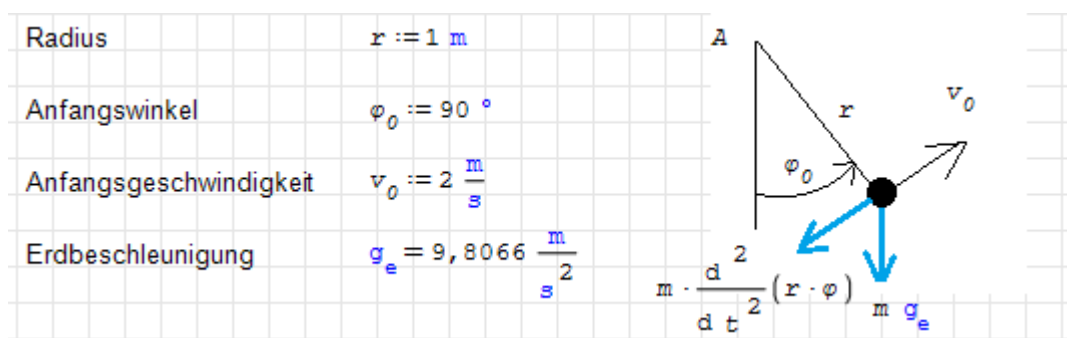
```
L := rkfixed(y_0; x_0; x_e; m; D(x; y))
row(L; 1) = [ 0 0 1 ]
row(L; 10) = [ 1,88 0,951 -0,309 ]
row(L; m+1) = [ 6,28 -9,92·10-5 1 ]
```

Zeitverlauf und Phasendiagramm mit Startpunkt:



### Beispiel: Nichtlineare Pendelschwingungen

Das folgende Beispiel zeigt, wie Differentialgleichungen mit dem Runge-Kutta-Verfahren integriert werden können. Wir betrachten ein mathematisches Pendel (Massenpunkt an starrer, masseloser Stange) mit gegebener Länge  $r$ . Gesucht ist die Bewegungsfunktion  $\varphi(t)$  für die Anfangsgeschwindigkeit  $v_0$  und den Anfangswinkel  $\varphi_0$ .



Die Bewegungsgleichung kann man beispielsweise aus der Drehimpulsbilanz um den Aufhängungspunkt gewinnen (SMath dient hier nur der Dokumentation, die Gleichungen sind nicht aktiv)

$$m \cdot r^2 \cdot \frac{d^2}{dt^2} \varphi = -r g_e \cdot m \cdot \sin(\varphi)$$

$$\frac{d^2}{dt^2} \varphi = -\frac{1}{r} g_e \cdot \sin(\varphi)$$

Das ist eine gewöhnliche Differentialgleichung zweiter Ordnung mit der unabhängigen Variable  $t$  und der abhängigen Variable  $\varphi$ .

Als Lösungsverfahren steht uns das Runge-Kutta-Verfahren zur Verfügung, welches Systeme von Differentialgleichungen erster Ordnung integriert. Wir können z.B. die Funktion `rkfixed()` aus dem Plug-in *ODE Solvers* benutzen (ODE = *Ordinary Differential Equation*, gewöhnliche Differentialgleichung) benutzen.

Dafür müssen wir die Differentialgleichung zweiter Ordnung in ein System von Differentialgleichungen erster Ordnung umwandeln. Das geht immer, indem man für die niedrigeren Ableitungen neue Variable einführt. Im vorliegenden Fall wird die Winkelgeschwindigkeit  $\omega$  als erste Ableitung des Winkels eingeführt. Die zweite Ableitung des Winkels nach der Zeit ist dann gleich der ersten Ableitung der Winkelgeschwindigkeit nach der Zeit.

$$\omega = \frac{d}{dt}\varphi, \quad \frac{d^2}{dt^2}\varphi = \frac{d}{dt}\omega = -\frac{g}{r}\sin(\varphi)$$

Die abhängigen Variablen und ihre ersten Ableitungen werden zu einem Vektor zusammengefasst:

$$\underline{y} = \begin{bmatrix} \varphi \\ \omega \end{bmatrix}, \quad \underline{y}' = \begin{bmatrix} y_2 \\ -\frac{g}{r}\sin(y_1) \end{bmatrix}.$$

Der Ableitungsvektor muss als Funktion der unabhängigen und der abhängigen Variable bereitgestellt werden (in unserem Fall kommt darin die unabhängige Variable  $t$  nicht vor).

$$D(t; \underline{y}) := \begin{bmatrix} y_2 \\ -\frac{g}{r}\sin(y_1) \end{bmatrix}$$

Außerdem wird der Vektor der Anfangsbedingungen benötigt:

$$\underline{y}_0 := \begin{bmatrix} \varphi_0 \\ \frac{v_0}{r} \end{bmatrix}$$

Die Funktion `rkfixed()` integriert die abhängigen Variablen über einen Zeitraum von  $t_0$  bis  $t_e$  unterteilt in  $n$  gleichgroße Schritte.

Zurückgeliefert wird eine Matrix mit den Werten der unabhängigen Variable in der ersten Spalte und den abhängigen Variablenwerten in den weiteren Spalten. Jede Zeile entspricht einem Zeitpunkt. Maßeinheiten sind im Ergebnis nicht mehr enthalten, dies ist ein Mangel des verwendeten Plug-ins. Die Endzeit  $t_e$  wird so bestimmt (durch Ausprobieren), dass sich in der Animations-Endlosschleife ein flüssiger Bewegungsablauf ergibt.

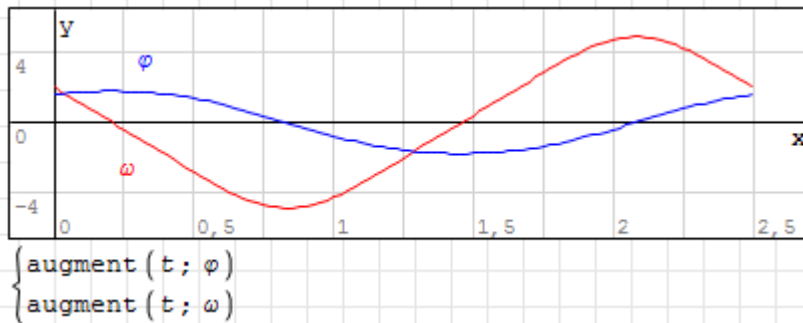
```
t_0 := 0 s      t_e := 2,49 s      n := 100
L := rkfixed(y_0; t_0; t_e; n; D(t; y))
row(L; 101) = [ 2,49 1,5649 2,0289 ]
```

Die Ergebnisgrößen werden als Vektoren extrahiert und können dann grafisch dargestellt werden.

```

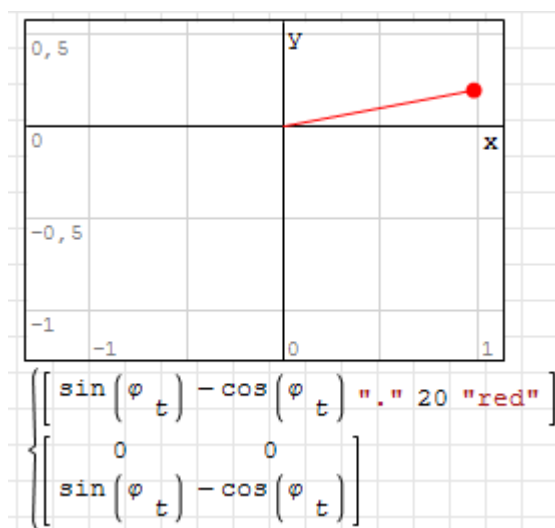
t := col(L; 1)      unabhängige Variable
φ := col(L; 2)     abhängige Variable 1
ω := col(L; 3)     abhängige Variable 2
k := [1..(rows(L)-1)]  Indexvektor für die Animation

```

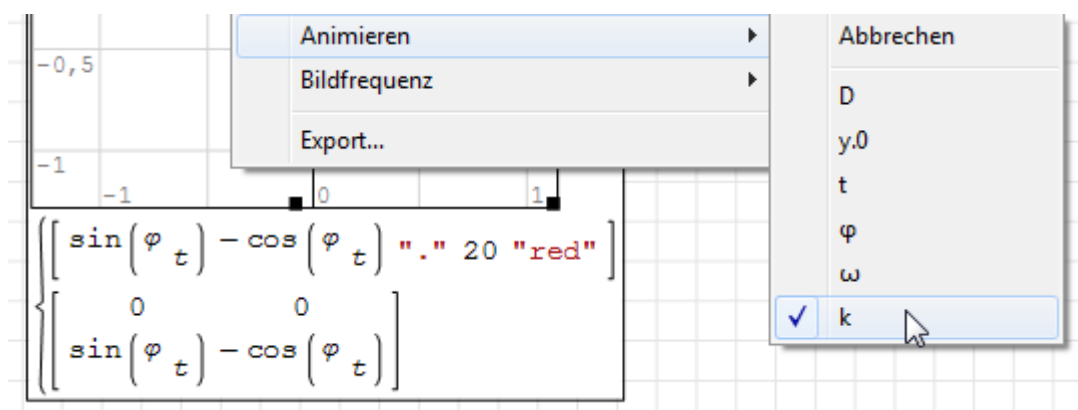


Besonders interessant ist die Möglichkeit, die Bewegung des Pendels animiert darzustellen. Die Koordinaten des Massenpunkts ergeben sich aus dem Winkel  $\varphi(t)$  und dem Radius  $r$ :

$$x = \sin \varphi, \quad y = -\cos \varphi$$



Die Animation wird erzeugt, indem im Kontextmenü (rechte Maustaste) der oben erzeugte Indexvektor  $k$  als Wertebereich für  $t$  ausgewählt wird.  $t$  entspricht hier nicht der Zeit, sondern einem Zähler, der die berechneten Datensätze durchläuft.





## 6.13. Lorentz-Attraktor

Dieses Beispiel ist zu finden unter  > **Beispiele** > **GDGL: Lorentz-Attraktor**. Es zeigt:

- die Integration von Systemen gewöhnlicher Differenzialgleichungen
- die Verwendung des X-Y Plot-Bereichs

Attraktoren sind Gebiete im Phasenraum dynamischer Systeme, die auf die Umgebung anziehend wirken. Der Lorentz-Attraktor ist der Attraktor des folgenden Differenzialgleichungssystems:

$$\begin{aligned}\dot{x} &= a(y - x) \\ \dot{y} &= x(b - z) - y \\ \dot{z} &= xy - cz\end{aligned}$$

Dieses System kann mit dem Runge-Kutta-Verfahren integriert werden, geeignet ist dafür z.B. die Funktion `Rkadapt()` aus dem Plug-in *ODE Solvers*.

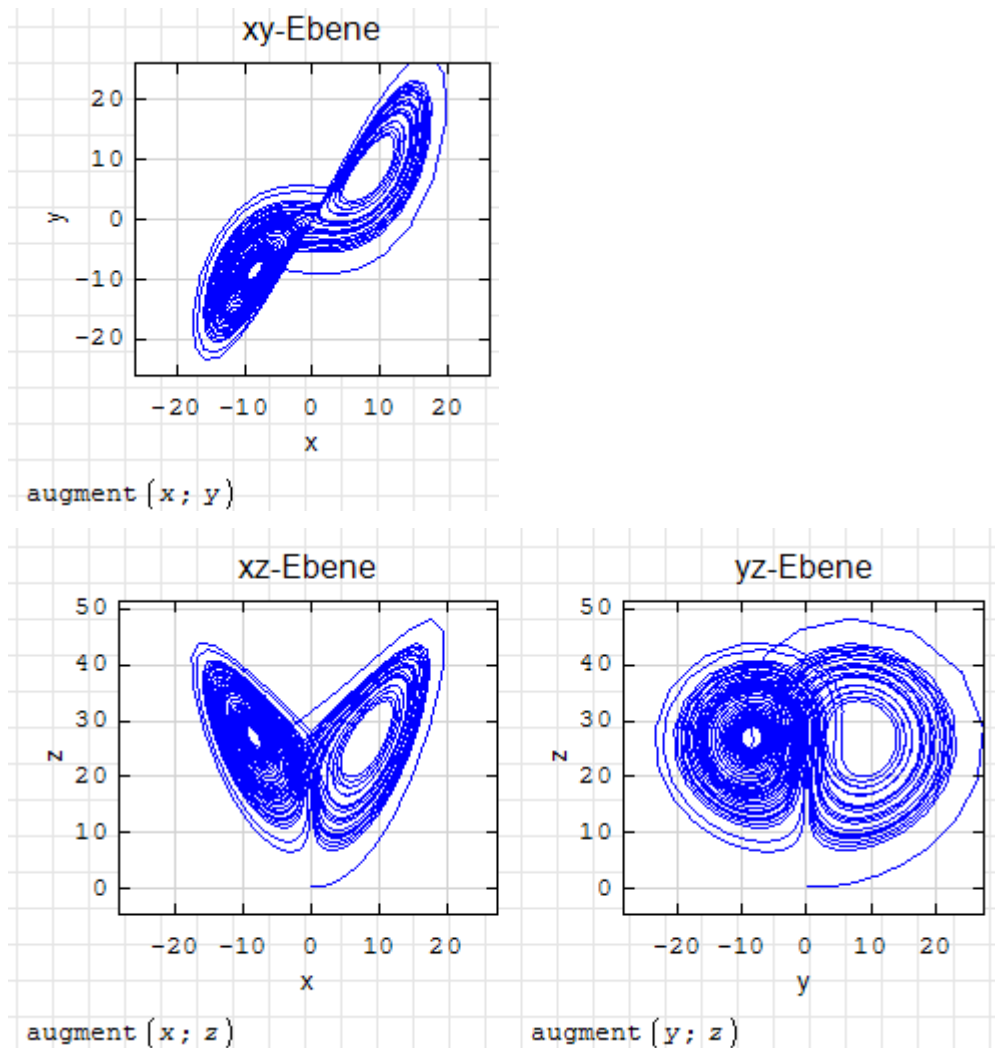
Im Beispiel ist  $t$  die unabhängige Variable, die von 0 bis 50 Sekunden läuft. Auszugeben sind 2000 Zeitschritte. Die abhängigen Variablen  $x, y, z$  sind Komponenten des Vektors  $\mathbf{x}$ , dessen Anfangswert ist  $\mathbf{x}_0$ .

$$D(t; \mathbf{x}) := \begin{bmatrix} a \cdot (x_2 - x_1) \\ x_1 \cdot (b - x_3) - x_2 \\ x_1 \cdot x_2 - c \cdot x_3 \end{bmatrix} \quad \mathbf{x}_0 := \begin{bmatrix} 0,1 \\ 0,1 \\ 0,5 \end{bmatrix} \quad t_0 := 0 \quad t_{\max} := 50 \quad n := 2000$$

Im folgenden Abschnitt werden die Systemparameter angegeben, das Integrationsverfahren aufgerufen und die Spalten der Ergebnismatrix extrahiert.

$$\begin{aligned}a &:= 10 & b &:= 28 & c &:= \frac{8}{3} & & \text{Zeitnahme} \\ & & & & & & & t_0 := \text{time}(1) \\ L &:= \text{Rkadapt}(x_0; t_0; t_{\max}; n; D(t; \mathbf{x})) & & & & & & \text{time}(1) - t_0 = 1,579 \text{ s} \\ x &:= \text{col}(L; 2) & y &:= \text{col}(L; 3) & z &:= \text{col}(L; 4) & & \end{aligned}$$

Die  $xy$ -,  $xz$ - und  $yz$ - Phasenebenen werden mit dem X-Y Plot-Bereich aus den Plug-in *X-Y Plot Region* dargestellt.



### 6.13.1. Löser aus dem Maxima-Plug-in

Bemerkenswerte Erweiterungen für Differenzialgleichungen durch das Plug-in *Maxima* sind:

- die Funktion `ODE.2()` zur symbolischen Lösung gewöhnlicher DGL bis zweiter Ordnung,
- die Maxima-Funktionen zur Laplace-Transformation, ein Beispiel dazu findet man unter [?](#) > **Beispiele** > **Beispiel: Laplace-Transformation**.
- die Richtungsfeld-Diagrammfunktionen des Pakets *drawdf* (draw direction field). Dokumentation dazu findet sich unter [?](#) > **Grafik** > **Diagramme mit Maxima** > **Beispiele für Richtungsfeld-Diagramme**.

#### Funktion `ODE.2()`

Für die Lösung von Differenzialgleichungen bis zu zweiter Ordnung steht die Funktion `ODE.2()` zur Verfügung. Anzugeben sind die Differenzialgleichung, die unbekannte Funktion und die unabhängige Variable.

$$L := \text{ODE}_2 \left( \frac{d^2}{dt^2} v(t) + \omega^2 \cdot v(t) = 0; v(t); t \right)$$

$$L = \begin{cases} \text{"}\omega \text{ is assumed to be positive.} \\ v(t) = k1 \cdot \sin(t \cdot \omega) + k2 \cdot \cos(t \cdot \omega) \end{cases}$$

Die Meldung „ $\omega$  is assumed to be positive“ bedeutet, dass Maxima intern nach dem Vorzeichen von  $\omega$  gefragt hat. Solche Fragen werden vom Plug-in abgefangen und immer mit + beantwortet. Damit man als Anwender eine Chance hat, dies zu erkennen, wird die Meldung in die Lösung integriert, was aber die Weiterverarbeitung nicht behindert. Man kann die Rückfrage durch Mitteilung einer Annahme vermeiden:

$$\text{assume}(\omega > 0) = \{\omega > 0\}$$

$$L := \text{ODE}_2 \left( \frac{d^2}{dt^2} v(t) + \omega^2 \cdot v(t) = 0; v(t); t \right)$$

$$L = v(t) = k1 \cdot \sin(t \cdot \omega) + k2 \cdot \cos(t \cdot \omega)$$

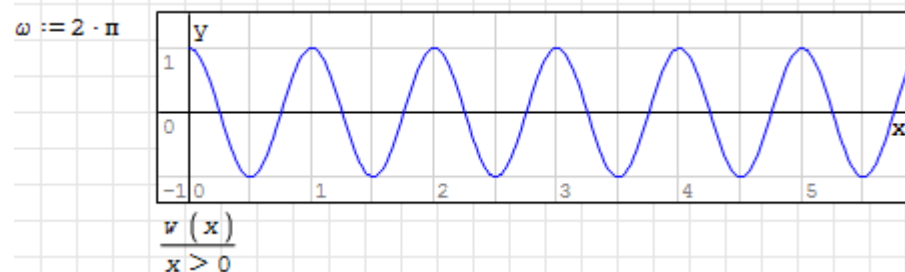
Zurückgegeben wird eine boolesche Gleichung, die mit `Assign()` zur Definition der unbekanntenen Funktion verwendet werden kann. Die Lösung enthält abhängig von der Ordnung der DGL eine oder zwei Integrationskonstanten,  $c$  (erste Ordnung) oder  $k1$  und  $k2$  (zweite Ordnung).

$$\text{Assign}(L) = k1 \cdot \sin(t \cdot \omega) + k2 \cdot \cos(t \cdot \omega)$$

$$v(t) = k1 \cdot \sin(t \cdot \omega) + k2 \cdot \cos(t \cdot \omega)$$

Die Konstanten können aus den Anfangsbedingungen bestimmt werden, hier z.B. wird die Auslenkung zu Eins gesetzt und die Geschwindigkeit gleich Null.

$$\text{Assign} \left( \text{Solve} \left( \left( \begin{cases} v(0) = 1 \\ \left. \frac{d}{dt} v(t) \right|_{t=0} = 0 \end{cases}; \begin{cases} k1 \\ k2 \end{cases} \right) \right) = \begin{cases} 0 \\ 1 \end{cases} \quad v(t) = \cos(t \cdot \omega)$$



## 6.14. Datenimport und -export

Standardmäßig steht nur die Funktion `importData()` zur Verfügung, die es erlaubt, Zahlen oder SMATH-Ausdrücke aus Textdateien zu lesen. Dabei können das Dezimaltrennzeichen und das Spaltentrennzeichen sowie Start- und End-Zeilen und -Spalten vorgegeben werden.

Die Plug-ins *Data Exchange* und *XLSX Import/Export* bieten die Funktionen `ImportData.ODS()` und `ImportData.XLSX()`. ODS ist das Dateiformat von OpenOffice Calc, XLSX ist das Format von MS Excel.

Das Plug-in *Mathcad File Access* bietet Funktionen für das Einlesen von Grafikdateien sowie für das Lesen und Schreiben von Audio- und Binärdateien.

Das gerade in der Entwicklung befindliche Plug-in **SQLite Interface** bietet Zugriff auf **SQLite relationale Datenbanken**. Es wird im vorliegenden Handbuch noch nicht berücksichtigt.

Ebenfalls neu ist das Plug-in **Excel IO**, welches aus SMath heraus mit Excel kommunizieren kann. Das beinhaltet

- Schreiben von Daten
- Lesen von Daten
- Erzeugen von Excel-Screenshots für die Anzeige im Image-Bereich in SMath

Dieses Plug-in wird im vorliegenden Handbuch ebenfalls noch nicht berücksichtigt.

Dateinamen können mit relativen oder absoluten Pfaden versehen sein. Beachten Sie, dass das aktuelle Verzeichnis in SMath nicht unbedingt das Verzeichnis ist, in dem das Rechenblatt gespeichert ist.

Stellen Sie dies notfalls explizit sicher, indem Sie das aktuelle Verzeichnis auf das Dokumentverzeichnis setzen.

```
dir := CurrentDirectory(DocumentDirectory(""))  
dir = "C:\FHB\Software\SMath\SMath Skript\Bilder\"
```

Die Regeln für das aktuelle Verzeichnis beschreibt Abschnitt 4.5.

Achten Sie beim Einlesen von größeren Datenmengen darauf, die Optimierung des Formelbereichs auf „Numerisch“ zu stellen (**Kontextmenü** > **Optimierung** > **Numerisch**). Durch die standardmäßige symbolische Verarbeitung kann die Zeit für das Einlesen extrem ansteigen.

### 6.14.1. Funktionsübersicht

| Format                     | Endung   | Importfunktion             |
|----------------------------|----------|----------------------------|
| RGB Grauwert               | png,...  | READ_IMAGE()               |
| RGB roter Kanal            | png,...  | READ_RED()                 |
| RGB grüner Kanal           | png,...  | READ_GREEN()               |
| RGB blauer Kanal           | png,...  | READ_BLUE()                |
| RGB drei Kanäle            | png,...  | READRGB()                  |
| BMP Grauwert               | bmp      | READ_BMP()                 |
| Audiodaten                 | wav      | READWAV()                  |
| Binärdateien<br>(Matrizen) | beliebig | READBIN()                  |
| Textdatei, auch CSV        | beliebig | importData() <b>6.14.2</b> |
| Open Document<br>Tabelle   | .ods     | importData.ODS()           |
| Excel Tabelle              | .xlsx    | importData.XLSX()          |

Tabelle 6.9.: Funktionen zum Einlesen von Daten

Die folgende Tabelle listet die Möglichkeiten zum Rausschreiben von Daten auf.

| Format                          | Endung   | Exportfunktion    |
|---------------------------------|----------|-------------------|
| Text mit<br>Spaltentrennzeichen | csv      | exportData.CSV()  |
| MS Excel                        | xlsx     | exportData.XLSX() |
| OpenOffice Calc                 | ods      | exportData.ODS()  |
| OpenOffice Formel               | odf      | exportData.ODF()  |
| Audiodatei                      | wav      | WRITEWAV()        |
| Binärdatei (Matrizen)           | beliebig | WRITEBIN()        |

Tabelle 6.10.: Funktionen zum Export von Daten und Formeln

### 6.14.2. Textdateien lesen mit importData()

Die Funktion importData() liest Textdateien. Es gibt zwei Formen:

```
importData(name)
importData(name; ddec; darg; dcol; r1; r2; c1; c2; fsym)
```

Die Funktion liest aus der Datei name Werte mit dem Dezimaltrennzeichen ddec, dem Argumenttrennzeichen darg und dem Spaltentrennzeichen dcol. Dabei werden Zeilen r1

bis r2 und Spalten c1 bis c2 gelesen. Falls `fsym=1`, werden die Werte als symbolische SMath-Ausdrücke betrachtet.

Beispieldatei: `data.txt`

```
1 "abc" 2,2
2 "cde" 3,3
3 "efg" 4,4
"dfr" 2 3
```

Mit `fsym=1` können Zahlen und Text gelesen

```
D := importData ("data.txt"; ", "; ";"; " "; 0; 0; 1; 3; 1)
D = [
  1  "abc" 2,2
  2  "cde" 3,3
  3  "efg" 4,4
  "dfr" 2  3
]
```

Mit `fsym=0` (letztes Argument) können nur Zahlenwerte gelesen werden, z.B. die dritte Spalte der Datei:

```
D := importData ("data.txt"; ", "; ";"; " "; 0; 0; 3; 3; 0)
D = [
  2,2
  3,3
  4,4
  3
]
```

### 6.14.3. XLSX- und ODS-Tabellen

Die Exportfunktionen `exportData.XLSX()` und `exportData.ODS()` schreiben den Inhalt des ersten Arguments in eine Datei des passenden Typs. Typischerweise ist dieses Argument eine Matrix, die je nach Inhalt der Einträge unterschiedliche Exportergebnisse erzeugt:

- Zahlenkonstanten und Zeichenketten (z.B. 4,5, "ein Beispiel") werden als numerische bzw. Textzelle direkt übernommen.
- Ausdrücke werden zunächst numerisch ausgewertet. Ist das Ergebnis eine Zahl, wird diese exportiert (z.B. 2,0943951023932).
- Kann das Objekt nicht zu einer Zahl oder Zeichenkette ausgewertet werden, wird die interne symbolische Darstellung des Ausdrucks als Formel exportiert (z.B. `mat(1,2,2,1)`).

Die Importfunktionen `importData.XLSX()` und `importData.ODS()` lesen ein anzugebendes Blatt aus einer anzugebenden Datei. Optional kann eine Zelladresse oder ein Adressbereich angegeben werden.

Die Adressen sind als Zeichenkette aus Spalten-Kennbuchstabe und Zeilennummer anzugeben, also z.B. "A1", "B10".

Die Zellen werden je nach Inhalt unterschiedlich importiert:

- Zahlenkonstanten werden als solche übernommen.

- Textzellen werden als Zeichenketten importiert.
- numerisch auswertbare Formelzellen werden als Zahlen importiert.
- Formelzellen der Form `=mat(1,2,2,1)` werden als symbolische Ausdrücke importiert.

### Symbolische SMath-Ausdrücke in Rechentabellen

Symbolische Ausdrücke werden in ODS- oder XLSX-Dateien als Formelzelle in der Form `=mat(1,2,2,1)` gespeichert oder erwartet, also mit Gleichheitszeichen und eingeschlossen in Anführungszeichen.

Dabei werden die Standardeinstellungen für Dezimaltrenner (Punkt) und Argumenttrenner (Komma) verwendet. Die in diesem Handbuch empfohlene Einstellung (Komma und Semikolon) wird ignoriert.

In symbolischen Ausdrücken enthaltene Zeichenketten werden durch doppelte Hochkommas (') anstelle von Gänsefüßchen eingeschlossen, da letztere bereits den Ausdruck selbst markieren.

## 6.15. Statistik und Datenanalyse

Hier geht es um

- deskriptive Statistik, also die Berechnung von statistischen Merkmalen gegebener Daten,
- Wahrscheinlichkeitsverteilungen und Spezialfunktionen für statistische Modelle,
- Regression, also die Anpassung statistischer Modelle an gegebene Daten,
- Interpolation ein- und zweidimensionaler Datenfelder.

Daten liegen in der Regel als Matrizen oder Vektoren vor, der Umgang mit solchen Objekten, einschließlich Import und Export, wird in Abschnitt 6.8 besprochen.

### 6.15.1. Deskriptive Statistik

| Funktion       | Bedeutung                 |
|----------------|---------------------------|
| HarmonicMean() | Harmonischer Mittelwert   |
| Max()          | Maximum                   |
| Mean()         | Arithmetischer Mittelwert |
| Median()       | Median                    |
| Min()          | Minimum                   |
| Mode()         | Modalwert                 |
| Moment()       | Moment                    |
| Skewness()     | Schiefe                   |
| Slope()        | Geradensteigung           |
| StdDev()       | Standardabweichung        |
| Variance()     | Varianz                   |
| WeightedMean() | Gewichteter Mittelwert    |

Tabelle 6.11.: Funktionen für die statistische Beschreibung von Daten (bereitgestellt durch das Plug-in *Statistical Tools*).

### 6.15.2. Spezielle Funktionen

Die folgenden speziellen Funktionen werden für die Berechnung statistischer Verteilungen benötigt und können für eigene statistische Modelle verwendet werden.

| Name                 | Bedeutung  |
|----------------------|--|
| Beta()               | Beta-Funktion, Eulersches Integral 1. Art        |
| BetaRegularized()    | Regularisierte unvollständige Beta-Funktion      |
| Dirac()              | Dirac-Funktion                                   |
| erf()                | Fehlerfunktion                                   |
| erfc()               | Komplementäre Fehlerfunktion                     |
| erfinv()             | Inverse Fehlerfunktion                           |
| Gamma()              | Gamma-Funktion                                   |
| GammaRegularized.P() | Regularisierte Gamma-Funktion der oberen Grenze  |
| GammaRegularized.Q() | Regularisierte Gamma-Funktion der unteren Grenze |
| Heaviside()          | Sprungfunktion                                   |
| Heaviside.D()        | unsymmetrische Sprungfunktion                    |

Abbildung 6.4.: Spezielle Funktionen aus dem Plug-in *Statistical Tools*



### 6.15.3. Verteilungen

Verteilungen werden in der Regel mit jeweils drei Funktionen unterstützt, die mit den Namen CDF, pdf oder ICDF beginnen und den Verteilungsnamen als Index tragen:

- **CDF** (*cumulative density function*) Verteilungsfunktion oder kumulative Dichtefunktion, sie gibt die Wahrscheinlichkeit  $p$  an, gemäß der Verteilung einen Wert kleiner gleich  $x$  zu finden.
- **pdf** (*probability density function*) Häufigkeitsfunktion oder Wahrscheinlichkeitsdichtefunktion.
- **ICDF** (*inverse cumulative density function*) Quantilfunktion oder inverse Verteilungsfunktion, sie gibt an, bei welchem  $x$ -Wert die Verteilungsfunktion einen bestimmten Wahrscheinlichkeitswert  $p$  erreicht hat.

| Verteilung                          | Name                          | CDF | ICDF | pdf |
|-------------------------------------|-------------------------------|-----|------|-----|
| Binomialverteilung                  | <code>Binomial</code>         | x   | x    | x   |
| Cauchy-Verteilung                   | <code>Cauchy</code>           | x   | x    | x   |
| $\chi^2$ -Verteilung                | <code>ChiSquare</code>        | x   |      | x   |
| Exponentialverteilung               | <code>Exponential</code>      | x   | x    | x   |
| Fisher-Verteilung                   | <code>F</code>                | x   |      | x   |
| Geometrische Verteilung             | <code>Geometric</code>        | x   | x    | x   |
| Verschobene geometrische Verteilung | <code>GeometricShifted</code> | x   | x    | x   |
| Normalverteilung                    | <code>Normal</code>           | x   | x    | x   |
| Poisson-Verteilung                  | <code>Poisson</code>          | x   | x    | x   |
| Rayleigh-Verteilung                 | <code>Rayleigh</code>         | x   | x    | x   |
| Studentsche t-Verteilung            | <code>t</code>                | x   |      | x   |
| Gleichverteilung                    | <code>Uniform</code>          | x   | x    | x   |
| diskrete Gleichverteilung           | <code>UniformDiscrete</code>  | x   | x    | x   |
| Weibullverteilung                   | <code>Weibull</code>          | x   | x    | x   |

Tabelle 6.12.: Statistische Verteilungen mit den zugehörigen Verteilungs- (CDF), Quantil- (ICDF) und Häufigkeitsfunktionen (pdf) aus dem Plug-in *Statistical Tools*.

### 6.15.4. Multilineare Regression

Dieses Beispiel ist der **SMath-Einführung**<sup>5</sup> von Bernard V. Liengme (Kanada) entnommen. Die Aufgabe besteht darin, die Koeffizienten einer Funktion

$$f(x) = c_1 f_1(x) + c_2 f_2(x) + \dots + c_n f_n(x)$$

so zu bestimmen, dass ein Datensatz aus  $m$  zugeordneten  $x$ - und  $y$ -Werten möglichst gut dargestellt wird.

<sup>5</sup><http://people.stfx.ca/bliengme/SMath/SMathPrimer.pdf>

Eine Möglichkeit dafür ist die Methode der kleinsten Fehlerquadrate, die den mittleren quadratischen (vertikalen) Abstand zwischen den Funktionswerten  $f(x)$  und den Datenwerten  $y(x)$  minimiert. Die Funktion kann als Skalarprodukt des Koeffizientenvektors  $\underline{c}$  mit dem Funktionsvektor  $\underline{f}$  dargestellt werden:

$$f(x) = [c_1 \ c_2 \ \cdots \ c_n] \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix} = \underline{c}^\top \underline{f}$$

Man baut nun eine Matrix  $\underline{X}$  auf, in der die Spalten durch Anwendung der Funktionen  $f_i$  auf den ganzen Datenvektor  $\underline{x}$  gebildet werden.

$$\underline{X} = \begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_n(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_m) & f_2(x_m) & \cdots & f_n(x_m) \end{bmatrix}$$

Die Koeffizienten des Modells berechnen sich dann mit der Pseudoinversen von  $\underline{X}$

$$\underline{c} = (\underline{X}^\top \underline{X})^{-1} \underline{X}^\top \underline{y}$$

Diese Berechnung soll nun in SMath ausgeführt werden. Zunächst die Definition der Daten:

$$\underline{x} := \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \quad \underline{y} := \begin{bmatrix} 0,9 \\ 2,4 \\ 7,0 \\ 17,4 \\ 30,5 \\ 48,3 \end{bmatrix} \quad j := [1..rows(\underline{x})]$$

Es soll der Ansatz  $f(x) = c_1 + c_2x + c_3x^3$  verwendet werden. Damit ist  $f_1(x) = 1$ ,  $f_2(x) = x$  und  $f_3(x) = x^3$ . Dann geben wir den Vektor  $\underline{f}$  vor und berechnen die Matrix  $\underline{X}$ :

$$\underline{f}(\underline{x}) := \begin{bmatrix} 1 \\ \underline{x} \\ \underline{x}^3 \end{bmatrix} \quad k := [1..rows(\underline{f}(\underline{x}))]$$

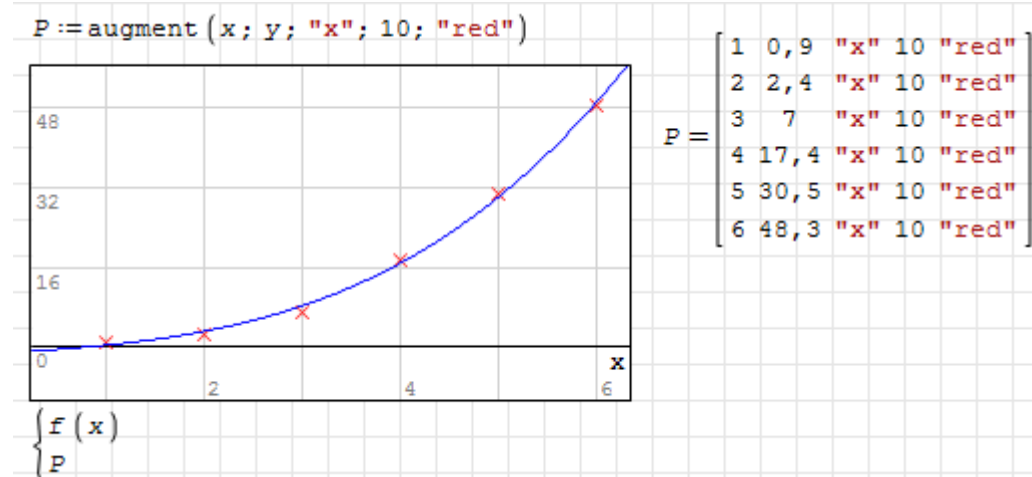
$$\underline{X}_{j \ k} := \underline{f}(\underline{x}_j)_k \quad \underline{X} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 8 \\ 1 & 3 & 27 \\ 1 & 4 & 64 \\ 1 & 5 & 125 \\ 1 & 6 & 216 \end{bmatrix}$$

Nun berechnen wir die Koeffizienten und bilden daraus die Modellfunktion  $f(x)$  als Skalarprodukt aus dem Koeffizientenvektor und dem Funktionsvektor. Alternativ kann man auch die entsprechende Funktion aus dem Plug-in *Dot Numerics* verwenden:

$$c := (X^T \cdot X)^{-1} \cdot X^T \cdot y = \begin{bmatrix} -1,36 \\ 1,40 \\ 0,193 \end{bmatrix} \quad f(x) := c \cdot f(x)$$

$$\text{dn\_LinAlgLLS\_SVD}(X; y) = \begin{bmatrix} -1,36 \\ 1,4 \\ 0,193 \end{bmatrix}$$

Nun soll noch die so bestimmte Funktion mit den Datenpunkten verglichen werden.



### 6.15.5. Nichtlineare Regression

Für nichtlineare Regressionsprobleme kann das Maxima-Paket *lsquares* benutzt werden. Die Funktion `Fit()` des Maxima-Plug-ins bildet hierfür eine bequeme Schnittstelle.

`Fit(data; vars; eqn; pars; init)`

`data` (Matrix) gegebene Daten. Jede Zeile ist ein Datensatz, jede Spalte eine Variable.

`vars` (Liste) Variablennamen (für jede Spalte in `data` ein Eintrag).

`eqn` Gleichung, verknüpft die in `vars` aufgelisteten Größen und die in `pars` aufgelisteten Ausgleichsparameter.

`pars` (Liste) Namen der freien Parameter in der Gleichung `eqn`.

`init` (Liste) der Startwerte für die Suche nach den optimalen Parametern.

Rückgabewert ist eine Liste mit Gleichungen der Form `Parameter=Wert`. Diese kann genau wie bei `Solve()` mit `Assign()` für die Zuweisung der Werte an die Parameter benutzt werden.

Zur Beurteilung der Qualität des Abgleichs gibt es die Funktionen `MSE()` und `Residuals()`, die den mittleren quadratischen Fehler bzw. die Abweichung an den Datenpunkten liefern

**Beispiel 1:**

$$\text{Data} := \begin{bmatrix} 1 & 1,5 & 2,25 & 3 & 2 \\ 1 & 1 & 2 & 2 & 2 \\ 1 & 2 & 1 & 2 & 1 \end{bmatrix}^T$$

$$L := \text{Fit} \left( \text{Data}; \left\{ \begin{array}{l} z \\ x; (z + D)^2 = (C + A \cdot x + B \cdot y) \\ y \end{array} \right. \right); \left\{ \begin{array}{l} A \\ B \\ C \\ D \end{array} \right\}; \left\{ \begin{array}{l} 0 \\ 0 \\ 0 \\ 0 \end{array} \right\}$$

Daten und Spaltennamen
Ansatz
Parameternamen und Startwerte

Die Lösung besteht aus einer Liste von Gleichungen. Die Funktion `MSE()` kann die Parameterwerte aus einer solchen Liste

entnehmen:

$$L = \left\{ \begin{array}{l} A = -3,628998943824585 \\ B = -1,65262581323713 \\ C = 10,45496853509095 \\ D = -3,319624288317968 \end{array} \right. \quad \text{Optimierung abgeschaltet}$$

$$\text{MSE} \left( \text{Data}; \left\{ \begin{array}{l} z \\ x; (z + D)^2 = (C + A \cdot x + B \cdot y) \\ y \end{array} \right. \right); L = 0,0661$$

Wenn die Lösung mit `Assign()` auf die Variablen zugewiesen wird, kann die verkürzte Form von `MSE()` benutzt werden.

$$\text{Assign}(L) = \left\{ \begin{array}{l} -3,63 \\ -1,65 \\ 10,5 \\ -3,32 \end{array} \right.$$

$$\text{MSE} \left( \text{Data}; \left\{ \begin{array}{l} z \\ x; (z + D)^2 = (C + A \cdot x + B \cdot y) \\ y \end{array} \right. \right) = 0,0661$$

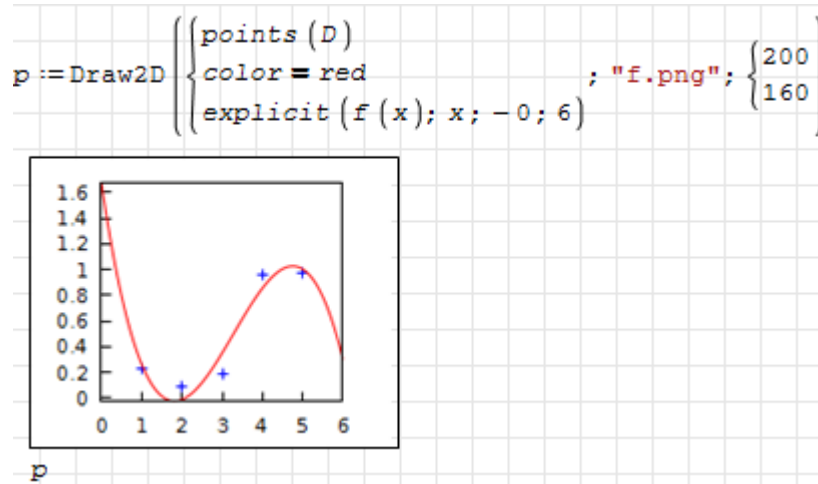
**Beispiel 2:**

$$D := \text{augment} \left( [1..5]; \text{Random}(5) \right) = \begin{bmatrix} 1 & 0,237 \\ 2 & 0,0942 \\ 3 & 0,19 \\ 4 & 0,96 \\ 5 & 0,976 \end{bmatrix}$$

$$f(x) := a \cdot x^3 + b \cdot x^2 + c \cdot x + d \quad D = \begin{bmatrix} 1 & 0,237 \\ 2 & 0,0942 \\ 3 & 0,19 \\ 4 & 0,96 \\ 5 & 0,976 \end{bmatrix}$$

$$\text{Assign} \left( \text{Fit} \left( D; \left\{ \begin{array}{l} x \\ y; y = f(x) \end{array} \right. \right); \left\{ \begin{array}{l} a \\ b \\ c \\ d \end{array} \right\}; \left\{ \begin{array}{l} 1, \\ 3, \\ 1, \\ 1, \end{array} \right\} \right) = \left\{ \begin{array}{l} -0,0828 \\ 0,816 \\ -2,14 \\ 1,67 \end{array} \right.$$

Das Diagramm wird mit der Maxima-Funktion `Draw2D()` erzeugt und mit einem Image-Bereich angezeigt.



**Beispiel 3:** Aus der Werkstofftechnik, Buchmayr (2002): Abhängig von der Abkühlgeschwindigkeit  $t_{85}$  von  $800^\circ\text{C}$  auf  $500^\circ\text{C}$  kommt es in der Wärmeeinflusszone einer Schweißnaht zu einer Aufhärtung.

In der Datei *HV\_WEZ.prn* stehen Messwerte für  $t_{85}$  (Spalte 1, in s) und für die Vickers-Härte HV (Spalte 2).

|   |     |
|---|-----|
| 1 | 440 |
| 2 | 410 |
| 3 | 370 |

...

An diese Daten soll ein Ansatz der Form

$$HV(t_{85}) = a_1 + a_2 \arctan(a_3 \ln t_{85} + a_4)$$

angepasst werden.

Zunächst werden die Daten eingelesen:

```
dir := CurrentDirectory ( DocumentDirectory ( "" )
Daten := importData ( "HV_WEZ.prn" )
Daten = [ 1 440
          2 410
          3 370
          ⋮ ]
```

Nun werden die Ansatzfunktion definiert und der Parameterabgleich durchgeführt:

$$HV(t) := a_1 + a_2 \cdot \text{atan}(a_3 \cdot \ln(t) + a_4)$$

$$\text{ans} := \text{Fit} \left( \text{Daten}; \left\{ \begin{array}{l} t \\ h \end{array} \right\}; h = HV(t); \left\{ \begin{array}{l} a_1 \\ a_2 \\ a_3 \\ a_4 \end{array} \right\}; \left\{ \begin{array}{l} 300 \\ -150 \\ 4 \\ 0 \end{array} \right\} \right) = \left\{ \begin{array}{l} a_1 = 328,9888421056474 \\ a_2 = -116,9161889992861 \\ a_3 = 0,9557815582826997 \\ a_4 = -1,438413063418649 \end{array} \right.$$

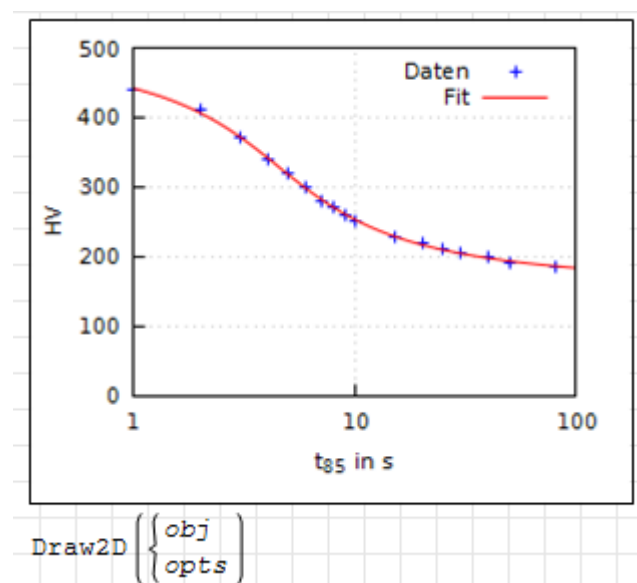
$$\text{Assign}(\text{ans}) = \begin{cases} 329 \\ -117 \\ 0,956 \\ -1,44 \end{cases}$$

Die Qualität des Abgleichs soll mit einem Diagramm mit logarithmischer  $x$ -Achse dargestellt werden.

Die Funktion `Draw2D()` verarbeitet beliebig geschachtelte Listen mit Grafikobjekten und -optionen. Hier sind die Befehle geordnet nach Grafikobjekten (`obj`) und Diagrammsteuerung (`opts`).

```
obj := {
  key = "Daten"
  points (Daten)
  key = "Fit"
  color = red
  explicit (HV(t); t; 1; 100)
}
opts := {
  logx = true
  yrange = {
    0
    500
  }
  grid = true
  xlabel = "t_{85} in s"
  ylabel = "HV"
}
```

Der Grafikbefehl `Draw2D()` erzeugt eine Grafikdatei, die mit einem Image-Bereich (**Einfügen** > **Image-Bereich**) angezeigt wird.



### 6.15.6. Interpolation

SMATH bietet eingebaute Funktionen für die eindimensionale Interpolation:

- `ainterp(vx; vy; x)` kubische Spline-Interpolation der Vektoren **X-Daten** und **Y-Daten** am Punkt  $x$  nach Akima<sup>6</sup>. Das Verfahren berechnet vor der Interpolation an den Stützstellen die Neigung aus einer 5-Punkte-Umgebung.
- `cinterp(vx; vy; x)` kubische Spline-Interpolation
- `linterp(vx; vy; x)` lineare Interpolation

Beispiel: Interpolation von Zufallszahlen im Bereich von  $-0,5$  bis  $+0,5$

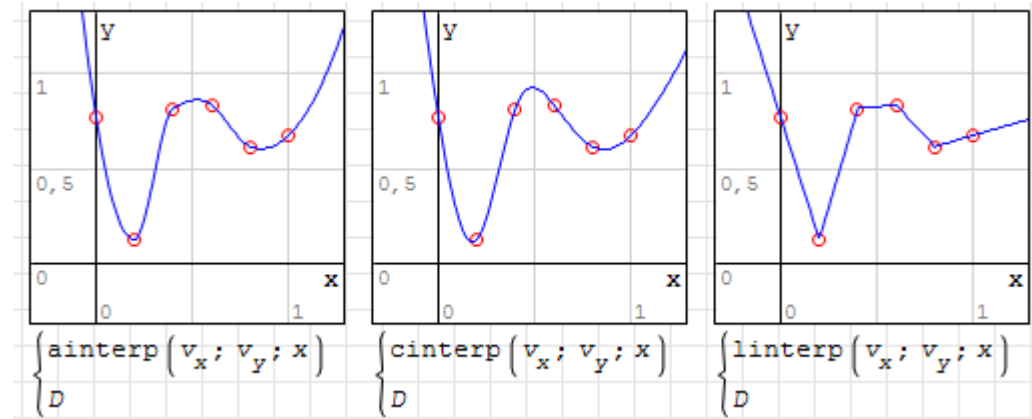
<sup>6</sup> Erläuterung des Verfahrens (TU Wien): [https://www.ads.tuwien.ac.at/docs/lva/mmgdv/k1\\_\\_011.htm](https://www.ads.tuwien.ac.at/docs/lva/mmgdv/k1__011.htm)

```

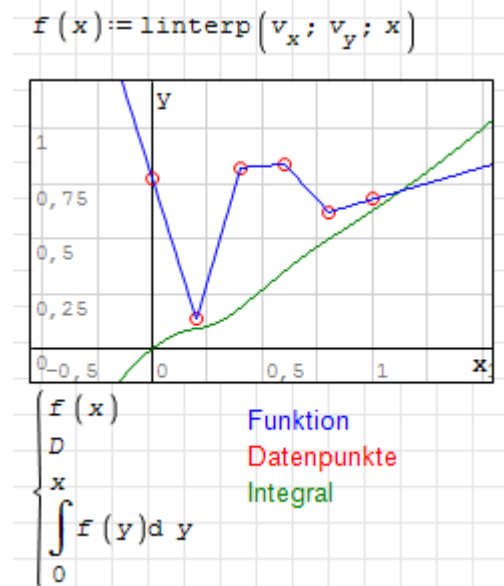
v_x := [0; 0,2..1]    v_y := Random(length(v_x))
v_x = [ 0 ]          v_y = [ 0,7714 ]    linterp(v_x; v_y; 0,3) = 0,4757
      [ 0,2 ]          [ 0,1325 ]
      [ 0,4 ]          [ 0,819 ]
      [ 0,6 ]          [ 0,8366 ]
      [ 0,8 ]          [ 0,6192 ]
      [ 1 ]            [ 0,6774 ]
D := augment(v_x; v_y; "o"; 10; "red")

```

Grafische Darstellung der drei Interpolationsarten zusammen mit den Datenpunkten:



Diese Funktionen können leider nicht abgeleitet werden und nicht mit Maßeinheiten umgehen. Allerdings können sie numerisch integriert werden.



Für die Interpolation von zweidimensionalen Tabellendaten steht die Funktion `InterpBilinear()` zur Verfügung. Die Daten sind als Matrix von  $z$ -Werten gegeben, deren Spalten bestimmten  $x$ -Werten und deren Zeilen bestimmten  $y$ -Werten zugeordnet sind. Die  $x$ - und  $y$ -Werte werden als Vektoren angegeben.

```

X := [ 20 50 100 150 200 250 ]

Y := [ 15 ]
      [ 20 ]
      [ 25 ]
      [ 40 ]
      [ 50 ]

M := [ 4 9 17 25 35 45 ]
      [ 4 10 19 28 39 50 ]
      [ 5 11 20 31 42 54 ]
      [ 5 12 23 35 47 60 ]
      [ 6 14 26 38 51 66 ]

InterpBilinear ( X ; Y ; M ; 30 ; 40 ) = 7,3333

```

Die Matrix und die Vektoren können übrigens übersichtlich mit dem Tabellenbereich dargestellt werden:

|    | 20 | 50 | 100 | 150 | 200 | 250 |
|----|----|----|-----|-----|-----|-----|
| 15 | 4  | 9  | 17  | 25  | 35  | 45  |
| 20 | 4  | 10 | 19  | 28  | 39  | 50  |
| 25 | 5  | 11 | 20  | 31  | 42  | 54  |
| 40 | 5  | 12 | 23  | 35  | 47  | 60  |
| 50 | 6  | 14 | 26  | 38  | 51  | 66  |

Table 1

M

**Table settings**

Data | Data appearance | Main layout | Resiz

X   show header

Y   show left stub



## 7. SMath Studio mit Maxima

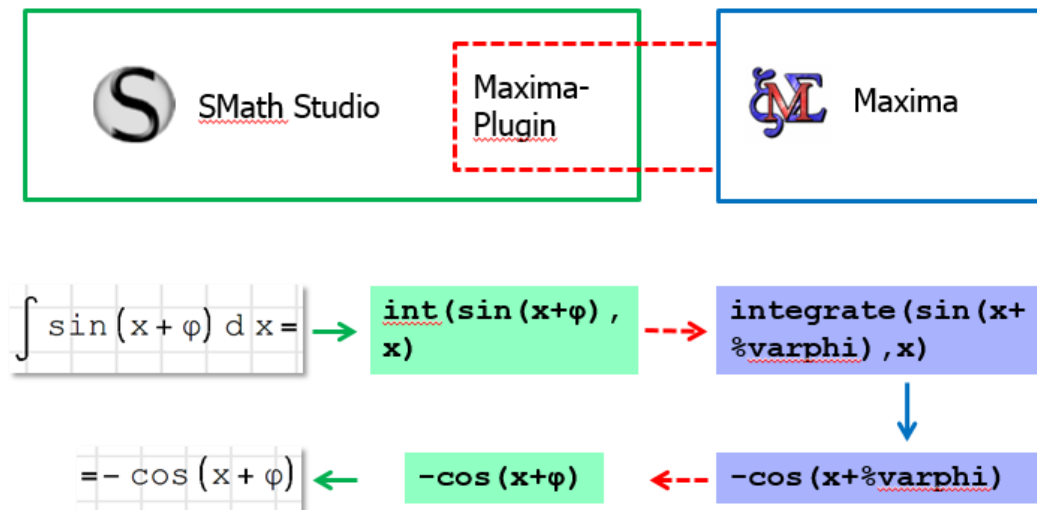


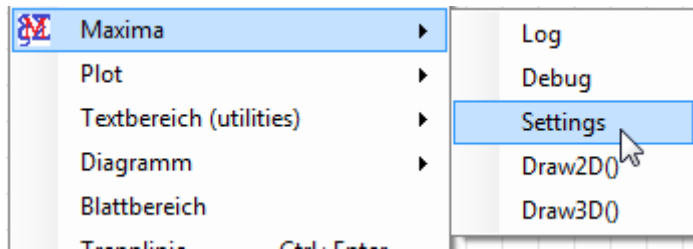
Abbildung 7.1.: Grundsätzliche Arbeitsweise der SMath-Maxima-Schnittstelle. Die Kommunikation erfolgt über einen TCP/IP-Socket.

Maxima ist ein quelloffenes und kostenloses Computeralgebra-Programm, also eine Software für das symbolische Rechnen mit Variablen. An der Technischen Hochschule Brandenburg wurde 2013 von Kay Graubmann und Martin Kraska mit Unterstützung der SMath-Entwickler Andrey Ivashov und Davide Carpi ein Plug-in entwickelt, welches den Zugriff auf Maxima von SMath Studio aus erlaubt.

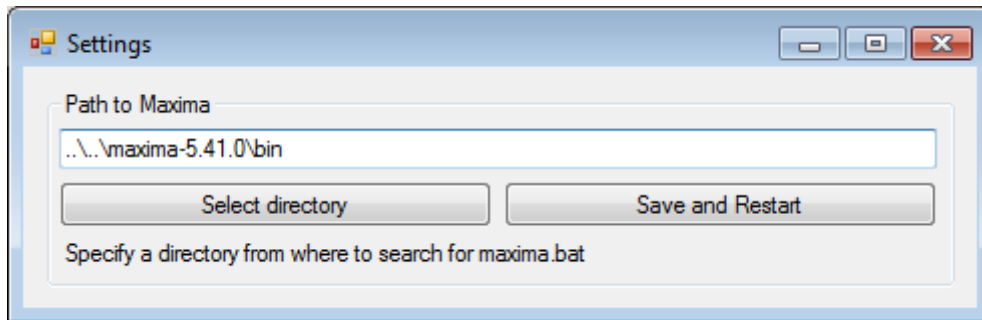
Wenn Sie die THB-Distribution benutzen, ist das Plug-in bereits installiert, Maxima selbst ist jedoch nicht Teil der Distribution.

### 7.1. Installation und Konfiguration

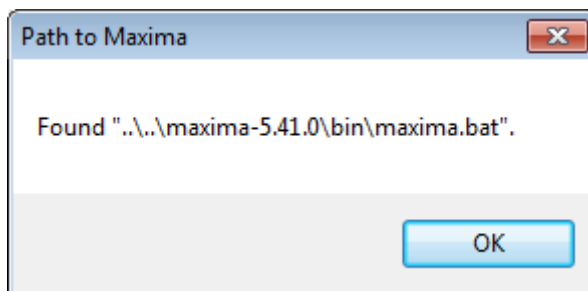
1. Laden Sie den [Maxima-Installer](#) herunter.
2. Installieren Sie Maxima im voreingestellten Verzeichnis (vermeiden Sie zumindest Leerzeichen und Sonderzeichen). Maxima und SMath müssen auf dem gleichen Laufwerk installiert sein.
3. Einstellung des Pfades zu Maxima in SMath Studio: **Hauptmenü** > **Einfügen** > **;Maxima** > **Settings**



Geben Sie dort den Pfad direkt an oder nutzen Sie den Auswahldialog. Es reicht, einen Startpunkt für die Suche nach dem Programm anzugeben:



Wählen Sie dann . Dann sollten ein Fortschrittsbalken und die folgende Meldung erscheinen:



Bestätigen Sie und schließen Sie den Settings-Dialog.

## 7.2. Funktionsübersicht

- Es gibt eine Funktion `Maxima()`, mit der Ausdrücke an Maxima übergeben, dort bearbeitet und an SMath zurückgegeben werden. Mit den gegenwärtig realisierten Übersetzungsfähigkeiten kann man bereits einen erheblichen Anteil des Funktionsumfangs von Maxima ansprechen.
- Das Plug-in stellt alternative Funktionen zu einigen eingebauten SMath-Funktionen bereit, die symbolisch rechnen können:
  - `Int()` für unbestimmte und bestimmte Integration,
  - `Lim()` für die Grenzwertbildung
  - `Diff()` für Ableitungen
  - `Det()` für Determinanten und
  - `Sum()` für Summen.

Mit Hilfe der Funktion `MaximaTakeover()` können die SMath-Funktionen umdefiniert werden, so dass sie die Maxima-Funktionen benutzen. Das hat den großen Vorteil, dass man dann die besser lesbare Operatorformatierung benutzen kann:

- SMath ohne Maxima-Unterstützung

$$\int \sin(x) dx = \int \sin(x) dx \quad \int_0^{\pi} \sin(x + \varphi) dx = \int_0^{\pi} \sin(x + \varphi) dx \quad \sin(x)' = \sin(x)'$$

$$\left| a \cdot \begin{bmatrix} 1 & 2 \\ b & 3 \end{bmatrix} \right| = \left| a \cdot \begin{bmatrix} 1 & 2 \\ b & 3 \end{bmatrix} \right| \quad \sum_{j=1}^n j = \blacksquare \quad \lim_{x \rightarrow 1+0} \frac{1}{x-1} = \lim_{x \rightarrow 1} \frac{1}{-1+x}$$

`lastError = "n - nicht definiert."`

- SMath mit Maxima-Unterstützung

```
MaximaTakeover("all") = "diff(), int(), lim(), sum(), det() handled by Maxima"
```

$$\int \sin(x) dx = -\cos(x) \quad \int_0^{\pi} \sin(x + \varphi) dx = 2 \cdot \cos(\varphi) \quad \sin(x)' = \cos(x)$$

$$\left| a \cdot \begin{bmatrix} 1 & 2 \\ b & 3 \end{bmatrix} \right| = a^2 \cdot (3 - 2 \cdot b) \quad \sum_{j=1}^n j = \frac{n \cdot (1 + n)}{2} \quad \lim_{x \rightarrow 1} \frac{1}{x-1} = \infty$$

- Weitere durch Maxima bereitgestellte Funktionen sind:
  - `Solve()` für die symbolische Lösung algebraischer Gleichungen und Gleichungssysteme (Abschnitt 6.11.1)
  - `ODE.2()` für die symbolische Lösung von Differenzialgleichungen (Abschnitt 6.13.1)
  - `Fit()` für nichtlineare Regressionen (Abschnitt 6.15.5)
  - `MSE()` Mittlerer quadratischer Abstand zwischen Daten und Modell
  - `Residuals()` Residuenvektor (Abstand zwischen Datenpunkten und Modell)
- Diagramme können in Maxima erzeugt und mit dem Image-Bereich angezeigt werden. Da Maxima intern mit Gnuplot für die Diagrammerstellung arbeitet, stehen viele Möglichkeiten von Gnuplot nun auch innerhalb von SMath zur Verfügung (siehe Abschnitt 8.1).
- Maßeinheiten, Textindizes und griechische Buchstaben in Namen werden voll unterstützt.
- Als Datentypen werden Zahlen (reell und komplex), Zeichenketten, Listen und Matrizen unterstützt.

Einschränkungen:

- Die Schnittstelle funktioniert nicht unter Linux.
- Namen mit Sonderzeichen (mit `Strg` K erzeugt) werden nicht unterstützt.
- Die numerischen Verfahren in Maxima kommen nicht mit Maßeinheiten zurecht.

- Logische Ausdrücke werden zwar übersetzt, aber in Maxima anders interpretiert. In Maxima gibt es den Datentyp `boolean`, der die Werte `true` oder `false` annehmen kann. In SMath entspricht dies den Zahlenwerten Null für `false` und ungleich Null für `true`.
- Es gibt Beschränkungen für die Länge von Ausdrücken, bedingt durch die Übersetzung mit Hilfe von regulären Ausdrücken in der Schnittstelle.
- Die Maxima-Schnittstelle ist vom Reifegrad her eine Konzeptdemonstration, es kann nicht damit gerechnet werden, dass immer alles wie beschrieben funktioniert.

Internet-Ressourcen zu Maxima:

- [Maxima Homepage](#)<sup>1</sup> auf Sourceforge,
- [Maxima Overview](#)<sup>2</sup> (englisch) für den Schnelleinstieg,
- Ein deutschsprachiges Maxima-Handbuch findet man [hier](#).<sup>3</sup>
- [Edwin T. Woollett](#)<sup>4</sup> Internetseite mit Maxima-Beispielen aus der Physik.

## 7.3. Die Maxima-Sitzung

Sobald erforderlich, baut SMath Studio über eine TCP-IP-Verbindung eine Maxima-Sitzung auf.

- Jede SMath-Instanz (Programmfenster) hat eine eigene Maxima-Sitzung, mit der das jeweils offene Rechenblatt kommuniziert.
- Einmal gestartet, läuft die Sitzung durch.
- Die Sitzung kann mit der Funktion `MaximaControl()` zurückgesetzt werden. Ein Neustart wird mit dem Argument „restart“ erzeugt, mit „cleanup“ werden definierte Variable und geladene Pakete wieder vergessen.

```
MaximaControl("restart")="Restart complete."
MaximaControl("cleanup")="Cleanup complete."
```

Alternativ kann die Sitzung mit **Einfügen** > **Maxima** > **Settings: Save and Restart** zurückgesetzt werden.

- Um unerwünschte Nebenwirkungen beim Umschalten zwischen Rechenblättern zu vermeiden, sollte man die Maxima-Sitzung am Anfang eines Rechenblatts zurücksetzen oder für jedes Rechenblatt ein neues SMath-Programmfenster öffnen.
- Es ist nicht auszuschließen, dass Maxima oder SMath abstürzen. Daher speichern Sie bitte ausreichend oft. Normalerweise schafft die Schnittstelle es, abgestürzte Sitzungen zu beenden. In Ausnahmefällen müssen Sie `sbcl.exe` (Steel Bank Common Lisp-Interpreter) im Windows-Taskmanager beenden.

<sup>1</sup><http://maxima.sourceforge.net/>

<sup>2</sup><http://www-com.physik.hu-berlin.de/~bunk/kurs/maxima/maxima.html>

<sup>3</sup>[http://maxima.sourceforge.net/docs/manual/de/maxima\\_12.html#SEC79](http://maxima.sourceforge.net/docs/manual/de/maxima_12.html#SEC79)

<sup>4</sup><http://www.csulb.edu/~woollett/>

Die Schnittstelle befindet sich noch in der Entwicklung, daher ist die Überschreibung der nativen SMath-Funktionen standardmäßig ausgeschaltet. Sie können einzeln oder gemeinsam ab- oder angeschaltet werden. Dazu dient die Funktion:

`MaximaTakeover()`. Sie nimmt eine oder mehrere Zeichenketten als Argumente, in denen nach folgenden Teilketten gesucht wird:

- `all` Alle Funktionen übernehmen.
- `int` Integrationsfunktionen übernehmen
- `dif` Ableitungsfunktionen übernehmen
- `lim` Grenzwertfunktion übernehmen
- `det` Determinantenfunktion übernehmen

Wird keine dieser Teilketten gefunden, dann wird die Übernahme zurückgesetzt. Diese Umschaltung kann an jeder Stelle eines Rechenblatts vorgenommen werden und gilt dann bis zum nächsten Aufruf von `MaximaTakeover()`.

Die Übernahme wird auch durch Restart oder Cleanup mittels `MaximaControl()` zurückgesetzt.

| Funktion              | SMath                | Operator           | Maxima               |
|-----------------------|----------------------|--------------------|----------------------|
| Ableitung             | <code>diff(1)</code> | $'$                | <code>Diff(1)</code> |
|                       | <code>diff(2)</code> | $\frac{d}{d}$      | <code>Diff(2)</code> |
|                       | <code>diff(3)</code> | $\frac{d}{d}$      | <code>Diff(3)</code> |
| Unbestimmtes Integral | <code>int(2)</code>  | $\int d$           | <code>Int(2)</code>  |
|                       | <code>int(4)</code>  | $\int d$           | <code>Int(4)</code>  |
| Determinante          | <code>det()</code>   | $ $                | <code>Det()</code>   |
| Grenzwert             | <code>lim()</code>   | $\lim \rightarrow$ | <code>Lim()</code>   |
| Summe                 | <code>sum(4)</code>  | $\sum_{=}$         | <code>Sum(4)</code>  |


Tabelle 7.1.: SMath-Funktionen und die Maxima-Entsprechungen, die mit `MaximaTakeover()` aktiviert werden können. Die Maxima-Entsprechungen stehen unabhängig vom Übernahmestatus immer zur Verfügung, haben aber keine Operator Darstellung.

Die symbolischen Ergebnisse können mit der Optimierungseinstellung „Symbolisch“ oder „abgeschaltet“ angezeigt werden. Bei „symbolisch“ versucht SMath selbst noch, das Ergebnis weiter zu vereinfachen. Probieren Sie im Einzelfall aus, welche Einstellung Ihnen besser gefällt.

## 7.4. Die Funktion Maxima()

Diese Funktion nimmt einen Ausdruck, übersetzt ihn in die Maxima-Syntax, führt ihn in Maxima aus und übersetzt das Ergebnis zurück nach Maxima. Nicht alles, was in Maxima geht, kann man auch von SMath Studio aus machen, es ist aber schon sehr viel.

Die `Maxima()`-Funktion besitzt eine Operatordarstellung (Maxima-Logo):

```
MaximaControl("restart")="Restart complete."


$$\left\{ \begin{array}{l} "a+b" \\ a+a \end{array} \right\} = \left\{ \begin{array}{l} "a+b" \\ 2 \cdot a \end{array} \right\}$$

```

| Objekt               | SMath Studio  | Maxima                |
|----------------------|---|-----------------------|
| Dezimaltrennzeichen  | Komma oder Punkt  | Punkt                 |
| Argumenttrennzeichen | Semikolon oder Komma  | Komma                 |
| Textindex            | $a_b$   | <code>a_%_b</code>    |
| Unterstrich in Namen | $a_b$   | <code>a_b</code>      |
| Spezielle Konstanten | <code>e, i</code>   | <code>%e, %i</code>   |
| Listen               | $\left\{ \begin{array}{l} \blacksquare \\ \blacksquare \end{array} \right\}$                              | <code>[]</code>       |
| Matrizen             | $\left[ \begin{array}{cc} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{array} \right]$ | <code>matrix()</code> |
| Adressindex          | $v_j$   | <code>v[j]</code>     |
| Zeichenketten        | <code>"abc"</code>  | <code>"abc"</code>    |
| Maßeinheiten         | <code>cm</code>   | <code>%unitcm</code>  |

Tabelle 7.2.: SMath-Objekte und ihre Entsprechung in Maxima

Die Übersetzung von Namen ist so gestaltet, dass sich bei Rückübersetzung wieder der Originalname ergibt. Was hinter den Kulissen passiert, kann man mit der Protokollfunktion `MaximaLog()` sehen. Sie zeigt den letzten von SMath übergebenen Ausdruck und die Antwort von Maxima.

```
MaximaLog( $\blacksquare$ )="Received Bytes: 27
(%i22) [%a+b$,2*a];
(%o22) [%a+b$,2*a]"
```

Alternativ kann man auch die Übersetzungsstufen im Maxima-Debugger verfolgen.

## 7.5. Logging

The communication between SMath and Maxima is logged by default. You can access the log using canvas functions or from the Insert menu.

There are three types of logs:

- **Socket Log:** This is a concatenation of the strings exchanged between SMath and Maxima
- **wxMaxima command log:** This contains the commands sent to Maxima formatted as wxMaxima input.
- **Translation Log:** For each request to Maxima, this log shows
  - The SMath expression in text format
  - The translation to Maxima
  - The response from Maxima
  - The translation to SMath in text format

### 7.5.1. MaximaLog()

```
MaximaLog(#)
```

shows the last input and output of Maxima with the respective labels.

```
MaximaLog("all")
```

displays Maxima input and output of the running Maxima session beginning from the last restart or cleanup or manual log reset. The output may be quite large. In this case, it is better to use the interactive Log window. This log is a simple concatenation of the strings exchanged between SMath and Maxima.

### 7.5.2. Log Window

The log window is called either from the canvas by `MaximaLog("big")` or via **Insert** > **Maxima** > **Log**.

## 7.6. Configuration

The plugin has some user-configurable items:

- Directory, where *maxima.bat* is located (path relative to the SMath installation directory. This path is best configured using the settings dialog **Insert** > **Maxima** > **Settings**. Usually it is sufficient to point to the top level Maxima installation directory.
- User definitions for Maxima, to be evaluated at the begin of a session and after restart or cleanup.
- User translations from SMath to Maxima and from Maxima to SMath as a pair of regular expressions.

The user definitions and translations are stored in the config file *maxima.xml*, which resides in the top level maxima plugin directory.

### 7.6.1. Example For Custom Definitions in *maxima.xml*

Let's assume we want to add handling of the SMath function  $\text{round}(x,n)$  which is not available in Maxima. In Maxima, we have just  $\text{round}(x)$ , which rounds to zero decimal places.

First, we have to define a helper function in Maxima, to which we later translate  $\text{round}(x,n)$ :

```
round2(x,n):=round(x*10^n)/10^n$
```

We add this definition to *maxima.xml*:

```
<CustomLoadCommands>
  <Command>round2(x,n):=round(x*10^n)/10^n$</Command>
</CustomLoadCommands>
```

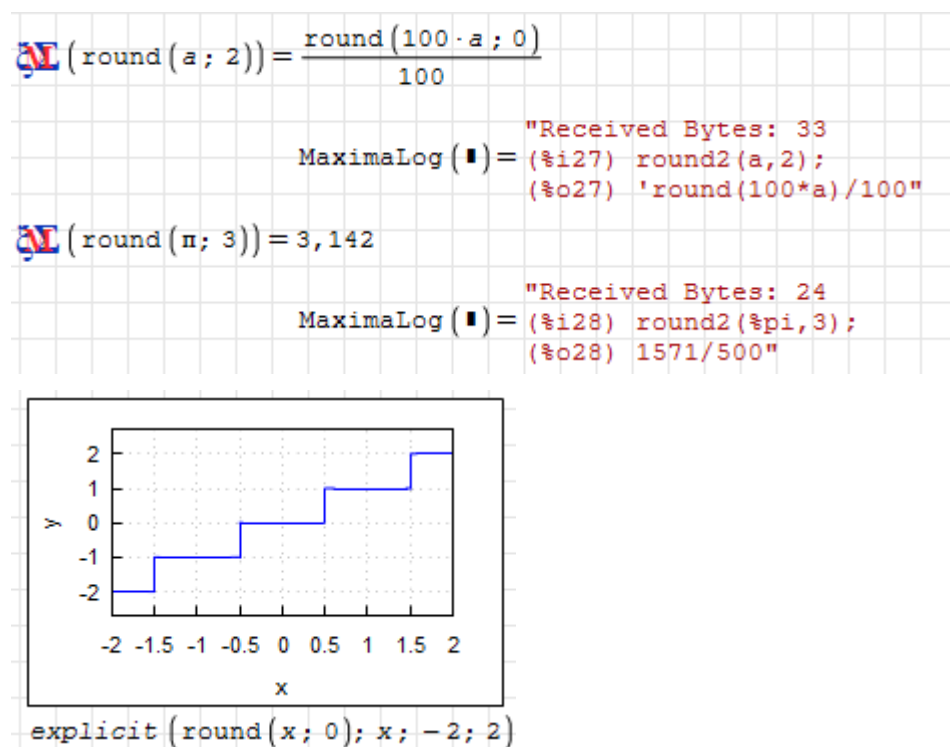
Second, we need to add translation rules to map  $\text{round}(x,n)$  to  $\text{round2}(x,n)$ :

```
<ConvertFromSMathToMaxima>
  <Expression regex="round\" replace="round2("></Expression>
</ConvertFromSMathToMaxima>
```

As it may happen that a symbolic expression returned from Maxima contains  $\text{round}(x)$ , we also want to add a translation into the SMath equivalent  $\text{round}(x,0)$ . This is a bit more involved, as we have to isolate the argument of  $\text{round}()$ , in order to add a second one:

```
<ConvertFromMaximaToSMath>
  <Expression regex="round\((.+)\)" replace="round($1,0)"></Expression>
</ConvertFromMaximaToSMath>
```

Note that by the time you are reading this, the above translation has been implemented in the plug-in.





The plot above is generated using a Maxima `Draw2D()` plot region.

## 8. Grafik

Grundsätzlich werden die auf Gnuplot basierenden Grafikfunktionen des *Maxima*-Plug-ins empfohlen, welches die Funktionen `Draw2D()` und `Draw3D()` bereitstellt. Diese erzeugen Grafikdateien, die mit dem Image-Bereich angezeigt werden können.

Alternativ dazu gibt es die Maxima `Draw2D()` und `Draw3D()`-Grafikbereiche, bei denen die Grafikobjekte direkt in den Platzhalter geschrieben werden können. Sie bieten interaktive Skalierung und Drehung sowie ein Einstellungsmenü für einige Grafikoptionen.

Um Diagramme mit Maxima geht es in Abschnitt 8.1.

Wenn man die Draw-Kommandos von Maxima scheut, dann gibt es als Alternative den *X-Y Plot-Bereich*, bei dem viele Optionen in einem Einstellungsdialog interaktiv gesetzt werden können (Abschnitt 8.3).

Ähnlich, im vorliegenden Handbuch aber nicht dokumentiert, funktioniert der *ZedGraph*-Bereich, bereitgestellt vom gleichnamigen Plugin.

Für einfache Diagramme mit wenig Anspruch an die Formatierung oder wenn Animationen erstellt werden sollen, eignet sich der 2D-Diagrammbereich, den *SMath Studio* standardmäßig mitbringt (Abschnitt 8.2).

Das Plug-in *GPC Wrapper* enthält Funktionen zum Umgang mit ebenen Polygonen. Da dies ein Spezialgebiet mit sehr engem Interessentenkreis ist, wurde die Beschreibung in den Anhang (Abschnitt G.3) ausgelagert.

Das *Modeller* Plug-in stellt einen flexiblen interaktiven Grafikbereich bereit, der allerdings nur low-level Objekte bietet (Abschnitt 8.4).

Weitere, allerdings nicht für den Standardanwender empfohlene Diagrammarten werden in Anhang G dokumentiert:

- Standard-3D-Diagrammbereich (fehlerbehaftet, nicht in der Weiterentwicklung),
- Plug-in *3D Plot Region*: 3D-Diagramme, allerdings ohne Speicherung der Einstellungen.

| Funktion             | 2D intern             | 3D intern             | X-Y Plot                 | ZedGraph        | 3D Plot         | Maxima                    |
|----------------------|-----------------------|-----------------------|--------------------------|-----------------|-----------------|---------------------------|
| Objekte              |                       |                       |                          |                 |                 |                           |
| Funktionen           | $f(x)$                | $f(x,y)$              | $f(1)$                   |                 | $f(2)$          | beliebig                  |
| Implizite Funktionen | -                     | -                     | <b><math>f(2)</math></b> |                 | -               | beliebig                  |
| Ausdrücke            | $x$                   | $x,y$                 | beliebig                 |                 | -               | beliebig                  |
| Matrizen             | $n \times 2$          | $n \times 3$          | $n \times 2$             |                 | -               | $n \times 2, n \times 3,$ |
| Text                 | <b><math>x</math></b> | -                     | -                        |                 | -               | $x$                       |
| Achsen               |                       |                       |                          |                 |                 |                           |
| Gitter               | fest                  | fest                  | <b>variabel</b>          | <b>variabel</b> | <b>variabel</b> | variabel                  |
| Grenzen              | Maus                  | Maus                  | Menü                     | Menü            | Menü            | variabel                  |
| lin/log              | -                     | -                     | -                        | $x$             | -               | $x$                       |
| polar                | -                     | -                     | -                        |                 | -               | ( $x$ )                   |
| Formatierung         |                       |                       |                          |                 |                 |                           |
| Linien               | fest                  | fest                  | <b>variabel</b>          | <b>variabel</b> | ?               | variabel                  |
| Marker               | separat               | -                     | <b>variabel</b>          | <b>variabel</b> | ?               | variabel                  |
| Sonderfunktionen     |                       |                       |                          |                 |                 |                           |
| Animation            | <b><math>x</math></b> | <b><math>x</math></b> | -                        | -               | -               | -                         |
| Entwicklung          | stabil                | inaktiv               | aktiv                    | aktiv           | inaktiv         | aktiv                     |

Tabelle 8.1.: Vergleich der unterschiedlichen Diagrammbereiche in SMath Studio.

## 8.1. Diagramme mit *Maxima*

Mit den Funktionen `Draw2D()` und `Draw3D()` können 2D- und 3D-Diagramme erstellt werden. Das geschieht unter Verwendung von **Gnuplot**<sup>1</sup>, einem flexiblen und leistungsfähigen quelloffenen Grafikprogramm. Fähigkeiten und Eigenschaften der Schnittstelle sind:

- 2D- und 3D-Diagramme
- Grafikobjekte:
  - explizit, implizit und parametrisch gegebene Kurven und Flächen. In 3D können Kurven und Flächen anhand beliebiger Ausdrücke eingefärbt werden,
  - Datenpunkte und Farbwerte aus Matrizen,
  - Vektoren (Linien mit Pfeilspitzen),
  - Polygone, Ellipsen, Rechtecke,
  - Beschriftungen einschließlich Hoch- und Tiefstellungen sowie Symbolen (z.B. griechische Buchstaben),
  - zusätzliche  $x$ - und  $y$ -Achsen.
- Objekte und Optionen (Einstellungen) werden in Listen an die Grafikbefehle übergeben.
- Andere Maxima-Pakete definieren Spezialdiagramme, z.B. aus der Statistik oder Richtungsfelder für Differenzialgleichungen.

<sup>1</sup>[www.gnuplot.info](http://www.gnuplot.info)

Neben dem Maxima-Handbuch findet man auch [hier](#)<sup>2</sup> eine Übersicht über die Möglichkeiten des Maxima-Pakets *draw*.

Für die Verwendung in SMath werden Diagramme als temporäre Grafikdatei exportiert (PDF, SVG oder PNG) und dann mit einem Image-Bereich dargestellt.

Einschränkungen sind momentan:

- Programmier-ähnliche Erstellung von Diagrammen (man muss die Objekte und Optionen kennen, keine Menü-Führung),
- Keine interaktive Skalierung und Drehung der Achsen. Dies geht allerdings mit den Maxima-Draw2D()/3D()-Bereichen (Abschnitt 199)
- Das Grafikobjekt `spherical()` funktioniert gegenwärtig nicht.
- Einheitenbehaftete Größen werden in Basiseinheiten umgerechnet und dann dargestellt. Soll eine spezifische Maßeinheit verwendet werden, so sind abhängige Variablen oder Funktionen durch die entsprechende Einheit zu teilen und unabhängige Variablen mit diesen Einheiten zu multiplizieren.
- Nicht alle Funktionen und Ausdrücke werden korrekt von SMath nach Maxima übersetzt. Nicht unterstützt werden beispielsweise logische Ausdrücke.
- Der Image-Bereich kann die Beschriftungen in von Maxima erzeugten SVG-Bildern nicht korrekt darstellen. Daher wird in den folgenden Beispielen meist das PNG-Format durch Angabe eines Dateinamens mit entsprechender Endung erzwungen. Für Vektorgrafik kann auf das PDF-Format ausgewichen werden.

### 8.1.1. Funktionen `Draw2D()` und `Draw3D()`

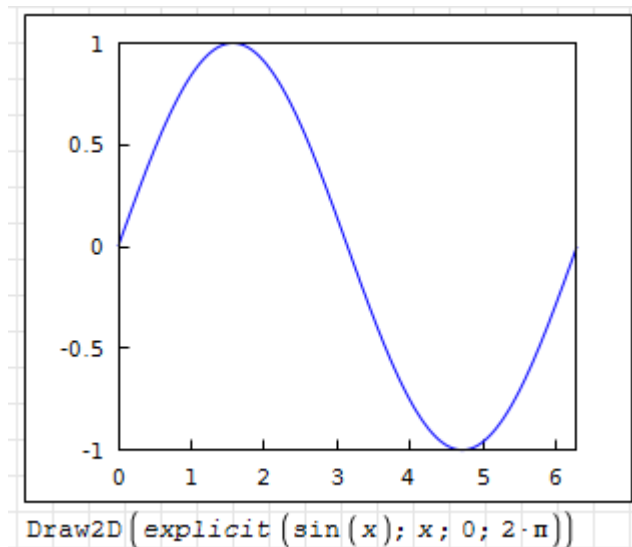
Diese Funktionen erzeugen Grafikdateien für die Anzeige mit dem Image-Bereich oder auch zur externen Verwendung.

Mit zusätzlichen Argumenten kann man Größe und Dateiname des Diagramms bestimmen:

```
Draw2D(Objekte)
Draw2D(Objekte; Name)
Draw2D(Objekte; {Breite
           Höhe})
Draw2D(Objekte; Name; {Breite
                       Höhe})
```

Rückgabewert der Funktion ist der vollständige Name mit Pfad der erzeugten Grafikdatei. Diese Datei kann dann in einem Image-Bereich angezeigt werden. Da dieser einen Dateinamen im Platzhalter erwartet, kann `Draw2D()` einfach dort eingesetzt werden.

<sup>2</sup><http://riotorto.users.sourceforge.net/Maxima/gnuplot/index.html>



Standardmäßig wird eine temporäre PNG-Datei mit zufälligem Namen erzeugt. Durch Angabe von "pdf" oder "svg" für den Namen wird die temporäre Datei im entsprechenden Format erzeugt.

Wird ein anderer Name angegeben, wird die Datei mit diesem Namen relativ zum aktuellen Verzeichnis erzeugt.

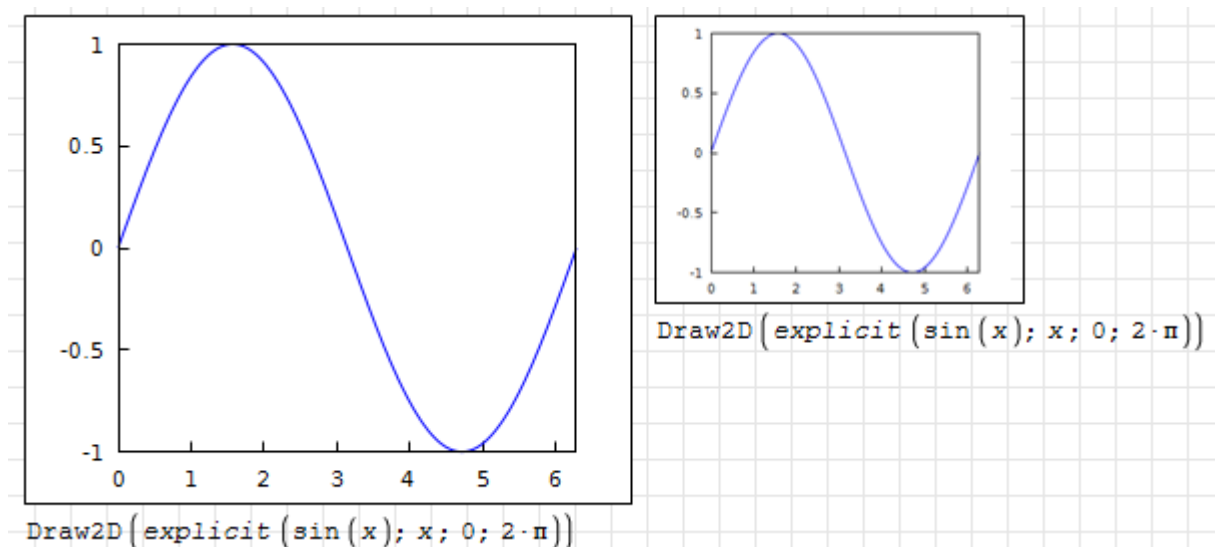
Die Größe beträgt standardmäßig  $300 \times 250$  Punkte (Breite mal Höhe).

Für dreidimensionale Diagramme verwenden Sie die Funktion:

`Draw3D()`

Sie hat die gleichen Argumente wie `Draw2D()`, allerdings bei einer Standardgröße von  $300 \times 300$  Punkten.

Die Bilder passen sich der Größe des Image-Bereichs an (bei Beibehaltung des Seitenverhältnisses), die Schrift skaliert dabei mit. In der Standardgröße entspricht die Schriftgröße 8 Punkten.



Die Diagramme in einem Rechenblatt sollten alle die gleiche Schriftgröße haben, daher sollten die Bilder gleich in der gewünschten Größe erzeugt werden und der Image-Bereich mit **Kontextmenü** > **Reset to original size** auf Originalgröße gestellt werden.

### 8.1.2. Achsen und Gitter

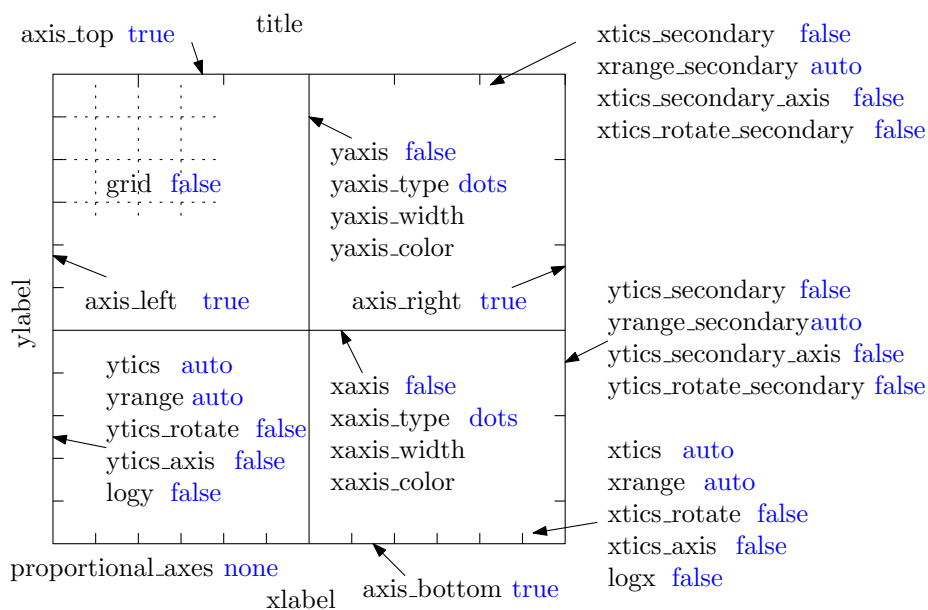


Abbildung 8.1.: Optionen zur Achsenformatierung in Draw2D(). Voreinstellungen in Blau.

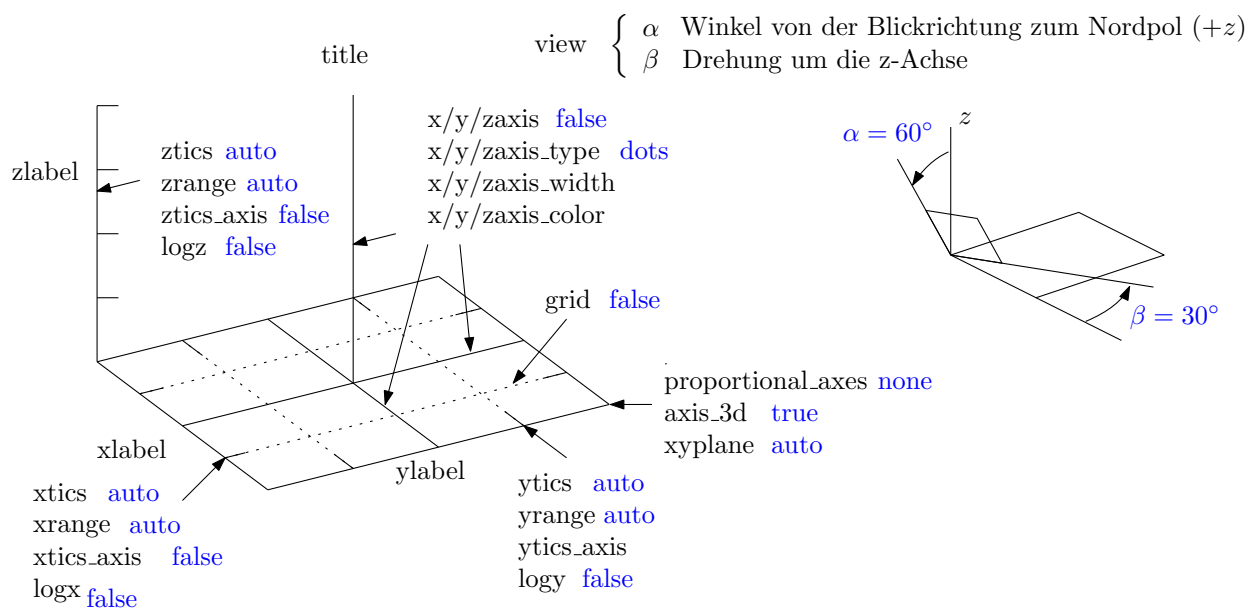


Abbildung 8.2.: Optionen zur Achsenformatierung in Draw3D(). Voreinstellungen in Blau.

Die Linienart für das Gitter (grid) kann mit Gnuplot-Preamble-Befehlen gesetzt werden, z.B.

```
user_preamble="set grid ls -1"
```

Dies erzeugt schwarze Gitterlinien.

### 8.1.3. Beschriftungen

Text kann man im Diagramm auf folgende Weisen unterbringen:

- Achsenbeschriftungen (Optionen xlabel, ylabel, zlabel, xlabel\_secondary, ylabel\_secondary)
- Diagrammtitel (Option title)
- Frei plazierte Beschriftungen (Objekt label()).
- Hochstellungen werden mit ^ erzeugt: a^b liefert  $a^b$ ,
- Tiefstellungen werden mit \_ erzeugt: a\_b liefert  $a_b$ ,
- Sonderzeichen können mit einem dreistelligen Oktalcode gemäß der folgenden Abbildung erzeugt werden.
  - Die linken Zeichen werden mit `\\xxx` erzeugt
  - Die rechten Zeichen werden mit `{/Symbol \\xxx}` erzeugt.
- Die letzere Variante kann auch mit lateinischen Buchstaben zur Erzeugung der entsprechenden griechischen Buchstaben benutzt werden. `{/Symbol a}` liefert  $\alpha$ . Das ist aber nicht nötig, da die griechischen Buchstaben korrekt von SMATH durchgereicht werden.

|     |     |         |          |         |           |          |         |         |          |
|-----|-----|---------|----------|---------|-----------|----------|---------|---------|----------|
| 000 | 000 | 001     | 002      | 003     | 004       | 005      | 006     | 007     |          |
| 010 |     | 011     | 012      | 013     | 014       | 015      | 016     | 017     |          |
| 020 |     | 021     | 022      | 023     | 024       | 025      | 026     | 027     |          |
| 030 |     | 031     | 032      | 033     | 034       | 035      | 036     | 037     |          |
| 040 |     | 041     | !!!      | 042 " ∇ | 043 # #   | 044 \$ ε | 045 % % | 046 & & | 047 ' '₃ |
| 050 | ( ( | 051 ) ) | 052 * *  | 053 + + | 054 , ,   | 055 - -  | 056 . . | 057 //  |          |
| 060 | 0 0 | 061 1 1 | 062 2 2  | 063 3 3 | 064 4 4   | 065 5 5  | 066 6 6 | 067 7 7 |          |
| 070 | 8 8 | 071 9 9 | 072 ::   | 073 ; ; | 074 < <   | 075 = =  | 076 > > | 077 ? ? |          |
| 100 | @ ≅ | 101 AA  | 102 BB   | 103 CX  | 104 D Δ   | 105 EE   | 106 F Φ | 107 G Γ |          |
| 110 | HH  | 111 II  | 112 J θ  | 113 KK  | 114 L Λ   | 115 MM   | 116 NN  | 117 O O |          |
| 120 | P Π | 121 Q Θ | 122 RP   | 123 S Σ | 124 TT    | 125 U Y  | 126 V ς | 127 W Ω |          |
| 130 | X Ξ | 131 Y Ψ | 132 Z Z  | 133 [ [ | 134 \ ∴   | 135 ] ]  | 136 ^ ∓ | 137 _ _ |          |
| 140 |     | 141 a α | 142 b β  | 143 c χ | 144 d δ   | 145 e ε  | 146 f φ | 147 g γ |          |
| 150 | h η | 151 i ι | 152 j θ  | 153 k κ | 154 l λ   | 155 m μ  | 156 n ν | 157 o o |          |
| 160 | p π | 161 q θ | 162 r ρ  | 163 s σ | 164 t τ   | 165 u u  | 166 v υ | 167 w ω |          |
| 170 | x ξ | 171 y ψ | 172 z ζ  | 173 { { | 174       | 175 } }  | 176 ~ ~ | 177     |          |
| 200 |     | 201     | 202      | 203     | 204       | 205      | 206     | 207     |          |
| 210 |     | 211     | 212      | 213     | 214       | 215      | 216     | 217     |          |
| 220 |     | 221     | 222      | 223     | 224       | 225      | 226     | 227     |          |
| 230 |     | 231     | 232      | 233     | 234       | 235      | 236     | 237     |          |
| 240 | €   | 241 Å Y | 242 ∼ '  | 243 t ≤ | 244 π /   | 245 L ∞  | 246 S f | 247 § ♣ |          |
| 250 | ˆ ♦ | 251 S ♥ | 252 \$ ♣ | 253 T ↔ | 254 Z ←   | 255 †    | 256 Z → | 257 Z ↓ |          |
| 260 | ° ° | 261 a ± | 262 , "  | 263 † ≥ | 264 ' ×   | 265 † ×  | 266 s θ | 267 ° • |          |
| 270 | ÷   | 271 s # | 272 s ≅  | 273 t ≈ | 274 z ... | 275      | 276     | 277 z ↓ |          |
| 300 | R N | 301 Å 3 | 302 Å 3  | 303 Å ρ | 304 Å ⊗   | 305 L ⊕  | 306 C ⊗ | 307 C n |          |
| 310 | C u | 311 É ∃ | 312 E ∃  | 313 E € | 314 E c   | 315 Í ≤  | 316 Í ε | 317 D ≠ |          |
| 320 | ∅ ∫ | 321 N ∇ | 322 N ⊗  | 323 O ⊗ | 324 O ™   | 325 O Π  | 326 O √ | 327 × - |          |
| 330 | R ∼ | 331 Ū λ | 332 Ū v  | 333 Ū ≅ | 334 Ū =   | 335 Y n  | 336 T = | 337 B u |          |
| 340 | f ◊ | 341 á < | 342      | 343     | 344       | 345   Σ  | 346     | 347     |          |
| 350 |     | 351     | 352      | 353     | 354       | 355      | 356     | 357     |          |
| 360 |     | 361 n > | 362 ñ f  | 363 ó f | 364       | 365 ó j  | 366     | 367     |          |
| 370 |     | 371     | 372      | 373     | 374       | 375      | 376     | 377 ° y |          |

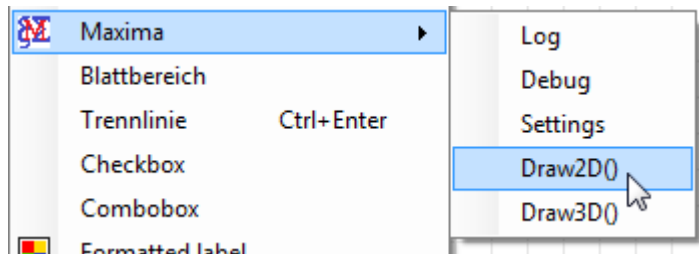
Abbildung 8.3.: Sonderzeichen für Beschriftungen. Das erste Zeichen nach dem Oktalcode xxx wird mit `\\xxx` erzeugt, das zweite mit `{/Symbol \\xxx}`. Funktioniert nur im PNG-Bildformat.

### 8.1.4. Maxima-Diagrammbereiche

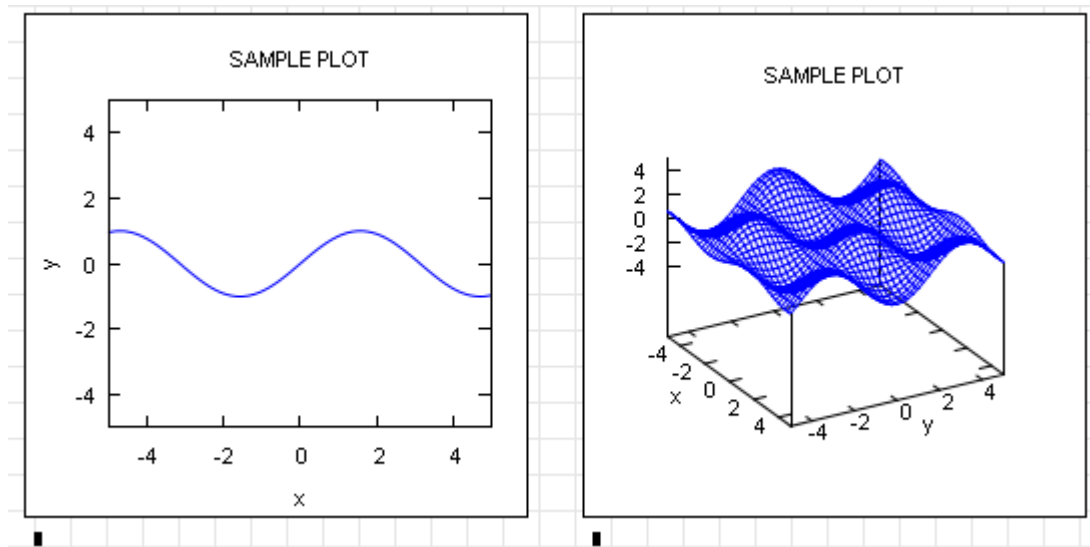
Das Maxima-Plug-in stellt eigene Diagrammbereiche bereit, bei denen die Grafikbefehle direkt in den Platzhalter eingegeben werden können. Vorteile:

- Maussteuerung für Verschiebung, Zoomen und Drehung (bei 3D) der Achsen
- Maussteuerung für die Bildgröße bei gleichbleibender Schriftgröße
- Einstellungsdialog für Achsenbeschriftungen, Titel, Achsbereiche, lin/log-Achsen, Schriftgröße, Bildformat (PNG oder SVG)

Solche Bereiche erzeugt man mit **Einfügen** > **Maxima** > **Maxima Draw2D()** oder **Maxima Draw3D()**



Es erscheint ein Beispieldiagramm (links: Draw2D(), rechts Draw3D()):



In den Platzhalter kann man eine Liste mit Draw-Objekten und Optionen einfügen, genauso wie bei den Funktionen `Draw2D()` und `Draw3D()`.

Einige Optionen können über das Einstellungsfenster gesteuert werden, welches sich mit Doppelklick auf das Diagramm öffnet.

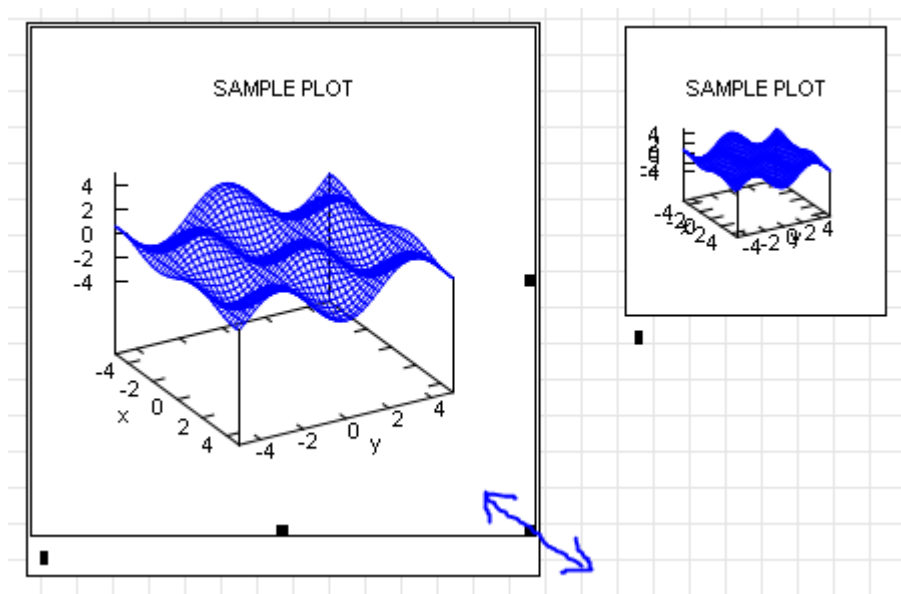
Das Kontextmenü bietet unter anderem die Optionen:

- Platzhalter mit den Grafikbefehlen verbergen,
- Außenrand des Bildes abschalten,
- Bild in eine Datei exportieren.

### Maussteuerung

Ziehen am Bildrahmen: Bildgröße ändern (Neuberechnung mit Beibehaltung der Schriftgröße)





| Action       | Draw2D()             | Draw(3D)       |                |
|--------------|----------------------|----------------|----------------|
|              |                      | Primary mode   | Secondary mode |
| Double click | Open Settings dialog |                |                |
| Drag         | Pan (move)           | Orbit (rotate) | Pan X and Y    |
| Mouse wheel  | Zoom                 | Zoom all       | Zoom Z         |

Tabelle 8.2.: Mouse actions in the interactive Maxima Draw-Regions. Primary and secondary mode are switched by pressing A when the region has focus.

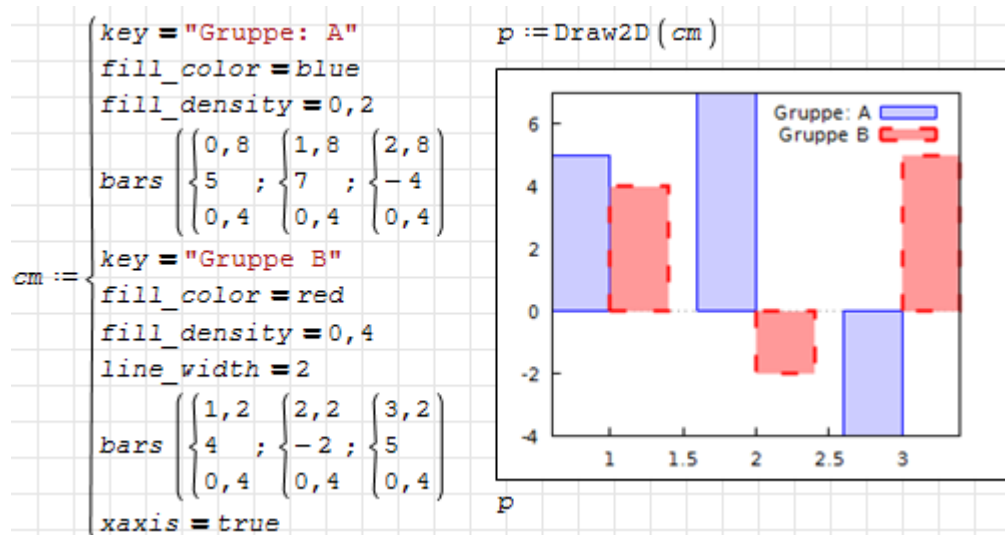
### 8.1.5. Grafikobjekte

Hier finden Sie eine Auswahl an Objekten, die in Diagrammen benutzt werden können.

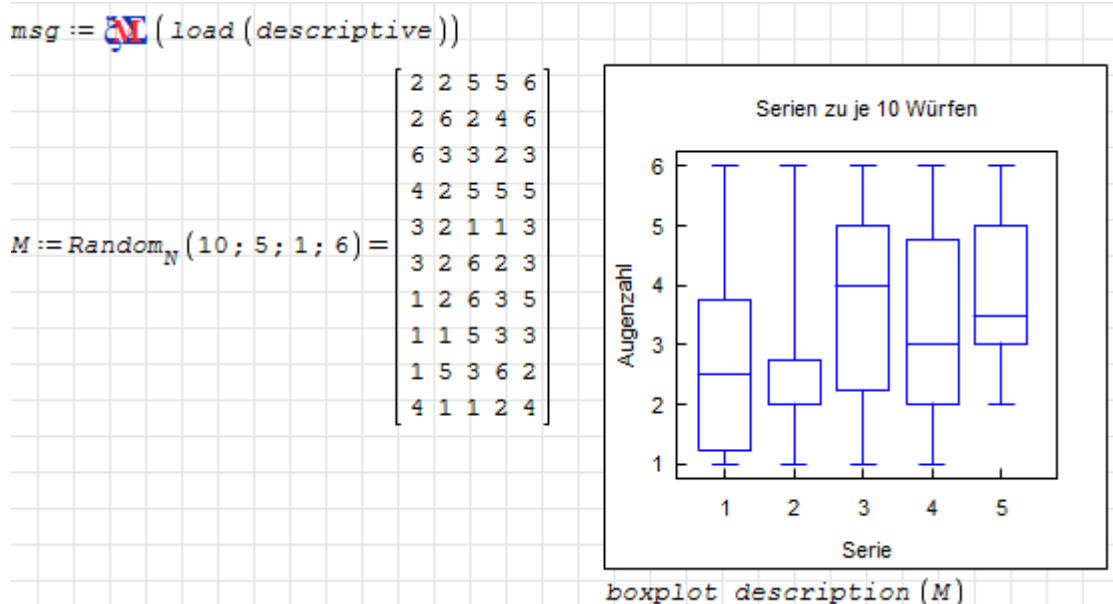
Hinweis: Der Code-Baustein (Snippet) `draw` kann man die dynamische Hilfe über diese Objekte und ihre Optionen informieren. Das vereinfacht die Erstellung von Diagrammen enorm.

`bars(b1; b2; b3...)` Elementare Funktion für Säulendiagramme, die einzelnen Säulen sind durch Listen mit der Position  $x$ , der Höhe  $h$  und der Breite  $b$  gegeben:

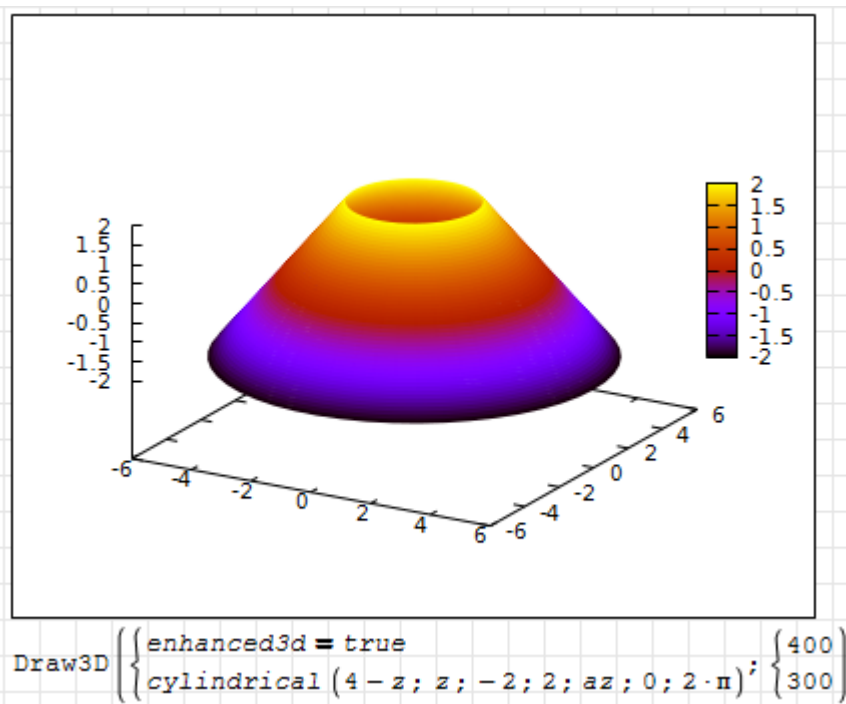
$$\begin{cases} x \\ h \\ b \end{cases}$$



`boxplot_description(M)` Boxplot zur Darstellung von Verteilungsparametern (Minimum, Maximum, Median, unteres und oberes Quartil).



`cylindrical(r(z; φ); z; zmin; zmax; φ; φmin; φmax)` Darstellung von Flächen, die als Funktion der axialen Koordinate  $z$  und des Azimut-Winkels  $az$  gegeben sind.



`elevation_grid( $M$ ;  $x_0$ ;  $y_0$ ;  $w$ ;  $h$ )` 3D-Darstellung einer  $m \times n$  Matrix von  $z$ -Werten.  $M_{11}$  entspricht  $x = x_0$  und  $y = y_0 + h$ ,  $M_{mn}$  entspricht  $x = x_0 + w$  und  $y = y_0$ .

`line_type` Linienart der Gitterlinien

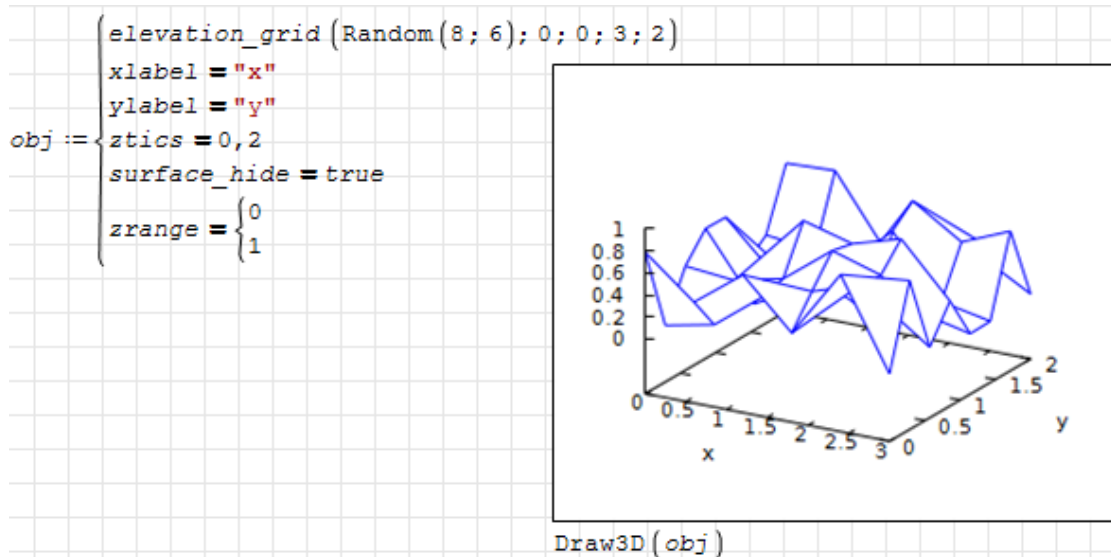
`line_width` Linienbreite der Gitterlinien

`key` Legendeneintrag

`color` Linienfarbe

`wired_surface` (false) oder true: Gitternetz auf eingefärbten Flächen anzeigen oder nicht.

Hier als Beispiel eine  $8 \times 6$ -Zufallsmatrix:



`ellipse(xc; yc; a; b; ang1; ang2;)` Ellipse, zentriert bei  $x_c$ ,  $y_c$  mit den Halbachsen  $a$  und  $b$ , von Winkel  $ang1$  bis Winkel  $ang2$

`nticks` Zahl der Punkte für die Randkurve

transparent ungefüllt oder gefüllt (hier kann keine Transparenz eingestellt werden)

fill\_color Füllfarbe

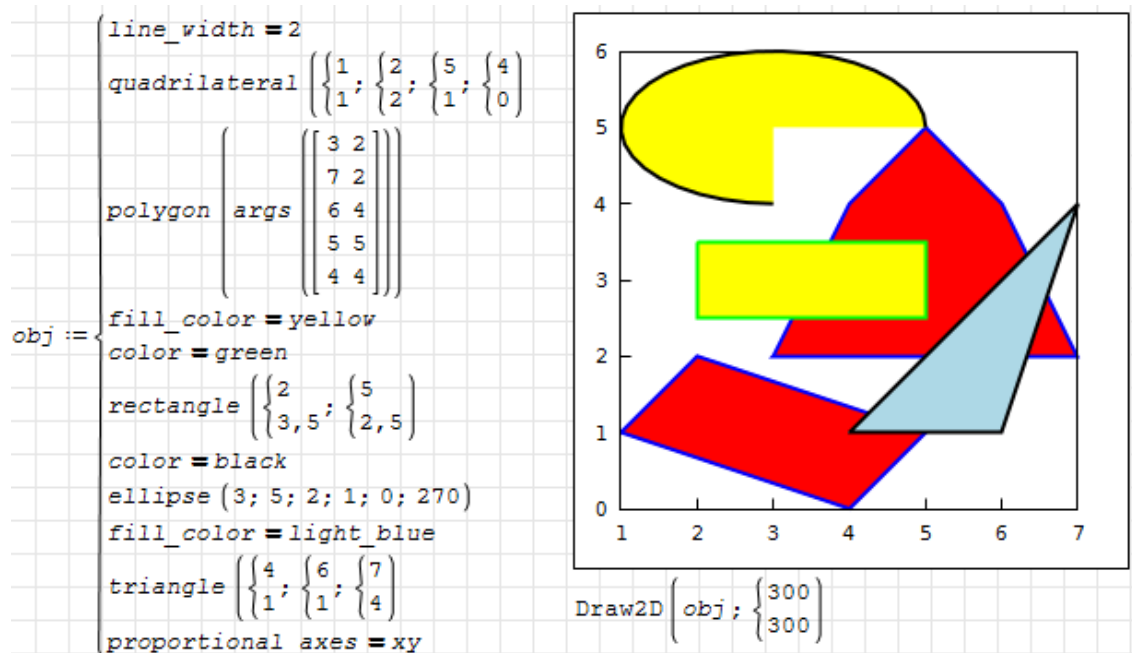
border Rand zeichnen

line\_type Linienart für den Rand

line\_width Linienbreite für den Rand

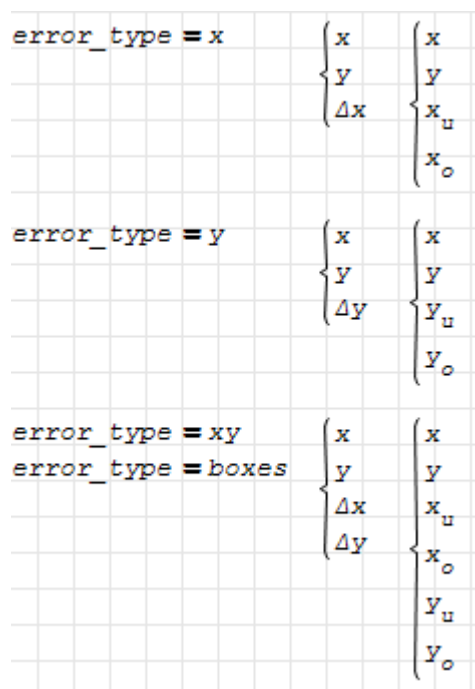
key Legendeneintrag

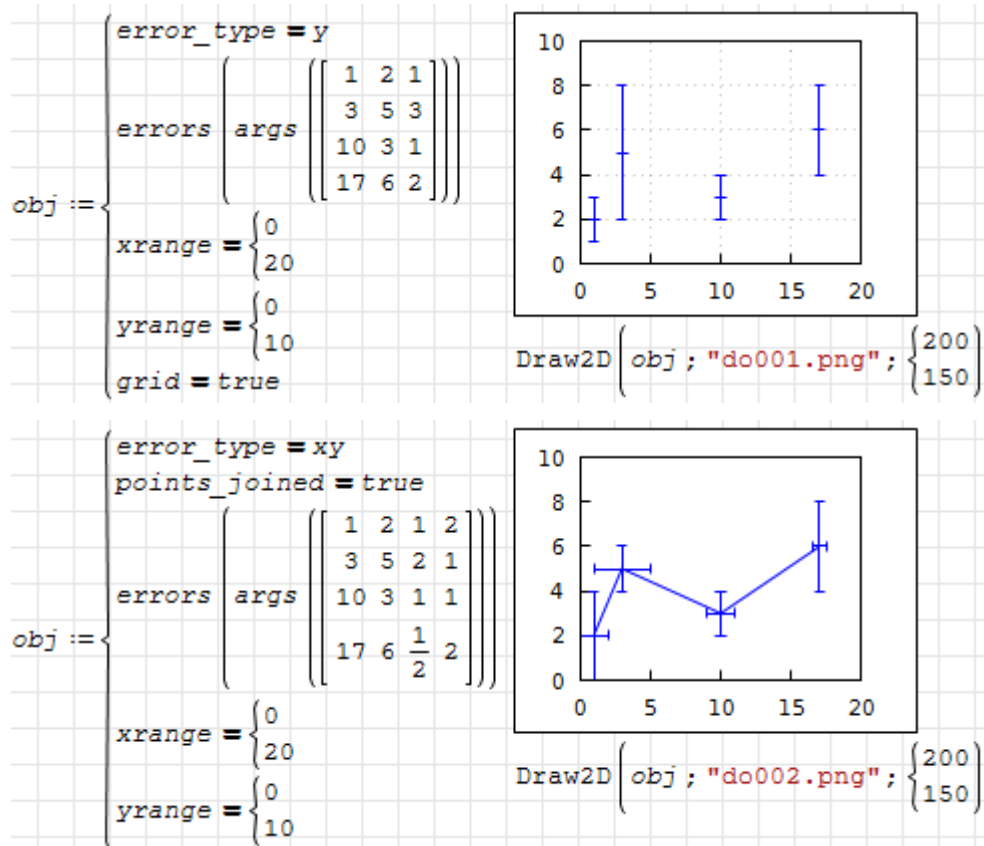
color Linienfarbe



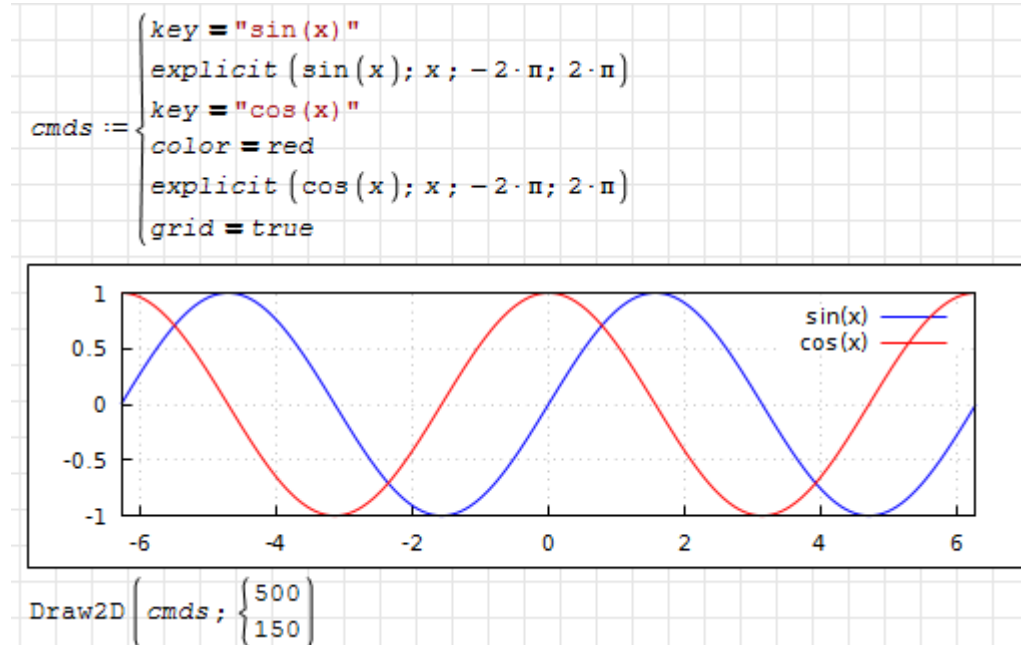
errors(11; 12; ...) Fehlerbalken. Die Argumente sind Listen, deren Einträge abhängig vom Balkentyp sind.

error\_type Balkenart (x, y, xy oder boxes)



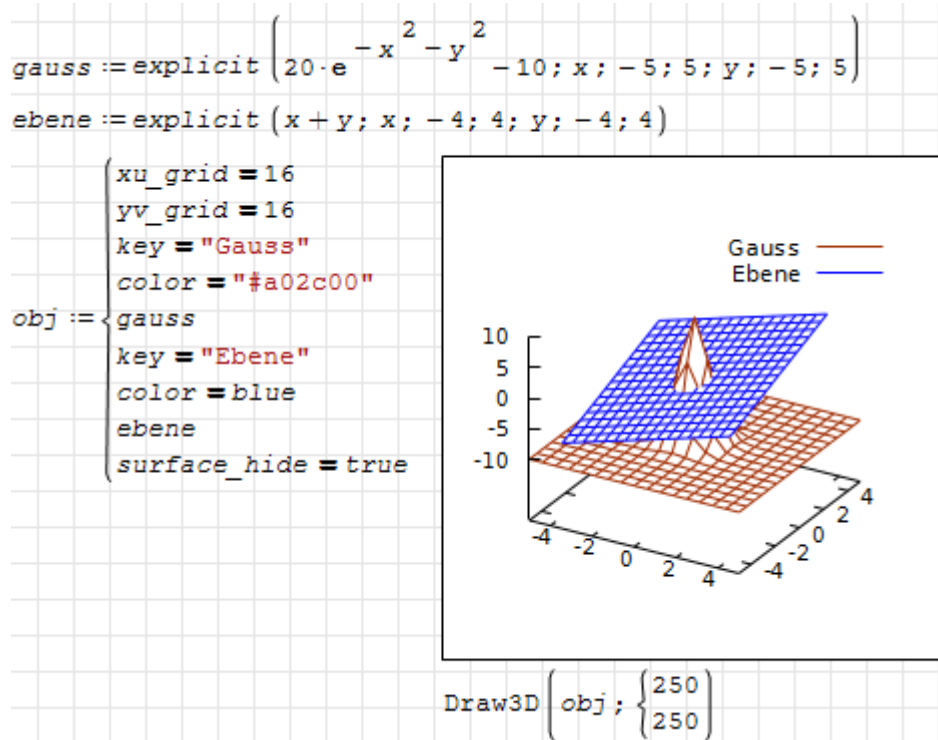


`explicit(expr; var; min; max)` Darstellung explizit gegebener Funktionen  $y(x)$  innerhalb eines vorgegebenen Intervalls



`explicit(z(x; y); x; xmin; xmax; y; ymin; ymax)` Darstellung explizit gegebener

Funktionen  $z(x, y)$  innerhalb eines vorgegebenen Intervalls:



`histogram_description(data; option1; option2...)` Histogramm-Objekt. `data` ist eine Liste oder ein Vektor. Die Optionen müssen separat angegeben werden (können nicht zu Listen zusammengefasst werden).

`nclasses` (10) Klassenzahl des Histogramms oder eine Liste mit den Bereichsgrenzen und optional deren Anzahl:

$$\left\{ \begin{array}{l} x_{\max} \\ x_{\max} \\ n_k \end{array} \right. \quad \text{oder} \quad \left\{ \begin{array}{l} x_{\max} \\ x_{\max} \end{array} \right.$$

`frequency (absolute)` Skalierung der Klassenwerte:

`absolute` Anzahl der Werte in der Klasse

`relative` Relativer Anteil der Werte in der Klasse

`percent` Relativer Anteil in %

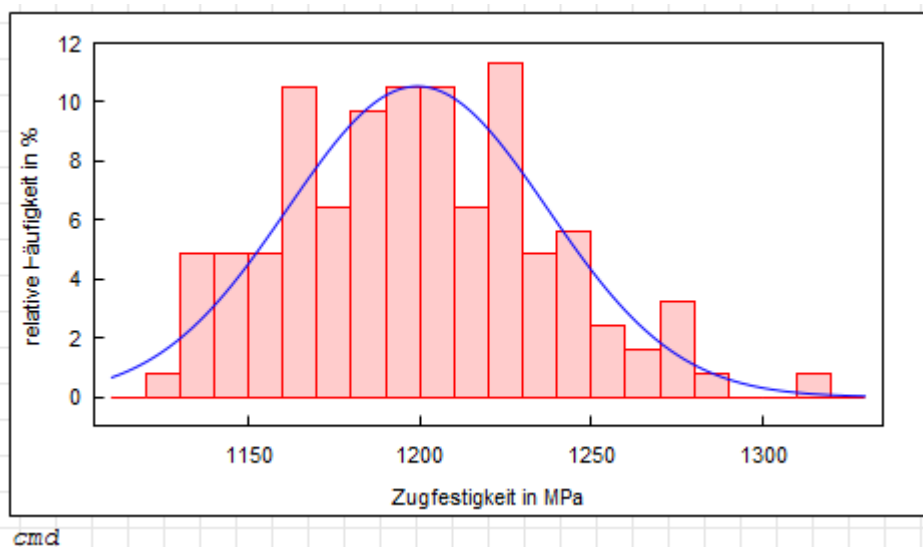
`density` Mittlere Verteilungsdichte (relativer Anteil durch Klassenbreite)

`htics (auto)`, `endpoints`, `intervals` oder eine Liste mit Marken

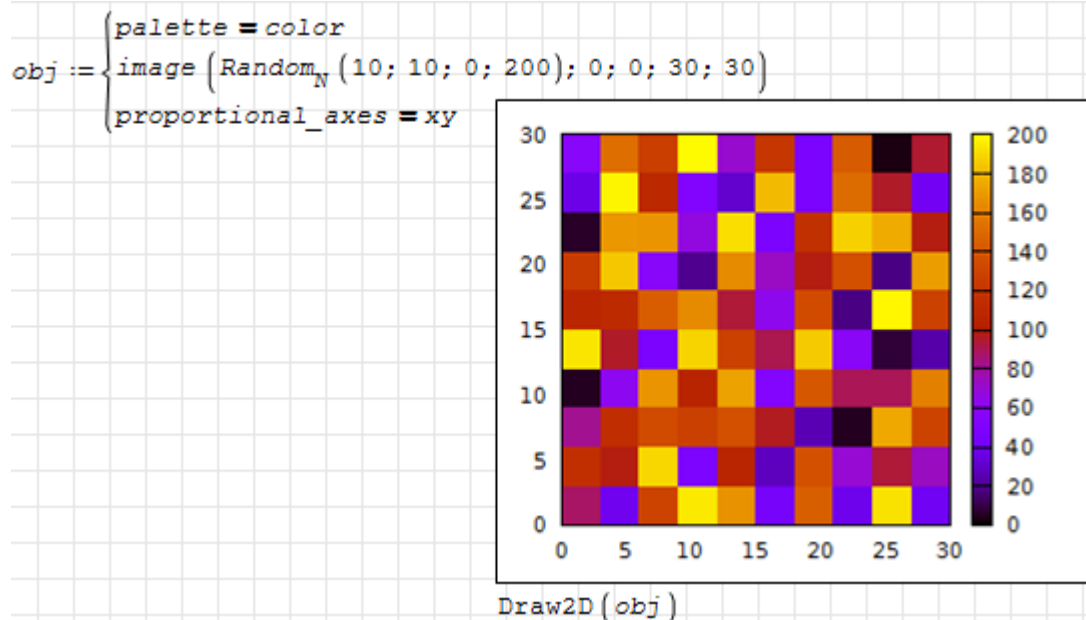
```

fill_density = 0,2
histogram_description ( D; nclasses = { xmin
                                       xmax; frequency = percent
                                       nk } )
cmd := {
  explicit ( 100 * δ * pdf_normal ( x; Rm_μ; Rm_σ ); x; xmin; xmax )
  xlabel = "Zugfestigkeit in MPa"
  ylabel = "relative Häufigkeit in %"
  yrange = { -1
            12 }

```



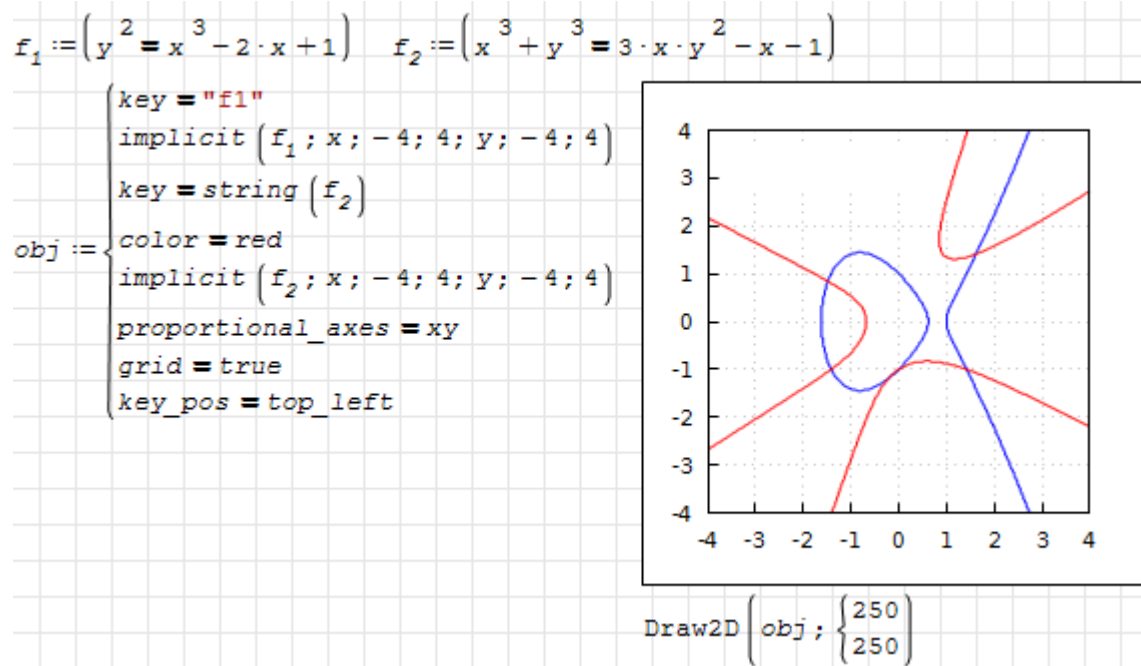
`image(Matrix; xmin; ymin; xmax; ymax)` Darstellung von Matrizen reeller Werte als Farbwerte in der  $xy$ -Ebene darstellen.



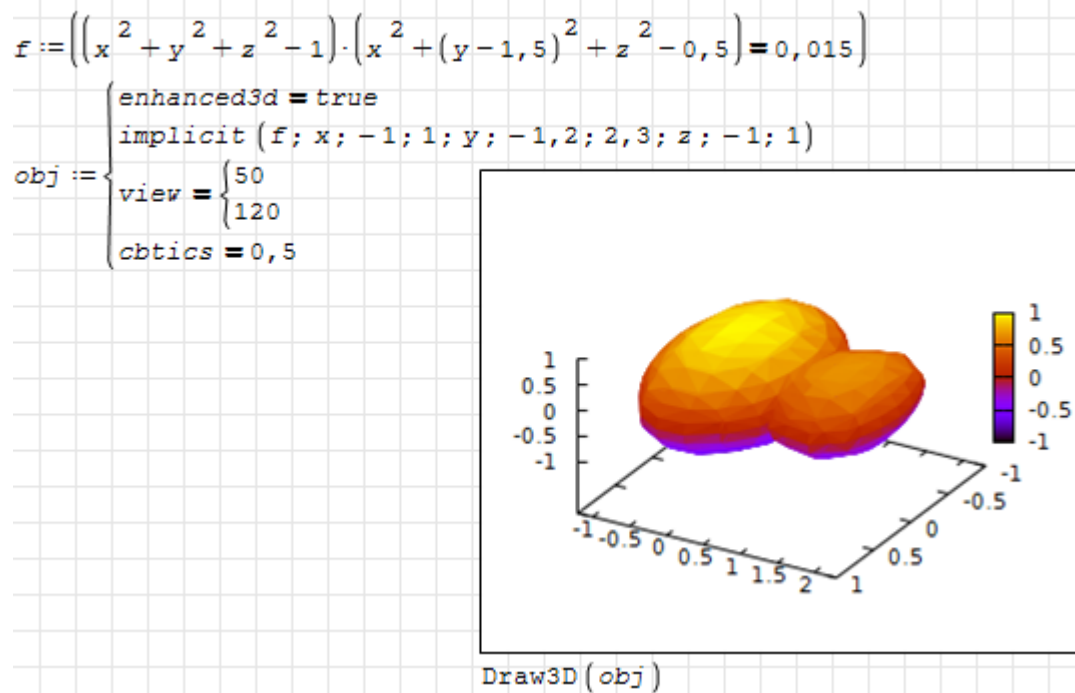
`implicit(f(x; y); x; xmin; xmax; y; ymin; ymax)` Darstellung implizit gegebener Funktionen

`ip_grid` Anzahl von Intervallen für die Funktionsabastung in  $x$ - und  $y$ -Richtung

`ip_grid_in` Sekundäre Intervallteilung



`implicit(f(x; y; z); x; xmin; xmax; y; ymin; ymax; z; zmin; zmax)` Darstellung implizit gegebener Funktionen in 3D



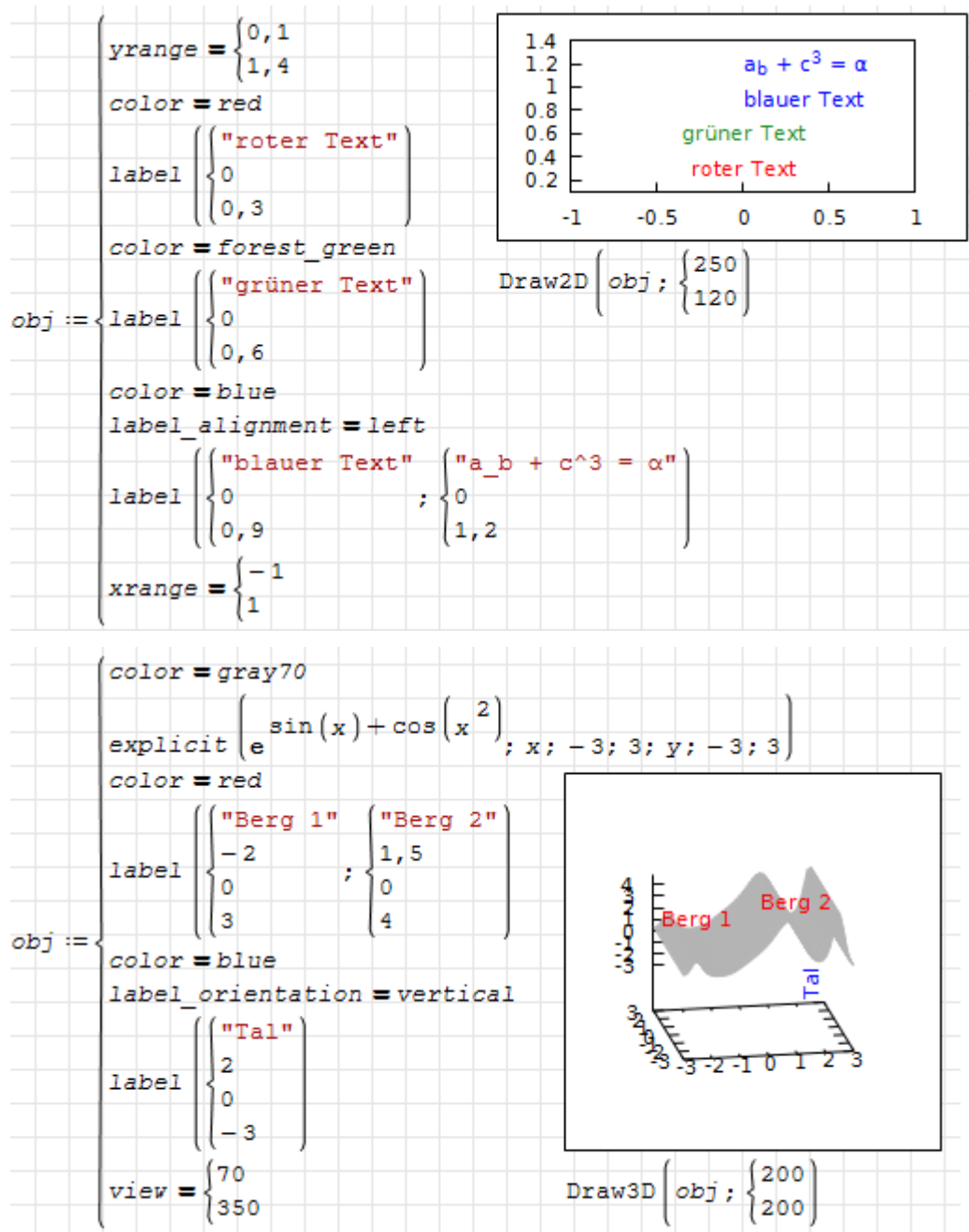
`label(Liste)` Beschriftung, einzufügen an beliebigen Positionen, die Liste hat die Form

$$\left\{ \begin{array}{l} \text{Text} \\ x \\ y \end{array} \right. \quad \text{oder} \quad \left\{ \begin{array}{l} \text{Text} \\ x \\ y \\ z \end{array} \right.$$

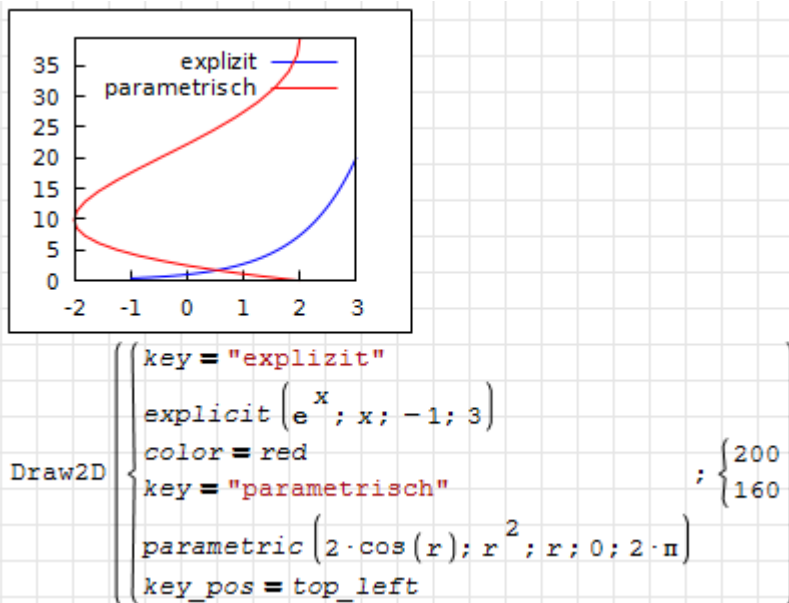
`label_alignment` Horizontale Ausrichtung (center, left, right)

`label_orientation` horizontal oder vertical

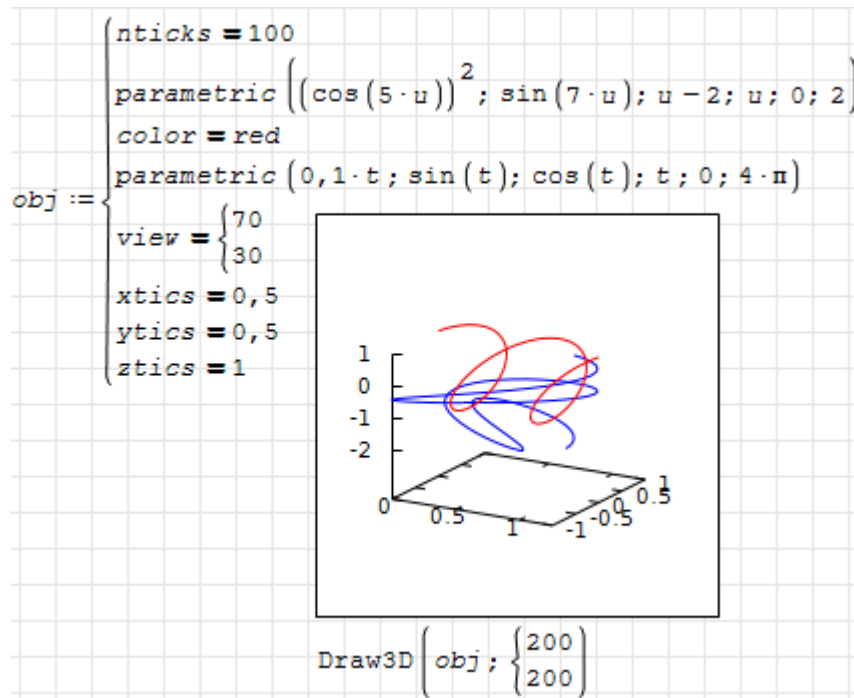




`parametric(x; y; t; tmin; tmax)` parametrisch gegebene Funktion in 2D. Die üblichen Optionen für Linienobjekte sind anwendbar.



`parametric(x; y; z; t; tmin; tmax)` parametrisch gegebene Funktion in 3D.



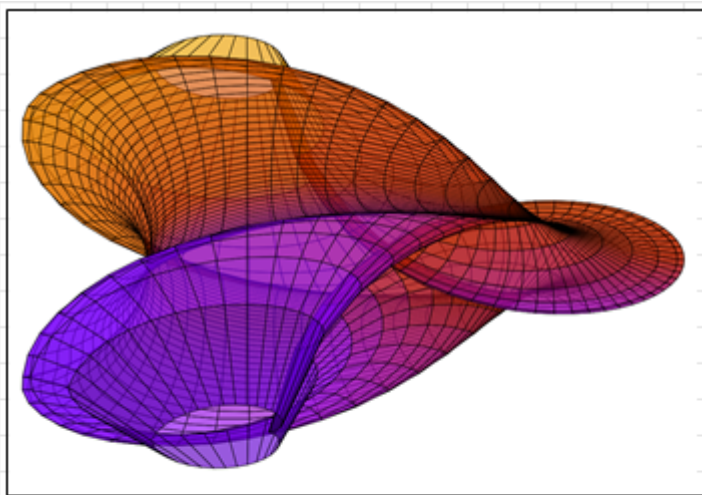
`parametric_surface(fx; fy; fz; u; umin; umax; v; vmin; vmax)` parametrisch gegebene Fläche in 3D.

Das folgende Beispiel ist von der [Gnuplot-Beispiel-Seite](#) inspiriert.

```

user_preamble = {"set style fill transparent solid 0.65 border"
                 "set view 120, 357, 1.75, 1.0"}
axis_3d = false
colorbox = false      a := 1
enhanced3d = true     fx := (2*a*(cos(u)+u*sin(u))*sin(v)) /
                        (1+u^2*(sin(v))^2)
xu_grid = 51
yv_grid = 51         fy := (2*a*(sin(u)-u*cos(u))*sin(v)) /
                        (1+u^2*(sin(v))^2)
xtics = none
ytics = none
ztics = none
wired_surface = true  fz := a*ln(tan(v/2)) + (2*cos(v)) /
                        (1+u^2*(sin(v))^2)
color = black
parametric_surface (fx; fy; fz; u; -4,5; 4,5; v; 0,05; pi-0,05)

```



```

Draw3D (obj; {1000
             700})

```

Hier noch ein Beispiel aus dem Maxima-Handbuch:

$$fx := \frac{u}{2} \cdot \cos(u) \cdot (\cos(v) + 1)$$

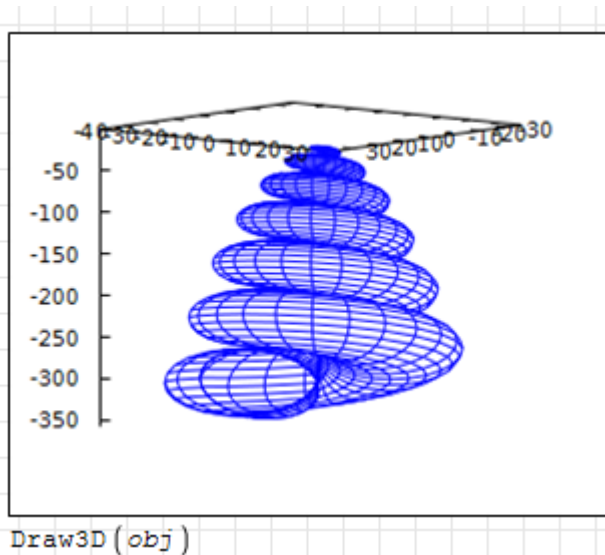
$$fy := \frac{u}{2} \cdot \sin(u) \cdot (\cos(v) + 1)$$

$$fz := u \cdot \sin(v) - \left( \frac{u+3}{8} \cdot \pi \right)^2 - 20$$

```

obj := {user_preamble = {"set view 100, 320, 1, 1.5"}
        xu_grid = 150
        yv_grid = 25
        parametric_surface (fx; fy; fz; u; 0; 13*pi; v; -pi; pi)
        surface_hide = true
        xyplane = 0}

```



`piechart_description(data; option1; option2...)` Sektordiagramm-Objekt. `data` ist eine Liste oder ein Vektor. Die Optionen müssen separat angegeben werden (können nicht zu Listen zusammengefasst werden). Das Diagramm wird aus `ellipse()`-Objekten aufgebaut.

`sector_colors` (Zufallsfarben) Liste mit den Farben der Sektoren

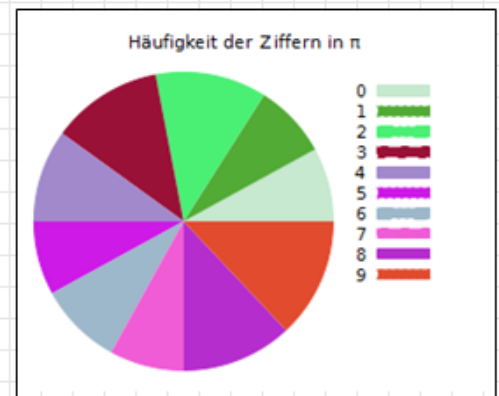
`pie_center`  $\begin{cases} 0 \\ 0 \end{cases}$  Mittelpunkt

`pie_radius` (1) Sektorradius

```
msg := M(load(descriptive))
```

```
S := M(read_matrix(file_search("pidigits.data")))
```

```
p := Draw2D(
  (
    title = "Häufigkeit der Ziffern in n"
    piechart_description(S)
    proportional_axes = xy
    xrange = { -1, 8 }
    {
      axis_top = false
      axis_bottom = false
      axis_left = false
      axis_right = false
      xtics = false
      ytics = false
    }
  )
)
```



`points(Matrix)` 2D oder 3D Punkte darstellen

`color=blue` Farbe

`key=""` Legendeneintrag

`point_size=1` Punktgröße

`point_type=plus` Punktart, gegeben als Zahl oder als Name.

`points_joined=false` Verbindungsart:

`false` keine Verbindung

`true` Polygonzug

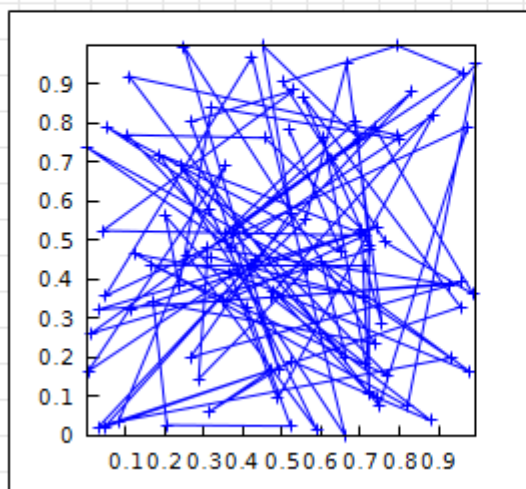
`impulses` Linien zur  $x$ -Achse oder zur  $xy$ -Ebene (bei 3D Punkten)

Die folgenden Diagramme zeigen verbundene Punkte mit zufälligen Koordinaten. Links die PNG-Version und rechts die PDF-Version.

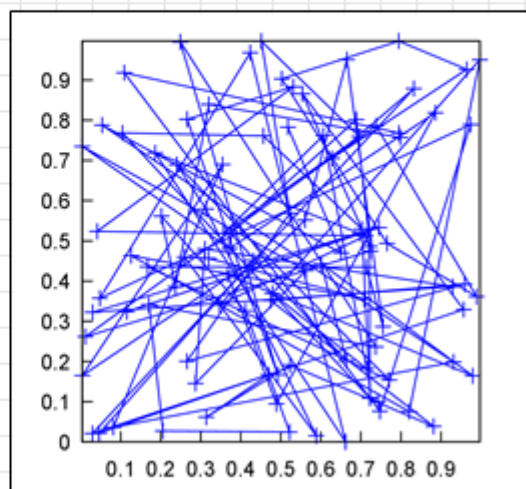
```

cm := {
  points_joined = true
  points (Random(100; 2))
  proportional_axes = xy
}
s := {
  260
  240
}
p1 := Draw2D(cm; s)
p2 := Draw2D(cm; "pdf"; s)

```



p1



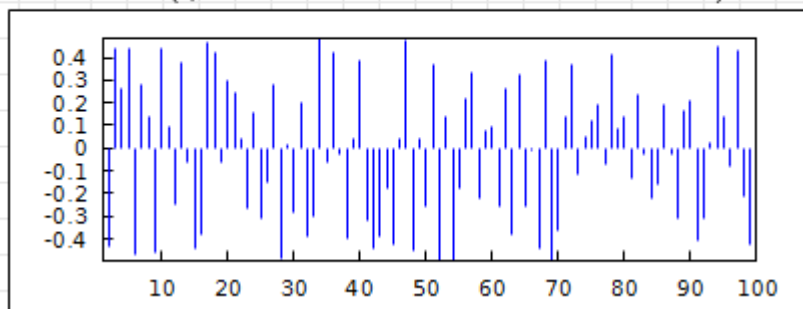
p2

Zufällige  $y$ -Werte, über laufender Nummer als  $x$ -Wert dargestellt

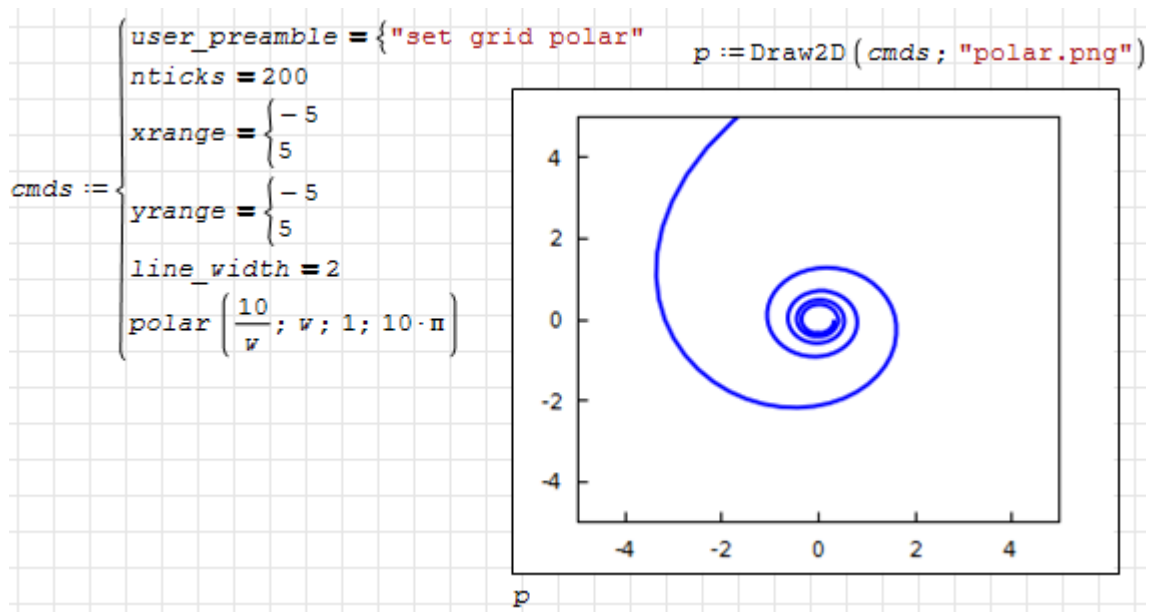
```

p := Draw2D({
  points_joined = impulses
  points (Random(100; 1) - 0,5);
})

```



`polar(r; phi; pmin; pmax)` 2D-Kurve, gegeben in Polarkoordinaten  $r(\varphi)$ .



`polygon(p1; p2; ...)` gefülltes Polygon. Die Argumente sind Koordinatenlisten für die einzelnen Punkte in der Form

$$2D: \begin{cases} x \\ y \end{cases} \quad \text{oder } 3D: \begin{cases} x \\ y \\ z \end{cases}$$

Sie können mit der Maximafunktion `args()` bequem aus einer Koordinatenmatrix  $M$  erzeugt werden: `polygon(args(M))`.

`transparent` ungefüllt oder gefüllt (hier kann keine Transparenz eingestellt werden)

`fill_color` Füllfarbe

`border` Rand zeichnen

`line_type` Linienart für den Rand

`line_width` Linienbreite für den Rand

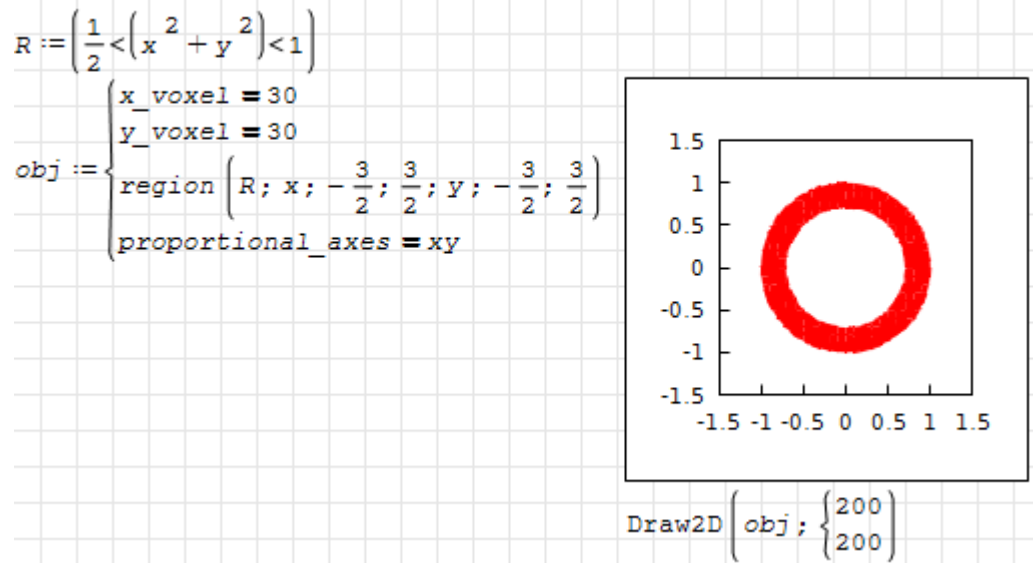
`key` Legendeneintrag

`color` Linienfarbe

`quadrilateral()` Viereck, Sonderfall von `polygon()`

`rectangle()` Rechteck, Sonderfall von `polygon()`

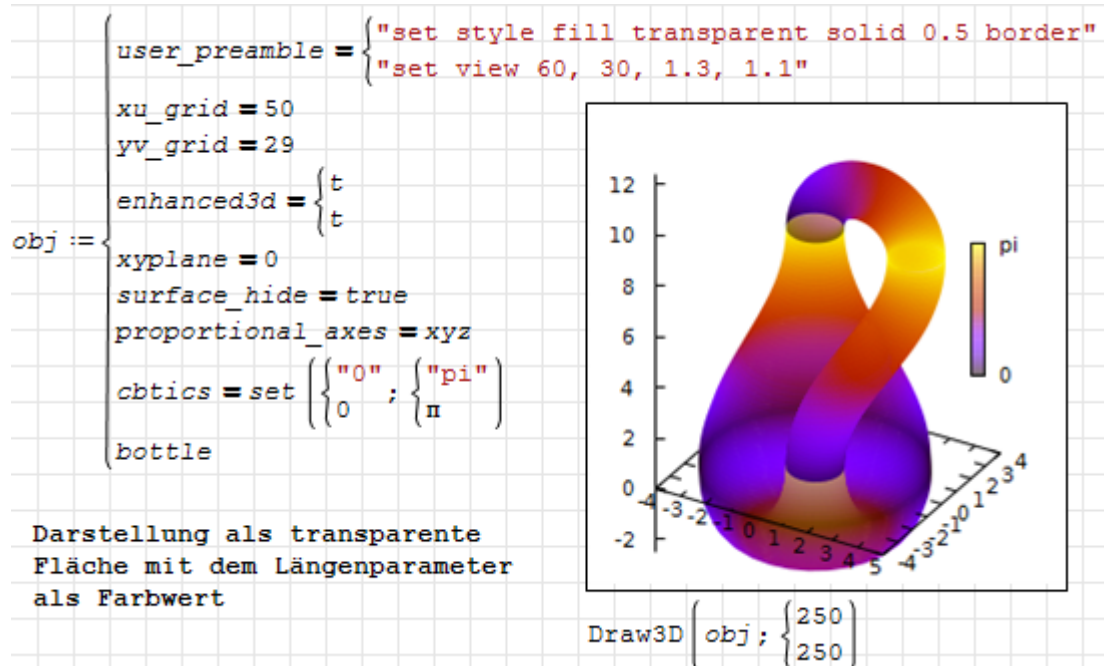
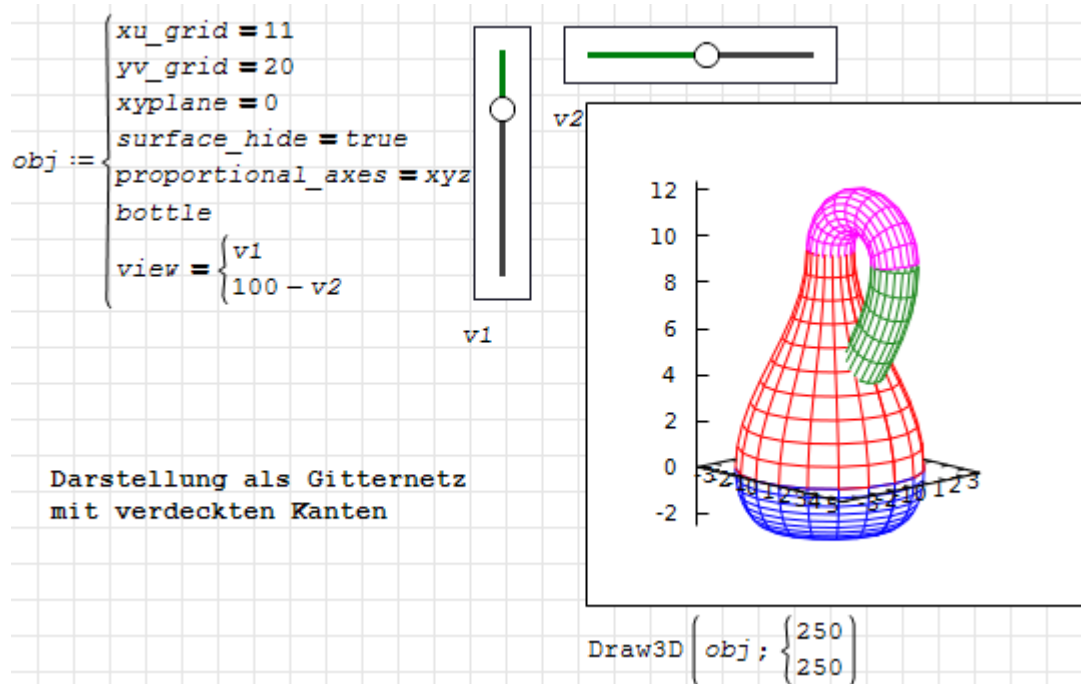
`region(Bedingung; x; xmin; xmax; y; ymin; ymax)` Füllung des Gebiets (ohne Randkontur) welches eine logische Bedingung erfüllt.



triangle() Dreieck

tube(x; y; z; r; p; min; max) Schlauchfläche, gegeben durch die Lage der Mittellinie  $x, y, z$  und den Radius  $r$  als Funktionen eines Längenparameters  $p$ . Hier als Beispiel die Kleinsche Flasche. Zunächst werden die Funktionen definiert. Den einzelnen Abschnitten werden Farben zugeordnet, damit man sie später im Bild eindeutig identifizieren kann.

```
a := 2,5  b := 1,5  c := 2  d := 3
{
  color = blue
  tube ( 0; 0; (-a)·sin(t); a + b·cos(t); t; 0; π/2 )
  tube ( 0; 0; (-a)·sin(t); a + b·cos(t); t; π/2; π )
bottle := {
  color = red
  tube ( 0; 0; d·t; a + b·cos(t); t; 0; π )
  color = forest_green
  tube ( c - c·cos(t); 0; d·t; a - b; t; 0; π )
  color = magenta
  tube ( c - c·cos(t); 0; d·π + c·sin(t); a - b; t; 0; π )
}
```



`vector(p; d)` Pfeil mit den Komponenten `d` beginnend am Startpunkt `p`, beide Argumente gegeben als Listen von 2 oder 3 Koordinaten

`head_angle (45)` Winkel zwischen Schaft und Spitze in  $^\circ$

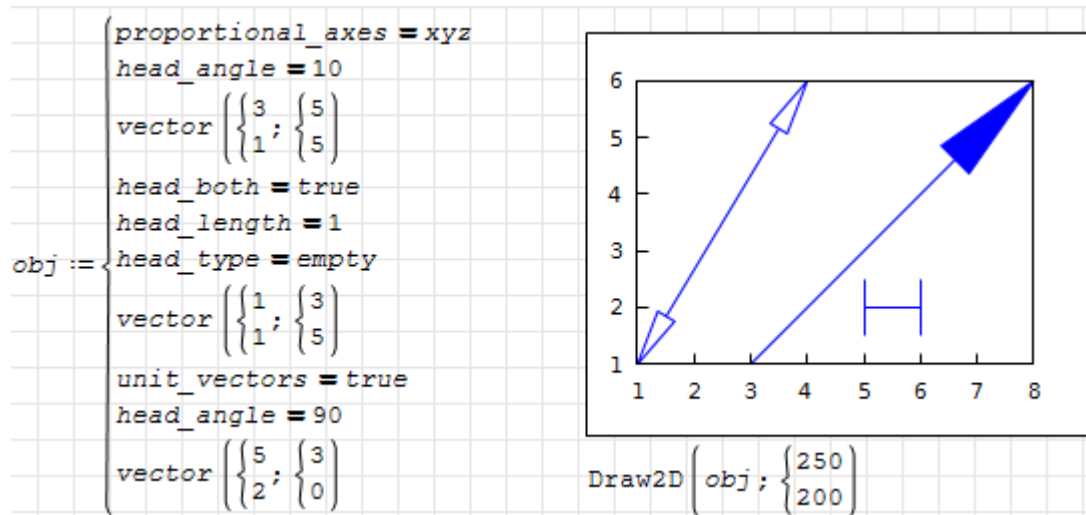
`head_both (false)` ein (`false`) oder zwei Spitzen (`true`)

`head_length (2)` Spitzenlänge in Einheiten der  $x$ -Achse

`head_type (filled)` Ausführung der Spitze: `filled`: gefüllt, `nofilled`: offen, `empty`: geschlossen, aber ungefüllt

`unit_vectors (false)` Falls `true`, wird der Vektor mit der Länge 1 gezeichnet.





### 8.1.6. Optionen

Die Optionen für die Funktionen Draw2D() und Draw3D() sind als logische Gleichungen Option=Wert anzugeben. Genau wie die Grafikobjekte können sie beliebig in Listen verschachtelt werden und gelten immer für die nachfolgenden Grafikobjekte oder global für das Diagramm.

| Option                             | (Voreinstellung): Bedeutung  |
|------------------------------------|--|
| adapt_depth                        | (10): Maximale Teilungstiefe der adaptiven Kurvenabstastung  |
| axis_3d                            | (true): Achsen im 3D zeichnen<br>false: keine Achse zeichnen   |
| axis_bottom/top<br>axis_left/right | analog für die Achsen im 2D  |
| background_color                   | (white): Hintergrundfarbe  |
| border                             | (true): Ränder von polygon, ellipse oder rectangle zeichnen  |
| capping                            | (false): Endflächen in tube()-Objekten. true, false oder Liste mit zwei Werten für Anfang und Ende.  |
| cbrange                            | (auto): Farbbereich (Liste mit min und max)  |
| cbtics                             | (auto): Marken am Farbbalken   |
| color                              | (blue): Farbe für Linien, Punkte, Kanten und Text, als Namen oder hexadezimal in der Form "#rrggbb"  |
| colorbox                           | (true): Farblegende ohne Beschriftung zeichnen<br>false: keine Farblegende zeichnen<br>text: Farblegende mit Beschriftung text zeichnen  |
| contour                            | (none): keine Höhenlinien<br>base: Höhenlinien auf der xy-Ebene<br>surface: Höhenlinien auf der Fläche<br>both: Höhenlinien auf der xy-Ebene und auf der Fläche<br>map: xy-Ebene und Blickrichtung senkrecht |
| contour_levels                     | (5): Zahl der Höhenlinien<br>Dreier-Liste mit Minimum, Schritt und Maximum<br>Satz (set) mit Werten  |

|                                |  |
|--------------------------------|--|
| <code>data_file_name</code>    | ("data.gnuplot"): Kurvendaten für Gnuplot  |
| <code>delay</code>             | (5): Zeitschritt in animierten gif-Dateien zwischen zwei Bildern   |
| <code>dimensions</code>        | (600x500): Liste mit Breite und Höhe   |
| <code>draw_realpart</code>     | (true): Realteil bei komplexen Funktionen darstellen.  |
| <code>enhanced3d</code>        | (none): keinen Farbverlauf verwenden, die Oberflächen werden als Gitter dargestellt (steuerbar mit <code>color</code> und <code>surface_hide</code> )<br>$\begin{cases} f(x, y, z) \\ x \\ y \\ z \end{cases}, \begin{cases} f(x, y) \\ x \\ y \end{cases}$ Färbung anhand der Koordinaten, 2D-Fall für <code>explicit()</code> und <code>elevation_grid()</code><br>$\begin{cases} f(u, v) \\ u \\ v \end{cases}, \begin{cases} f(u) \\ u \end{cases}$ Färbung anhand der Objektparameter<br>$\begin{cases} f(k) \\ k \end{cases}$ Färbung anhand der Reihenfolge (bei Punkten, <code>points()</code> )<br>true: entspricht Färbung gemäß der $z$ -Koordinate |
| <code>error_type</code>        | (y): oder <code>x</code> , <code>xy</code> oder <code>boxes</code> : Fehlerbalkenformat  |
| <code>file_name</code>         | ("maxima_out"): mit Endung gemäß Ausgabeformat   |
| <code>fill_color</code>        | (red): Füllfarbe für Polygone und explizite Funktionen $y(x)$ im 2D  |
| <code>fill_density</code>      | (0): Deckungsgrad der Füllfarbe in <code>bars()</code> -Objekten, zwischen 0 und 1   |
| <code>filled_func</code>       | (false): explizite Funktionen $y(x)$ werden nicht gefüllt<br>true: Füllung bis zur Unterkante des Diagramms<br>$g(x)$ Füllung zwischen $g(x)$ und $y(x)$   |
| <code>gnuplot_file_name</code> | ("maxout.gnuplot"): Kommandodatei für Gnuplot  |
| <code>gnuplot_term</code>      | (): Ausgabe-Format-Spezifikation (Text nach „ <code>set terminal</code> “)   |
| <code>gnuplot_out_file</code>  | ("maxplot.xxx"): Ausgabedateiname, <code>xxx</code> entspricht dem Dateiformat.  |
| <code>grid</code>              | (false): kein Gitterraster zeichnen<br>true: Gitter zeichnen   |
| <code>head_angle</code>        | (45): Pfeilwinkel an Vektoren (halber Öffnungswinkel in °)   |
| <code>head_both</code>         | (false): ob bei <code>vector()</code> -Objekten zwei Pfeilspitzen gezeichnet werden.   |
| <code>head_length</code>       | (2): Länge der Pfeilspitze in Einheiten der $x$ -Achse   |
| <code>head_type</code>         | (filled): $\blacktriangleright$ , <code>empty</code> $\triangleright$ oder <code>nonfilled</code> $\triangleright$   |

| <code>interpolate_color</code>                         | (false): Farb-Interpolation auf 3D Flächen wenn <code>enhanced3D</code> nicht false ist. false: keine Interpolation (einheitliche Farbe im Gitterkästchen). true: glatter Farbübergang. list (m,n): Interpolation mit m und n Punkten in u- und v-Richtung.   |                           |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
|--|---|---------------------------|-------------------------|------------------------|--------------------------|---------------------|---------------------------|--------------------------|----------------------------|---------------------------|-------|---|---|--------|-------|-----|---|--------------|---------|-----|---|-------------------|---------|-----|---|-------------------------|---------|-----|---|----------|-----------|--|--|
| <code>ip_grid</code>                                   | $\begin{cases} 50 \\ 50 \end{cases}$ Primäres Gitter für die Darstellung impliziter Funktionen  |                           |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>ip_grid_in</code>                                | $\begin{cases} 5 \\ 5 \end{cases}$ Sekundäres Gitter für die Darstellung impliziter Funktionen  |                           |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>key</code>                                       | (""): Legendeneintrag für das nachfolgende Grafikobjekt.  |                           |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>key_pos</code>                                   | (""): Legendenposition (" " entspricht <code>top_right</code> )<br><table border="1" style="margin-left: 20px;"> <tr> <td><code>top_left</code></td> <td><code>top_center</code></td> <td><code>top_right</code></td> </tr> <tr> <td><code>center_left</code></td> <td><code>center</code></td> <td><code>center_right</code></td> </tr> <tr> <td><code>bottom_left</code></td> <td><code>bottom_center</code></td> <td><code>bottom_right</code></td> </tr> </table>   | <code>top_left</code>     | <code>top_center</code> | <code>top_right</code> | <code>center_left</code> | <code>center</code> | <code>center_right</code> | <code>bottom_left</code> | <code>bottom_center</code> | <code>bottom_right</code> |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>top_left</code>                                  | <code>top_center</code>   | <code>top_right</code>    |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>center_left</code>                               | <code>center</code>   | <code>center_right</code> |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>bottom_left</code>                               | <code>bottom_center</code>  | <code>bottom_right</code> |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>label_alignment</code>                           | (center), left oder right: Ausrichtung von Texten   |                           |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>label_orientation</code>                         | (horizontal) oder vertical: Ausrichtung von Texten  |                           |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>line_type</code>                                 | (solid): Linienart. Optionen sind abhängig vom Bildformat.<br><table border="1" style="margin-left: 20px;"> <thead> <tr> <th>line_type</th> <th></th> <th>png</th> <th>svg</th> </tr> </thead> <tbody> <tr> <td>solid</td> <td>—————</td> <td>x</td> <td>x</td> </tr> <tr> <td>dots</td> <td>.....</td> <td>x</td> <td>x</td> </tr> <tr> <td>dashes</td> <td>- - -</td> <td>(x)</td> <td>x</td> </tr> <tr> <td>short_dashes</td> <td>- - - -</td> <td>(x)</td> <td>x</td> </tr> <tr> <td>short_long_dashes</td> <td>- - - -</td> <td>(x)</td> <td>x</td> </tr> <tr> <td>short_short_long_dashes</td> <td>- - - -</td> <td>(x)</td> <td>x</td> </tr> <tr> <td>dot_dash</td> <td>- . - . -</td> <td></td> <td></td> </tr> </tbody> </table> (x): nur im Maxima Draw2D() Bereich mit Einstellung <b>Diagram style &gt; term: png</b> | line_type                 |                         | png                    | svg                      | solid               | —————                     | x                        | x                          | dots                      | ..... | x | x | dashes | - - - | (x) | x | short_dashes | - - - - | (x) | x | short_long_dashes | - - - - | (x) | x | short_short_long_dashes | - - - - | (x) | x | dot_dash | - . - . - |  |  |
| line_type  |   | png                       | svg                     |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| solid  | —————   | x                         | x                       |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| dots   | .....   | x                         | x                       |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| dashes   | - - -   | (x)                       | x                       |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| short_dashes   | - - - -   | (x)                       | x                       |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| short_long_dashes                                      | - - - -   | (x)                       | x                       |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| short_short_long_dashes                                | - - - -   | (x)                       | x                       |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| dot_dash   | - . - . -   |                           |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>line_width</code>                                | (1): Linienbreite, > 0  |                           |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>logcb</code>                                     | (false) oder true: logarithmische Farbskala   |                           |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>logx/y/z</code><br><code>logx/y_secondary</code> | (false) oder true: logarithmische x-, y- oder z-Achse   |                           |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>nticks</code>                                    | (29): Punktezahl für die Kurvenabtastung  |                           |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>palette</code>                                   | (color): Standard-Farbpalette<br>gray: Graustufen<br>$\begin{cases} \text{Farbe}_1 \\ \text{Farbe}_2 \\ \vdots \\ \text{Farbe}_n \end{cases}$<br>false: Flächen nicht färben  |                           |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>point_size</code>                                | (1): Größe von <code>points()</code> -Symbolen  |                           |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>point_type</code>                                | (1): Symbol für <code>points()</code>   |                           |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |
| <code>points_joined</code>                             | (false), true oder impulses: Verbindungsart für <code>points()</code>   |                           |                         |                        |                          |                     |                           |                          |                            |                           |       |   |   |        |       |     |   |              |         |     |   |                   |         |     |   |                         |         |     |   |          |           |  |  |

|  |  |
|--|--|
| <code>proportional_axes</code>                             | (none), xy oder xyz: Achsen mit gleicher Skalierung  |
| <code>surface_hide</code>                                  | (false) oder true: Ausblendung verdeckter Flächen bei Linienrasterdarstellung  |
| <code>terminal</code>                                      | (screen): Separates Gnuplot-Fenster öffnet sich<br>png, pngcairo, jpg, gif, eps, svg, pdf,<br>animated_gif   |
| <code>title</code>   | (""): Überschrift für das Diagramm   |
| <code>transform</code>                                     | (none) Koordinatentransformation<br>$\begin{cases} f_1(x, y) \\ f_2(x, y) \\ x \\ y \end{cases} \text{ für 2D oder } \begin{cases} f_1(x, y, z) \\ f_2(x, y, z) \\ f_3(x, y, z) \\ x \\ y \\ z \end{cases} \text{ für 3D}$ |
| <code>transparent</code>                                   | (false) oder true: ob die Rechtecke, Polygone oder Ellipsen gefüllt werden sollen (bei false werden sie gefüllt)   |
| <code>unit_vectors</code>                                  | (false) oder true: Vektoren in originaler Länge oder als Einheitsvektoren zeichnen.  |
| <code>user_preamble</code>                                 | ("") Zeichenkette oder Liste von Zeichenketten, die unmittelbar vor dem eigentlichen Diagrammbefehl an Gnuplot übergeben werden. Beispiele sind in Abschnitt 8.1.7 angegeben.  |
| <code>view</code>  | $\begin{pmatrix} 60 \\ 30 \end{pmatrix}$ Blickrichtung   |
| <code>wired_surface</code>                                 | (false) oder true: Gitternetz auf eingefärbten Flächen anzeigen oder nicht.  |
| <code>x/y/z_voxel</code>                                   | (10) Bereichsteilungen bei implizit gegebenen Flächen  |
| <code>x/y/zaxis</code>                                     | (false) oder true: Anzeige der entsprechenden Achse  |
| <code>x/y/zaxis_color</code>                               | (black) Farbe der entsprechenden Achse   |
| <code>x/yaxis_secondary</code>                             | (false) oder true: Nutzung der zweiten x- oder y-Achse (oben oder rechts) für die nachfolgenden Grafikobjekte  |
| <code>x/y/zaxis_type</code>                                | (dots) oder solid: Linienart der entsprechenden Achse  |
| <code>x/y/zaxis_width</code>                               | (1): Linienbreite der entsprechenden Achse   |
| <code>x/y/zlabel</code><br><code>x/ylabel_secondary</code> | (""): Achsenbeschriftung.  |
| <code>x/y/zrange</code>                                    | (auto) oder $\begin{cases} \min \\ \max \end{cases}$ : Achsenbereich   |
| <code>x/yrange_secondary</code>                            | (auto) oder $\begin{cases} \min \\ \max \end{cases}$ : Achsenbereich der Zweitachsen   |
| <code>x/y/ztics</code><br><code>x/ytics_secondary</code>   | (auto): automatische Achsmarken<br>none: keine Achsmarken<br>Zahl: Schrittweite der Achsmarken<br>$\begin{cases} \min \\ \text{Schritt} \\ \max \end{cases}$   |

|  |   |
|--|---|
|  | Satz (set) von Zahlen: Explizite Vorgabe der Achsmarkenpositionen<br>Satz (set) von Listen $\left\{ \begin{array}{l} \text{"text"} \\ \text{pos} \end{array} \right.$ |
| x/y/ztics_axis<br>x/ytics_secondary_axis | (false): Achsmarken am Rand<br>true: Achsmarken an der Achse  |
| x/y/ztics_rotate                         | (false) oder true: Drehen der Achsmarkenbeschriftungen um 90°   |
| x/ytics_rotate_secondary                 | (false) oder true: Drehen der Achsmarkenbeschriftungen um 90°   |
| xu_grid                                  | (30): Werteanzahl in die erste Koordinatenrichtung  |
| xyplane                                  | (false): xy-Koordinatenebene automatisch plazieren<br>Zahl: bei vorgegebenem z-Wert plazieren   |
| yv_grid                                  | (30): Werteanzahl in die zweite Koordinatenrichtung   |

| Bezeichnung          | Kennzahl | Symbol |
|----------------------|----------|--------|
| none                 | -1       | keines |
| dot                  | 0        | .      |
| plus                 | 1        | +      |
| multiply             | 2        | ×      |
| asterisk             | 3        | ★      |
| square               | 4        | □      |
| filled_square        | 5        | ■      |
| circle               | 6        | ○      |
| filled_circle        | 7        | ●      |
| up_triangle          | 8        | △      |
| filled_up_triangle   | 9        | ▲      |
| down_triangle        | 10       | ▽      |
| filled_down_triangle | 11       | ▼      |
| diamant              | 12       | ◇      |
| filled_diamant       | 13       | ◆      |

Tabelle 8.4.: Punktarten (Werte für `point_type`)

**Farbspezifikation** Die Farbwerte für die Optionen `background_color`, `color`, `palette` und `x/y/zaxis_color` können als Wort oder als RGB-Wert (Zeichenkette) im Format `"#rrggbb"` angegeben werden.

Die folgenden Farben sind vordefiniert. Die Wörter können als Zeichenketten oder direkt als Variablenname angegeben werden, solange diese Namen im Rechenblatt nicht definiert werden.

```
white black gray0 grey0 gray10 grey10 gray20 grey20 gray30
grey30 gray40 grey40 gray50 grey50 gray60 grey60 gray70
grey70 gray80 grey80 gray90 grey90 gray100 grey100 gray grey
light_gray light_grey dark_gray dark_grey red light_red dark_red
yellow light_yellow dark_yellow green light_green dark_green
spring_green forest_green sea_green blue light_blue dark_blue
```

```

midnight_blue navy medium_blue royalblue skyblue cyan light_cyan
dark_cyan magenta light_magenta dark_magenta turquoise
light_turquoise dark_turquoise pink light_pink dark_pink
coral light_coral orange_red salmon light_salmon dark_salmon
aquamarine khaki dark_khaki goldenrod light_goldenrod
dark_goldenrod gold beige brown orange dark_orange violet
dark_violet plum purple

```

### 8.1.7. Nützliche Befehle für user\_preamble

Skalierung von 3D-Diagrammen

```
set view <zenit> <azimut> <plotscale> <zscale>
```

Farbstufen in 3D-Plots

```
set palette maxcolors n
```

Die Linienart für das Gitter (grid) kann mit Gnuplot-Preamble-Befehlen gesetzt werden, z.B.

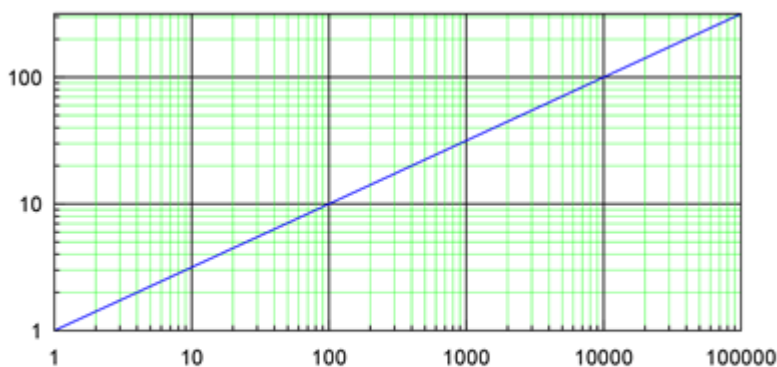
```
user_preamble="set grid ls -1"
```

The following example sets Mathcad-like line styles for the major and the minor grid:

```

cmd := {
  user_preamble = {
    "set style line 12 lt rgb '#00ff00' lw -1"
    "set style line 13 lt rgb '#000000' "
    "set grid linestyle 13 ,linestyle 12 "
    "set mxtics 0.1"
    "set mytics 0.1"
  }
  logx = true
  logy = true
  explicit (sqrt(x); x; 1; 100000)
  grid = true
}

```



```
Draw2D ( cmd; "pdf"; { 400
                    200 } )
```

Add space around the objects in autoscaled plots. The first version takes numbers in axes units, the second takes fractions of the plot range.

```
user_preamble="set offsets <left>, <right>, <bottom>, <top>
```

```
user_preamble="set offsets graph <left>, <right>, <bottom>, <top>
```

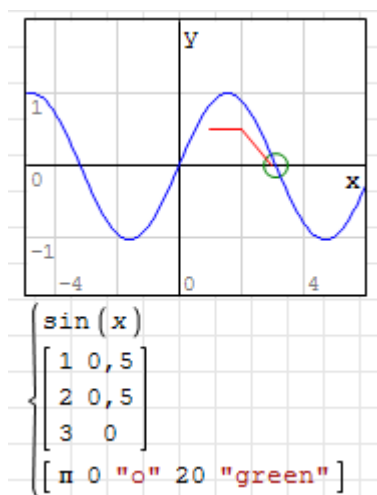
## 8.2. 2D-Diagramme (Standard)

2D-Diagrammbereiche werden erzeugt mit


- @
- **Einfügen > Diagramm > Zweidimensional**

Diagrammbereiche können vier Arten von Ausdrücken anzeigen:

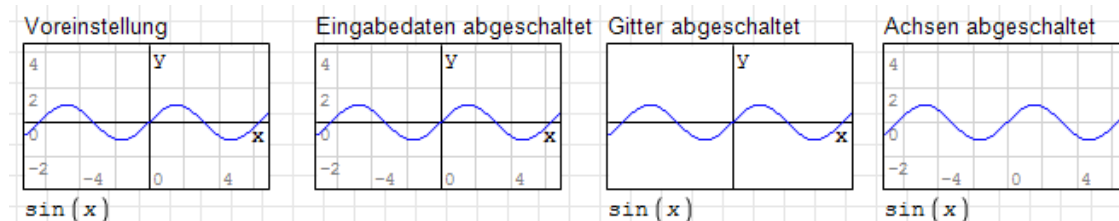
- Funktionen, als Argument wird ein dimensionsloses  $x$  bereitgestellt, als Ergebnis wird eine dimensionslose Zahl erwartet.
- 2-spaltige Matrizen, die einen Polygonzug definieren [ $x$   $y$ ].
- 3-spaltige Matrizen, die zeilenweise einen Text (Zeichenkette) anzeigen [ $x$   $y$  Text]. Größe und Farbe haben dann die voreingestellten Standardwerte.
- 5-spaltige Matrizen, die zeilenweise einen Text anzeigen [ $x$   $y$  Text Größe Farbe].



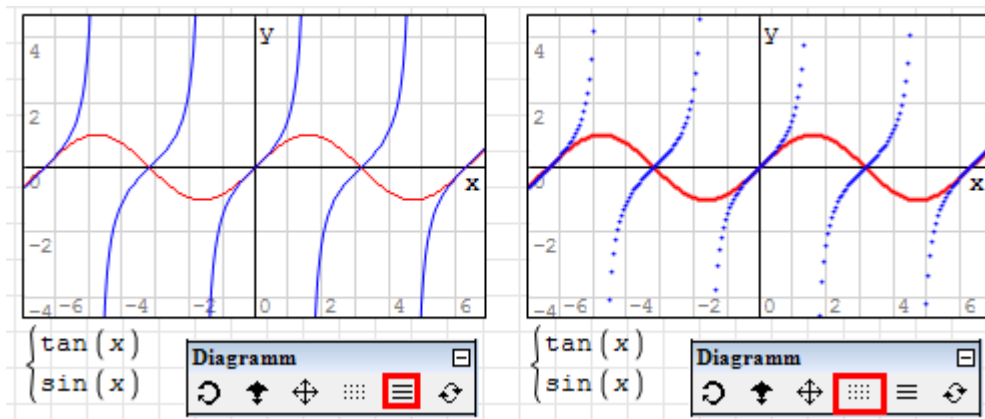
Alle zu plottenden Ausdrücke können mit der Variable  $t$  parametrisiert sein. Animierte Plots werden erzeugt, indem die Variable  $t$  die Werte eines beliebigen Vektors durchläuft, der im Rechenblatt definiert ist (siehe Abschnitt 8.2.6).

Wenn mehrere Ausdrücke in einem Bild dargestellt werden sollen, sind diese in einer Liste  zu kombinieren.

Im Kontextmenü kann man das Gitter und die Achsen abschalten.



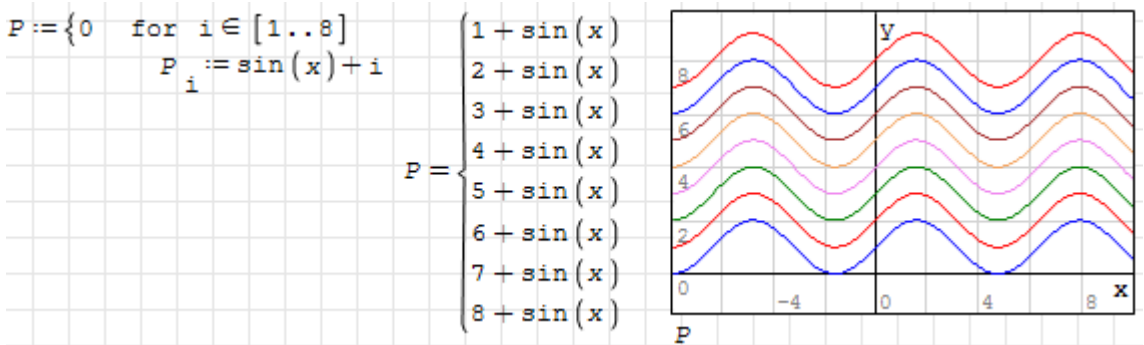
Das Aussehen der Kurven kann man durch Umschalten zwischen Punktdarstellung und Liniendarstellung im Panel „Grafik“ der Seitenleiste beeinflussen:



### 8.2.1. Funktionen

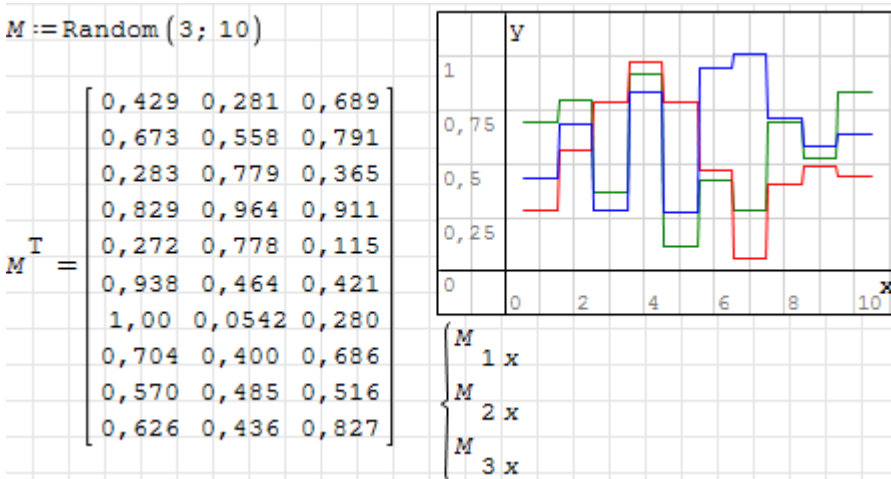
Funktionen werden als Polygonzug dargestellt, dessen  $x$ -Werte gleichmäßig über den Wertebereiche (sichtbarer Achsenbereich) verteilt sind und dessen  $y$ -Werte durch Aufruf der Funktion an diesen  $x$ -Werten berechnet werden.

Um mehrere Funktionen gleichzeitig darzustellen, ordnet man sie in einer Liste an. Für die Linien werden sechs Farben zyklisch verwendet.



$P$  wird als leere Liste initialisiert, denn sonst würde durch Zuweisung an  $P_i$  ein Vektor erzeugt werden.

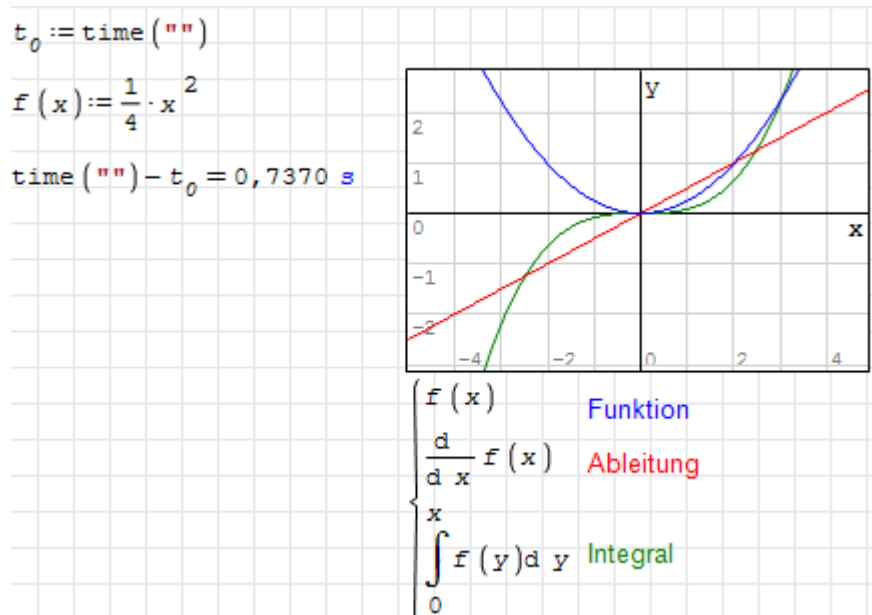
Wenn man in einem Diagramm  $x$  als Vektorindex verwendet, erhält man ein Stufendiagramm:



In Diagrammen kann man auch Integrale und Ableitungen verwenden. Die Integration wird allerdings für jeden Bildpunkt numerisch ausgeführt, deshalb kann es etwas dauern.

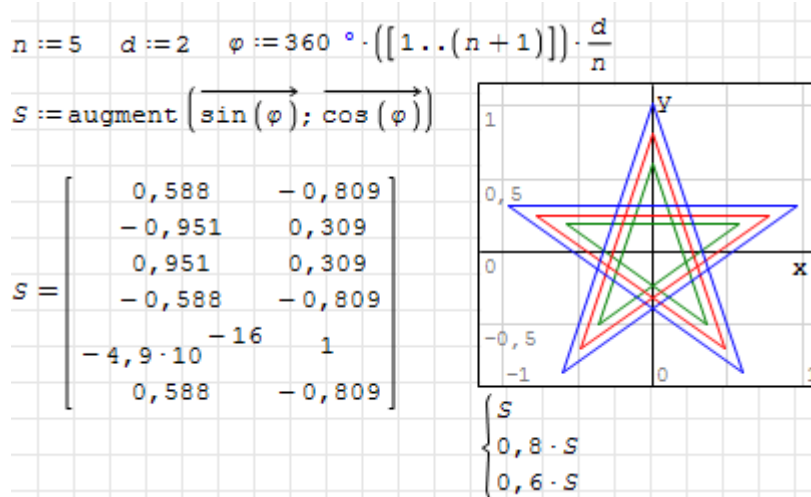


Als Ersatz für die nicht vorhandene Legendenfunktion des Standard-Diagrammbereichs mussten hier farbige Textbereiche erhalten:



### 8.2.2. Polygonzüge

SMath kann Polygonzüge darstellen, die als zweispaltige Matrizen gegeben sind, jede Zeile steht für einen Punkt. Die 1. Spalte enthält die  $x$ -Werte, die zweite Spalte die  $y$ -Werte.



### 8.2.3. Text

Einzelne Zeichen oder Textzeilen können im Plot frei platziert werden. Jede Zeile einer fünfspaltigen Matrix definiert einen Texteintrag:

$$\left[ x \ y \ \text{Text} \ s \ \text{Farbe} \right]$$

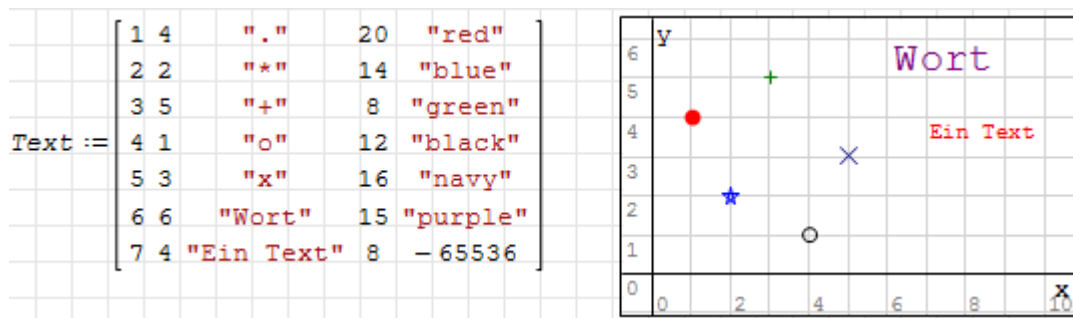
$x, y$  Koordinaten

Text darzustellender Text (als Zeichenkette anzugeben)

$s$  Schriftgröße (Einheit Punkt)

Farbe    Textfarbe (als Zeichenkette anzugeben)

Die Koordinaten geben die Position der linken oberen Ecke des Textfeldes an. Die Zeichen  $\cdot$   $*$   $+$   $\circ$   $\times$  werden bei  $x, y$  zentriert, wenn sie einzeln vorkommen. Damit können sie auf einfache Weise als Punktsymbole in Diagrammen verwendet werden.



### 8.2.4. Farben

Es gibt mehrere Wege, Farben für Textobjekte in Diagrammen zu spezifizieren:

1. Zeichenkette mit HTML-Farbnamen, z.B. "red"
2. Zeichenkette mit RGB-Hexcode, z.B. "#FF0000". Je zwei Bytes codieren die Werte R, G und B.
3. Zeichenkette mit  $\alpha$ RGB-Hexcode, z.B. "880000FF". Je zwei Bytes codieren die Werte  $\alpha$  (Deckkraft), R, G und B.
4. Dezimale 8-Byte-Ganzzahlen entsprechend den Hexcodes. Bei 6-Byte-Codes wird  $\alpha = 1$  ("FF") ergänzt. Auf diese Weise können Farbwerte auch berechnet werden.

SMath kennt die HTML-Farben, diese sind in Abbildung 8.4 dargestellt. Die Namen werden intern von Leerzeichen befreit und in Kleinbuchstaben umgewandelt, "darkblue", "DarkBlue", "DARKBLUE" und "dark blue" bezeichnen also alle die gleiche Farbe.

Das folgende SMATH-Blatt ordnet die Farbnamen jeweils passend eingefärbt in einer rechteckigen Matrix (Spaltenzahl  $n_c$ ) an. Der Operator  $\lfloor \cdot \rfloor$  kennzeichnet die Abrundung auf ganze Zahlen.

```

F := importData_ods ("Farben.ods"; "Farben")    n := length(F)
n_c := 4    P := matrix(0; 5)    for i ∈ [1..n]
    r := ⌊ i / n_c ⌋
    c := mod(i; n_c) + 1
    P := eval( stack( P; [ c - r F_i 10 F_i ] ) )

```

Abbildung 8.4 zeigt einen Plot der Liste P. Das Gitter und die Achsen wurden in diesem Bild ausgeblendet.

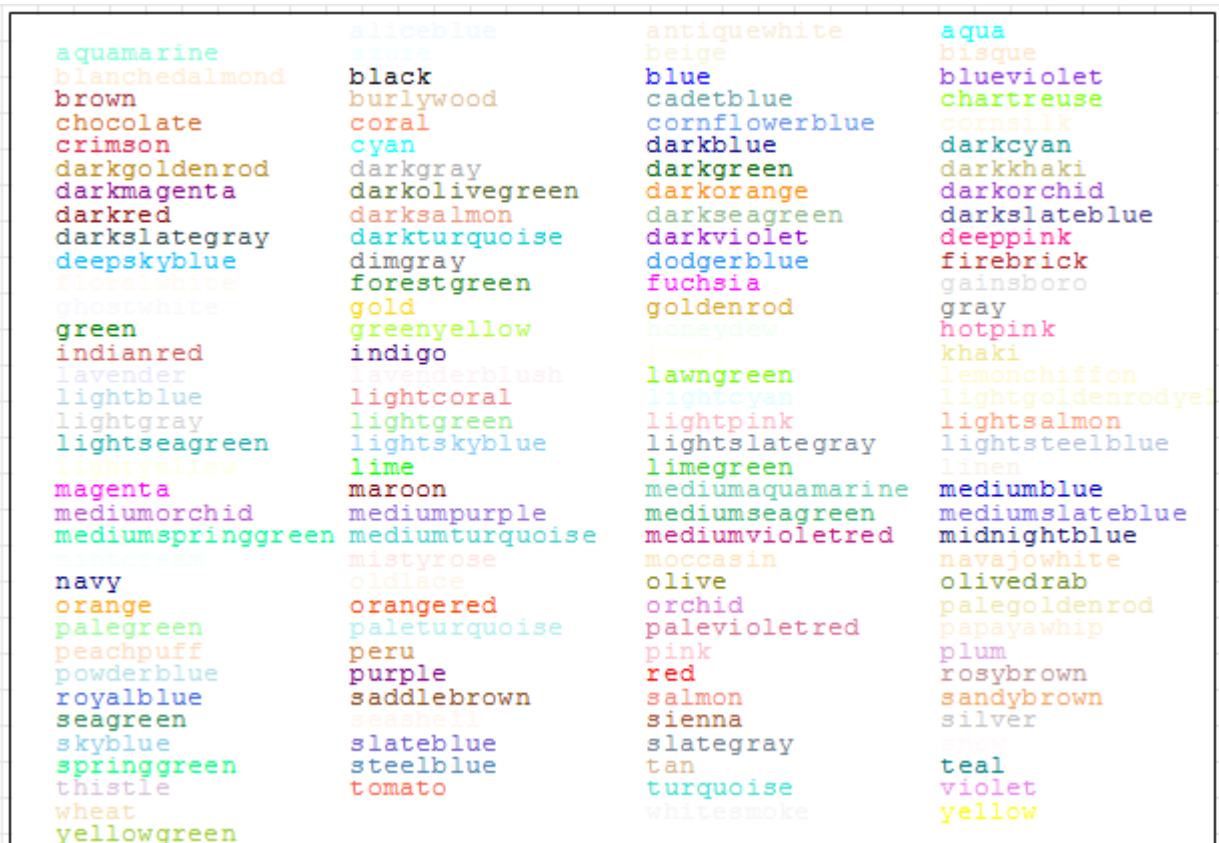




Abbildung 8.4.: Vordefinierte Farben für die Darstellung von Symbolen und Texten in 2D-Plots. Die Liste ist dem [Wiki-Eintrag zu Grafik \(englisch\)](#) entnommen. Die dortige Liste wurde in eine Textdatei kopiert. Leider gibt es keine Möglichkeit, Zeichenketten aus Textdateien einzulesen, die Funktion `importData()` kann nur Zahlenwerte lesen. Daher musste die Textdatei in `Farben.csv` umbenannt, mit Excel geöffnet und als ods-Datei wieder abgespeichert werden.

Zur 2D-Grafik gibt es auch einen umfangreichen [englischen Wiki-Eintrag](#)<sup>3</sup>.

### 8.2.5. Achsengrenzen ändern

Mauszeiger auf Diagrammbereich:

- Ziehen: Achsen verschieben
- Scrollen: Achsen skalieren
-  -Scroll: nur  $x$ -Achse skalieren
-  -Scroll: nur  $y$ -Achse skalieren

Es gibt keine Möglichkeit, die Achsengrenzen als Werte vorzugeben.

<sup>3</sup><http://smath.info/wiki/Graphs.ashx>

### 8.2.6. Animationen steuern

Wenn der darzustellende Ausdruck die Variable  $t$  enthält, kann SMath die Werte dafür aus einem vorhandenen Spaltenvektor entnehmen und die dadurch entstehende Bildsequenz als Animation abspielen.

Der Wertevektor muss als Variable definiert sein. Mit **Diagramm** > **Kontextmenü** (**rechte Maustaste**) > **Animieren** wird der Vektor ausgewählt. Angeboten wird alles, was definiert ist.

Ebenfalls im Kontextmenü kann man unter „Bildfrequenz“ die Grundeinstellungen für die Animation machen:

- Einstellung der Bildfrequenz (auf einzelne, vordefinierte Werte)
- Animationsmodus (unbegrenzt, keine Animation, einmalig)

– unbegrenzt: Die Bildfolge wird beliebig oft wiederholt


$$\left[ x \quad y \quad \text{Text} \quad s \quad \text{Farbe} \right]$$

$x, y$       Koordinaten

Text      darzustellender Text (als Zeichenkette anzugeben)

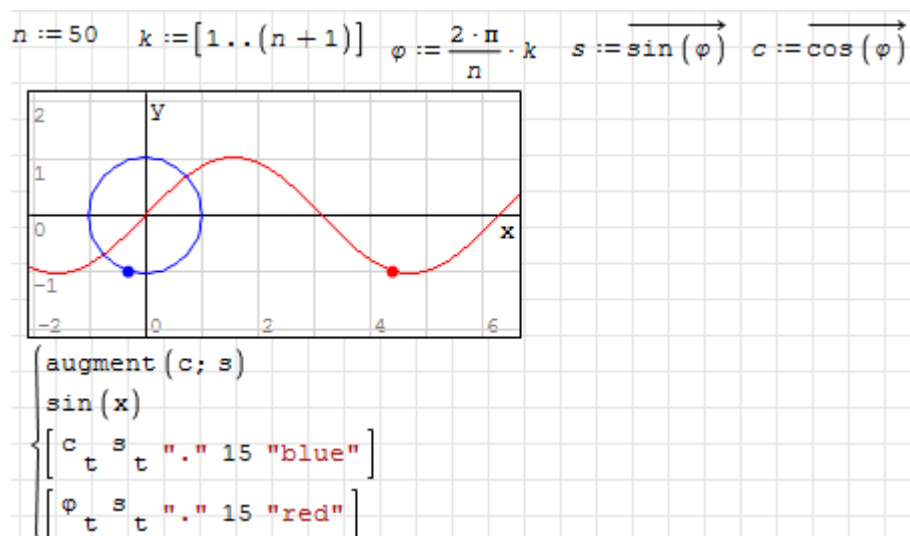
$s$         Schriftgröße (Einheit Punkt)

Farbe     Textfarbe (als Zeichenkette anzugeben)

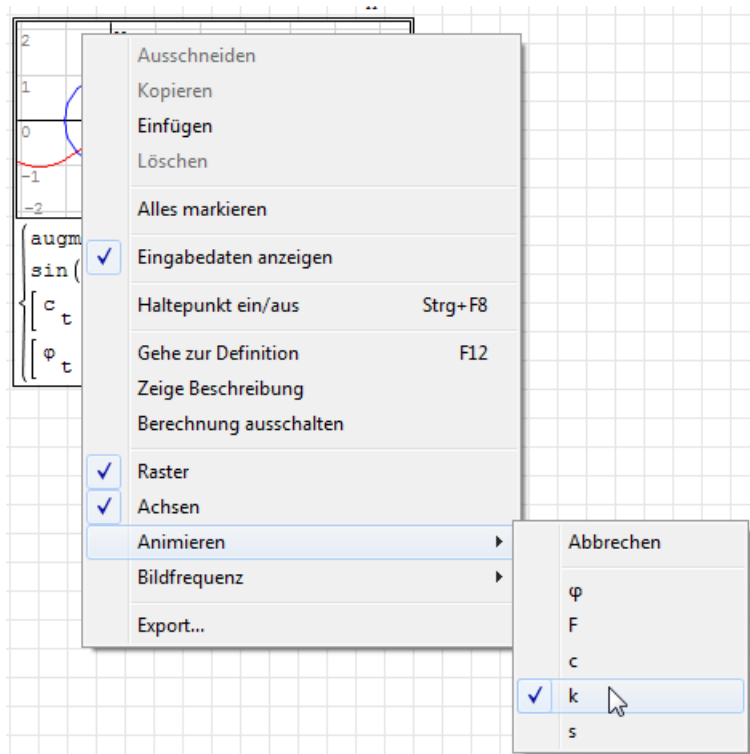
– einmalig: Die Bildfolge wird nur einmal wiederholt (nach jeder Neuberechnung), sie kann also mit  neu gestartet werden.

– keine Animation: man kann mit **[Alt]** - **[→]** und **[Alt]** - **[←]** vorwärts und rückwärts einzeln durch die Bilder schalten.

Hier ein Beispiel: Dargestellt sind eine Sinus-Kurve und der Einheitskreis, darauf bewegen sich Punkte mit  $\varphi = 0 \dots 2\pi$ .



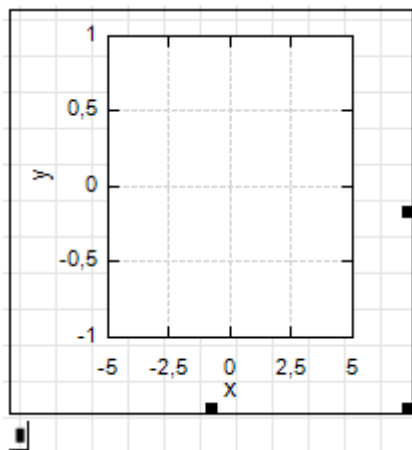
Im Kontextmenü des Diagramms wird der Wertevektor für  $k$  ausgewählt:



### 8.3. X-Y Plot (erweiterte 2D-Diagrammfunktion)

Das Plug-in *X-Y Plot Region* stellt einen alternativen Diagrammbereich bereit, der den internen Standard-Diagrammbereich ergänzt.

Ein X-Y Plot-Bereich kann über das **Einfügen** > **Diagramme** > **X-Y Plot** eingefügt werden.



Dieser Bereich bietet zusätzliche Möglichkeiten:

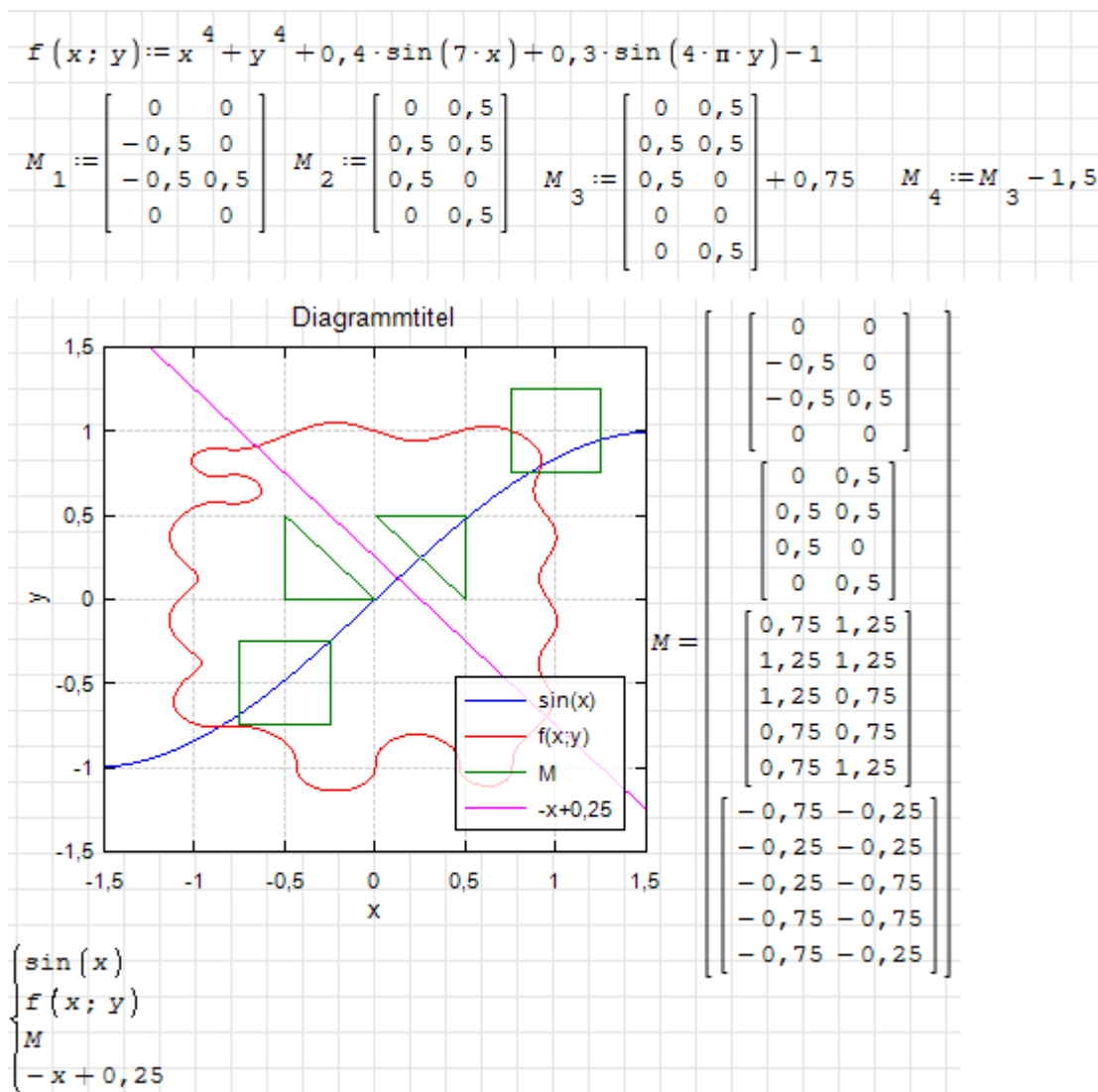
- Darstellung implizit gegebener Kurven  $f(x, y) = 0$
- Darstellung unterbrochener Linienzüge (Vektoren von Wertepaar-Matrizen)
- Frei formatierbare Achsenbeschriftungen und Diagrammtitel (ohne Formelformatierung)
- Frei einstellbare Linien- und Markerform für Funktionen und Wertematrizen.
- Frei einstellbarer Achsenbereich und Gitternetzweite

- Viele Einstellungen auch vom Rechenblatt aus steuerbar
- Zweite y-Achse
- Legende
- Diagrammtitel

Einschränkungen gegenüber dem Standard-2D-Diagramm sind

- Keine symbolische Auswertung der Ausdrücke vor dem Darstellen (kann aber durch Zuweisung an eine Variable erzwungen werden)
- Keine Animationsfunktion.

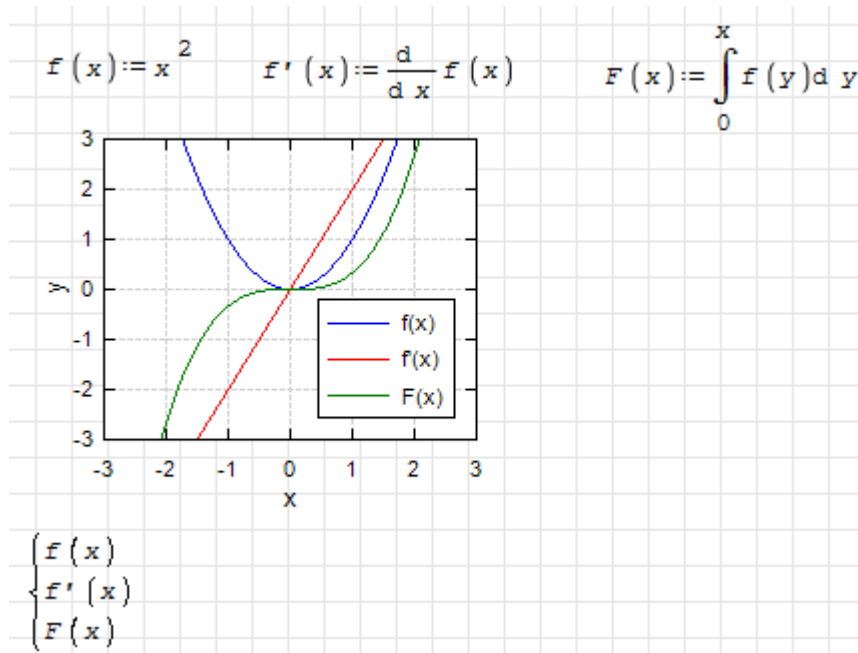
Hier ein Beispiel für die Darstellungsmöglichkeiten.



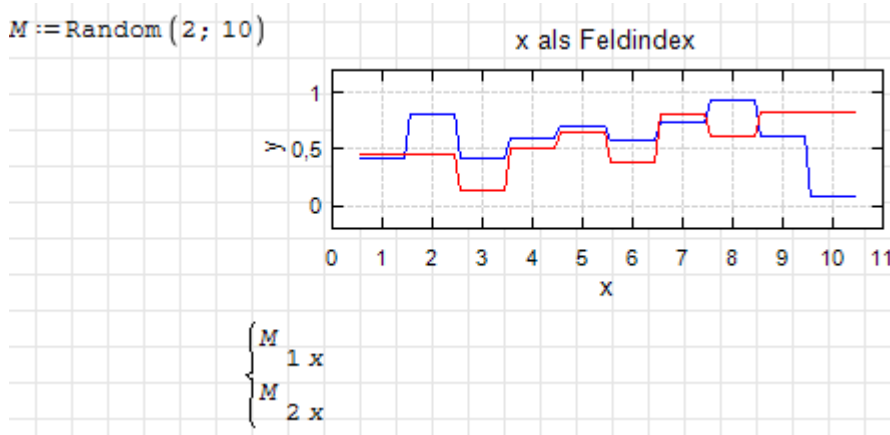
### 8.3.1. Explizite Funktionen

Explizite Funktionen sind Ausdrücke, die den Funktionswert direkt liefern. Mehrere Funktionen können wie im Standard-Diagrammbereich durch Listen kombiniert werden.

Mitunter müssen die Funktionen vorher definiert werden, können also nicht direkt in den Platzhalter geschrieben werden:



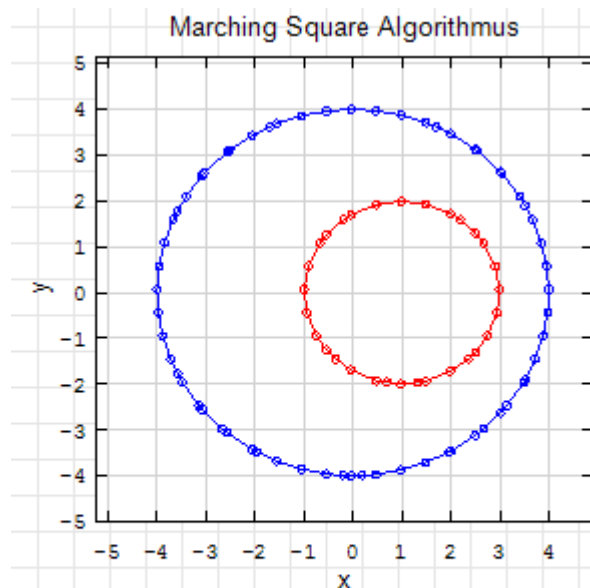
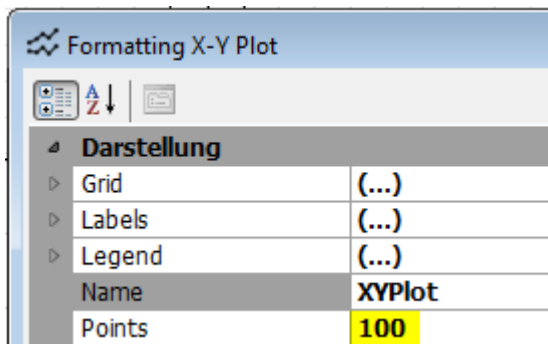
Die unabhängige Variable kann auch als Index an Feldern benutzt werden



### 8.3.2. Implizite Funktionen

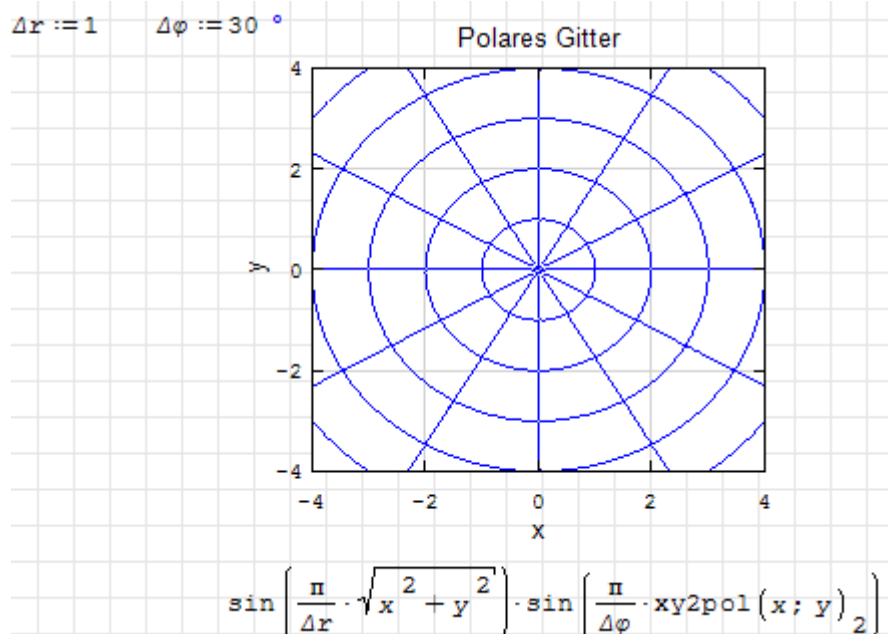
Implizite Funktionen erkennt der X-Y Plot-Bereich an zwei darin vorkommenden undefinierten Variablen.

Die Funktion ist als Ausdruck mit zwei undefinierten Variablen zu definieren, wobei die erste (für die x-Achse)  $x$  oder  $t$  heißen muss. Die Nullstellen dieses Ausdrucks werden dargestellt. Dafür wird der sogenannte **Marching-Squares-Algorithmus** verwendet. Dabei fallen nur Punkte auf einem regelmäßigen Gitter an, dessen Dichte mit der Format-Einstellung „Points“ vorgegeben wird. Hier ein Beispiel (Points = 11)



$$\begin{cases} x^2 + y^2 - 4^2 \\ (x-1)^2 + y^2 - 2^2 \end{cases}$$

Ein weiteres Beispiel mit einer komplexeren Funktion: die Nullstellen bilden ein Polarkoordinatenraster mit wählbarem Radius- und Winkelschritt.





### 8.3.3. Text

Einzelne Zeichen oder Textzeilen können ähnlich wie beim Standard-Diagrammbereich frei platziert werden. Jede Zeile einer fünfspaltigen Matrix definiert einen Texteintrag:

$$[ x \ y \ \text{Text} \ s \ \text{Farbe} ]$$

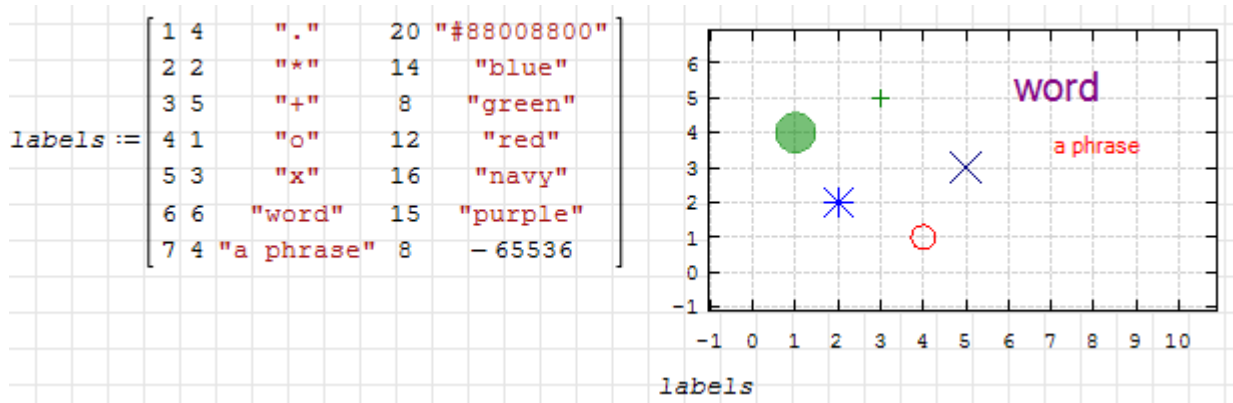
$x, y$       Koordinaten

Text        darzustellender Text (als Zeichenkette anzugeben)

$s$          Schriftgröße (optional, Einheit Punkt)

Farbe      Textfarbe (optional, siehe Abschnitt 8.2.4)

Die Koordinaten geben die Position der linken oberen Ecke des Textfeldes an. Die Zeichen  $.$   $*$   $+$   $o$   $x$  werden bei  $x, y$  zentriert, wenn sie einzeln vorkommen. Damit können sie auf einfache Weise als Punktsymbole in Diagrammen verwendet werden.



Wenn die Farbspalte fehlt, kann die Farbe im Formatdialog geändert werden (für alle in einer Matrix definierten Texte gleichzeitig)

Wenn auch die Größenspalte fehlt, ist für die Zeichen  $.$   $*$   $+$   $o$   $x$  die Einstellung im Formatdialog wirksam, ansonsten hat der Text dann Standardgröße.

### 8.3.4. Geometrische Grundformen

Im X-Y Plot-Bereich sind einige Grundformen definiert, bei denen man Position, Größe, Linienart, Linienstärke, Linienfarbe und Füllfarbe vorgeben kann. Die Grundformen werden durch Spaltenvektoren definiert (hier als transponierter Zeilenvektor geschrieben):

$$[ \text{Typ} \ \text{Punkte} \ \text{Linienfarbe} \ \text{Linienart} \ \text{Linienstärke} \ \text{Füllfarbe} ]^T$$

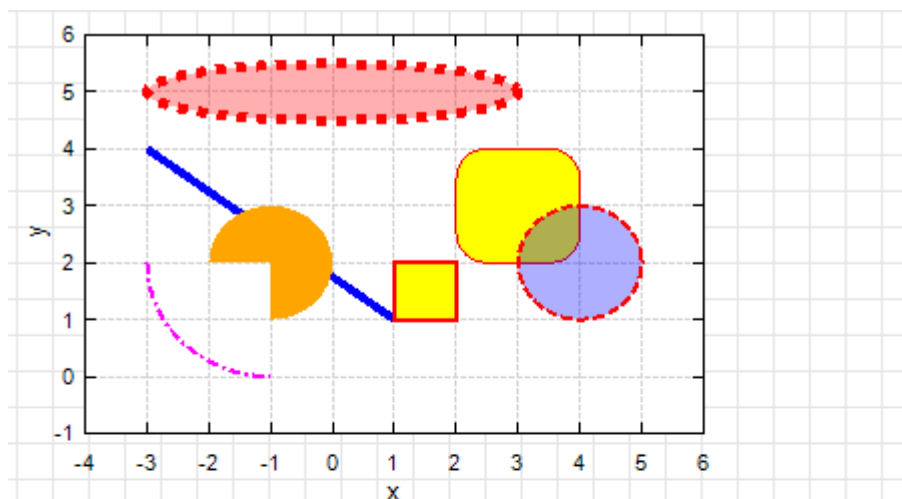
**Typ** Schlüsselwort für die Grundform

**Punkte** Koordinatenmatrix oder -vektor, abhängig vom Typ

**Linienfarbe** Zeichenkette oder Zahl. 0 bedeutet: Keine Linie

**Linienart** Schlüsselwort: "solid", "dash", "dot", "dashdot" oder "dashdotdot"

**Füllfarbe** Zeichenkette oder Zahl. Fehlt die Angabe, bleibt das Objekt ungefüllt



```

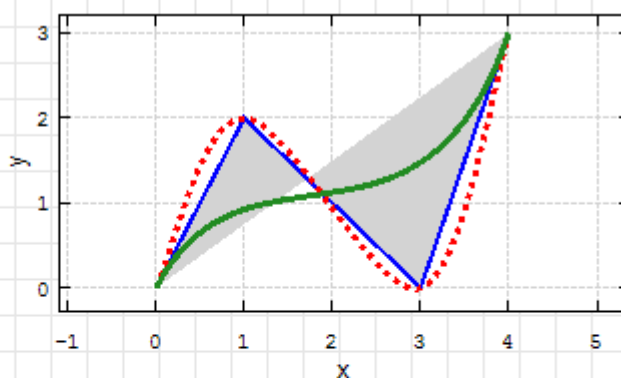
[
  ["line" [1 1 -3 4]T "blue" "solid" 4]T
  ["rect" [1 1 1 1]T "red" "solid" 2 "yellow"]T
  ["roundrect" [2 2 2 2 0,5]T "red" "solid" 1 "yellow"]T
  ["circle" [4 2 1]T "red" "dash" 2 "#50000ff"]T
  ["ellipse" [0 5 3 0,5]T "red" "dot" 5 "#50ff0000"]T
  ["arc" [-1 2 2 2 180 ° 90 °]T "magenta" "dashdot" 2]T
  ["pie" [-1 2 1 1 (-90) ° 270 °]T "" "" 0 "orange"]T
]

```

```

points :=
[
  [0 0]
  [1 2]
  [3 0]
  [4 3]
]

```



```

[
  ["polygon" points "" "" 0 "lightgray"]T
  ["polyline" points "blue" "solid" 2]T
  ["spline" points "red" "dot" 3]T
  ["bezier" points "forestgreen" "solid" 3]T
]

```

| Objekt                        | Koordinatenmatrix  | Füllung |
|-------------------------------|--|---------|
| line                          | $\begin{bmatrix} x_1 & y_1 & x_2 & y_2 \end{bmatrix}^T$                              |         |
| rect                          | $\begin{bmatrix} x_1 & y_1 & \Delta x & \Delta y \end{bmatrix}^T$                    | x       |
| roundrect                     | $\begin{bmatrix} x_1 & y_1 & \Delta x & \Delta y & r \end{bmatrix}^T$                | x       |
| circle                        | $\begin{bmatrix} x_c & y_c & r \end{bmatrix}^T$                                      | x       |
| ellipse                       | $\begin{bmatrix} x_c & y_c & r_x & r_y \end{bmatrix}^T$                              | x       |
| arc<br>pie                    | $\begin{bmatrix} x_c & y_c & r_x & r_y & \varphi_1 & \varphi_2 \end{bmatrix}^T$      | x       |
| polygon<br>polyline<br>spline | $\begin{bmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \end{bmatrix}^T$ | x       |
| bezier                        | $\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \end{bmatrix}^T$     |         |

Tabelle 8.5.: Belegung der Koordinatenmatrizen in X-Y Plot-Geometrieobjekten

### 8.3.5. Mausfunktionen im X-Y Plot

**Doppelklick links:** Öffnen des Format-Editors (alternativ zu **Kontextmenü** > **Format**)

**Klick links:** Aktivierung des Diagramms für die Bearbeitung (Größe ändern, Achsen mit der Maus skalieren und verschieben, Zugriff auf das Kontextmenü).

**Ziehen an den Randmarken:** Größe der Diagrammfläche ändern, dabei bleiben die Achsgrenzen erhalten, d.h. die Achsen werden entsprechend gedehnt oder gestaucht.

**Mausrad:** Skalierung der Achsen (x und y gleichzeitig). Eine eventuell aktive y2-Achse ist davon nicht betroffen.

**Shift-Mausrad:** Nur x-Achse zoomen

**Strg-Mausrad:** Nur y-Achse zoomen

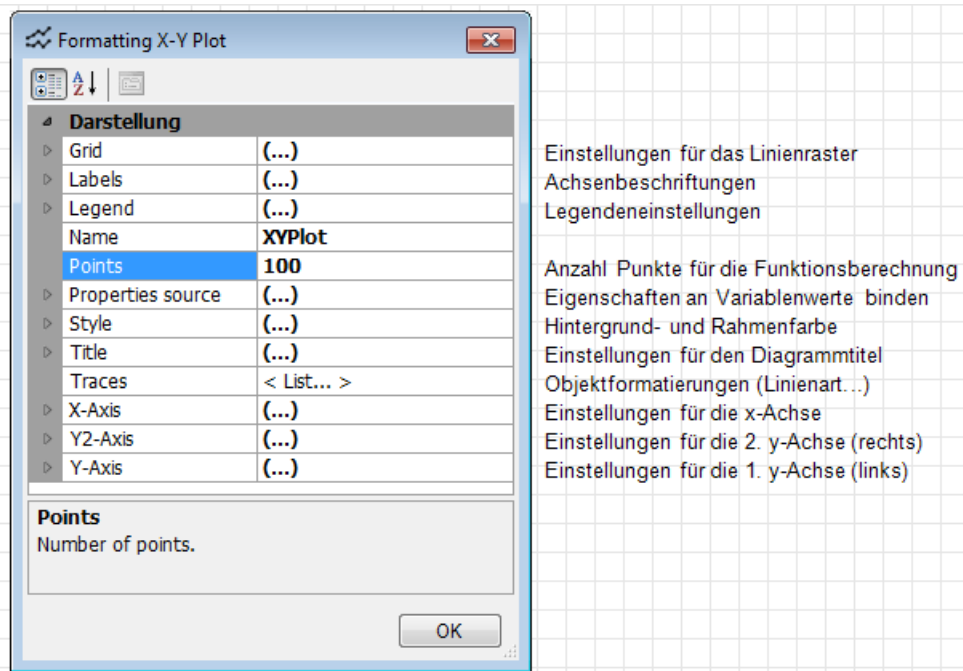
**Ziehen:** Achsen verschieben.

### 8.3.6. Einstellungen

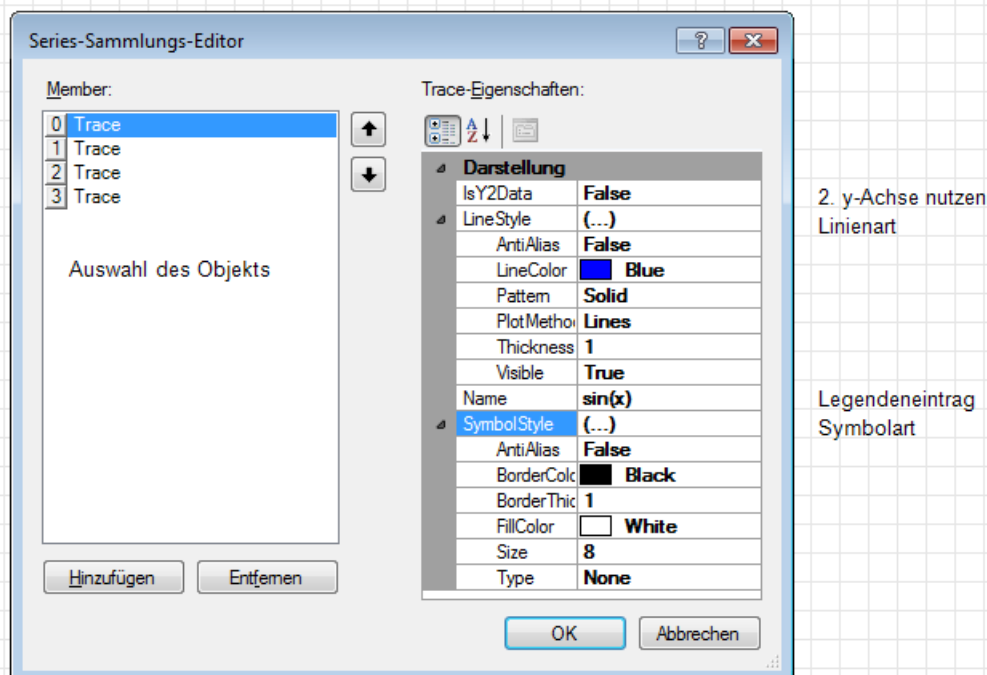
Die Formatierung der Achsen und der Datenlinien und -Punkte erfolgt in einer hierarchischen Menüstruktur, die aufgerufen wird über:

- Doppelklick auf das Diagramm
- **Kontextmenü** > **Format**

Hier können die allgemeinen Diagrammeinstellungen vorgenommen werden.

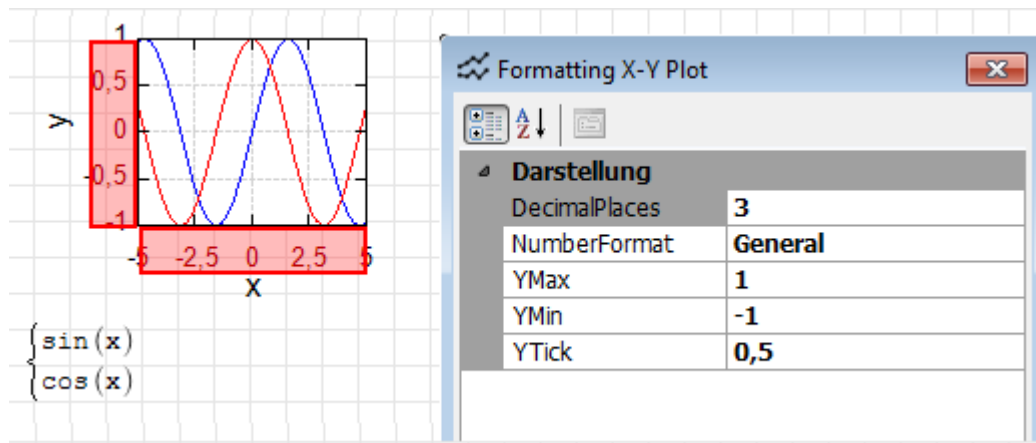


Mit **Traces** >  kommt man auf die Einstellungsseite für die Datenreihen.



Voreingestellt sind Linienfarben, wie sie auch im Standard-Diagrammbereich vorkommen.

**Doppelklick auf die Achsmarken** (*ticks*) öffnet einen direkten Einstellungsdialog für den Wertebereich und den Gitterabstand.

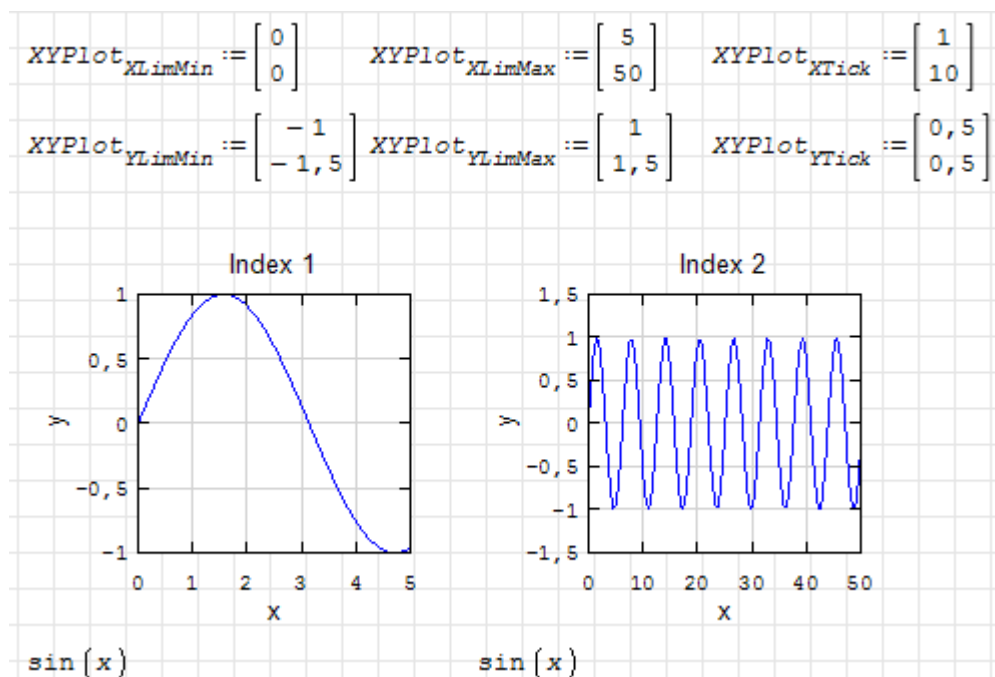


Eine sehr interessante Option ist die Steuerung der Grenzen und der Gitterweite durch **Variablen im Rechenblatt**. Sie wird in den Einstellungen aktiviert durch

- Zweig „Properties source“ aufklappen
- SourcType: „Sheet“
- Index: Vektorindex in den Steuervariablen auswählen.

|                   |              |
|-------------------|--------------|
| Properties source | (...)        |
| Index             | 2            |
| Source Type       | Sheet        |
| Style             | PropertyGrid |
| Title             | Sheet        |

Die folgende Abbildung zeigt die verfügbaren Steuervariablen und gibt ein Beispiel für die Definition und Verwendung zweier unterschiedlicher Einstellungen (links mit Index 1 werden die oberen Werte und rechts mit Index 2 die unteren benutzt):



## 8.4. Der Modeller-Bereich

Der Modeller-Bereich ist ein interaktiver Grafikbereich, der berechnete Grafikobjekte anzeigen und interaktive Manipulationen an Grafikobjekten auswerten kann.

Er wird vom Plug-in *Modeller Region* bereitgestellt (zum Download über die Erweiterungsverwaltung siehe Abschnitt 12.1).

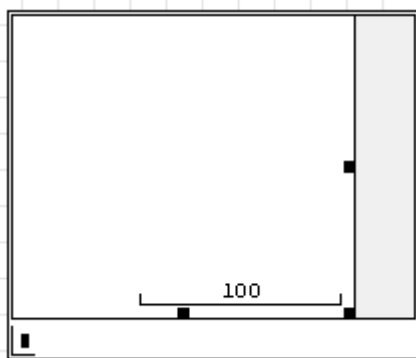
Gesteuert wird der Bereich über drei Variablen oder Funktionen:

- Objektdefinitionen (*templates*), welche die Beschreibung der Grafikobjekte enthalten, siehe Abschnitt 8.4.2,
- Vordefinierte Anordnung (*predefined layout*), siehe Abschnitt 8.4.3,
- Abhängige Anordnung (*dependent layout*)

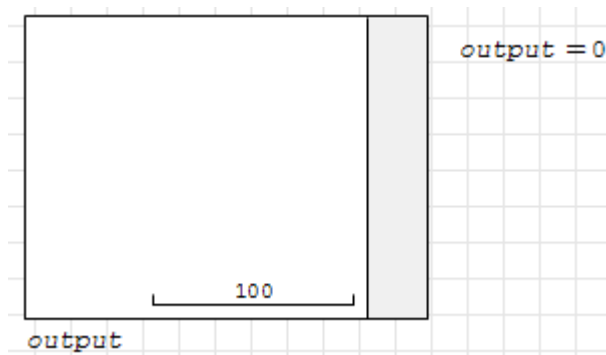
In den Platzhalter des Modeller-Bereichs wird eine Variable geschrieben, deren Wert vom Modeller-Bereich gesetzt wird. Sie enthält die Information über die vorhandenen Grafikobjekte, dies wird in Abschnitt 8.4.5 erläutert.

### 8.4.1. Erste Schritte mit dem Modeller

Einfügen > Modeller

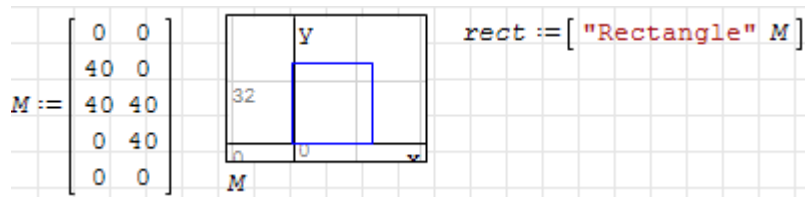


In den Platzhalter ist eine Ausgabevariable einzutragen. Solange noch keine Grafikobjekte vorhanden sind, ist ihr Wert Null.



Grafikobjekte im Modeller müssen in einer Vorlage (*template*) definiert werden. Eine Vorlage ist ein Zeilenvektor mit mehreren Parametern. Alle Vorlagen werden zu einer Liste zusammengefasst.

Die einfachste Vorlage enthält nur den Namen des Objekts und seine Randkurve. Hier wird ein Rechteck definiert:

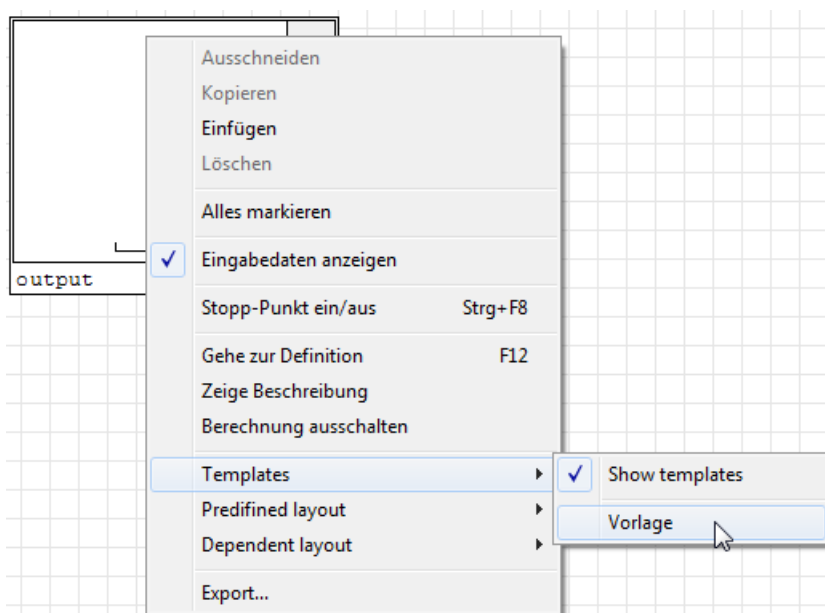


Wie man sieht, hat die Koordinatenmatrix der Randkurve das gleiche Format wie im 2D-Diagrammbereich.

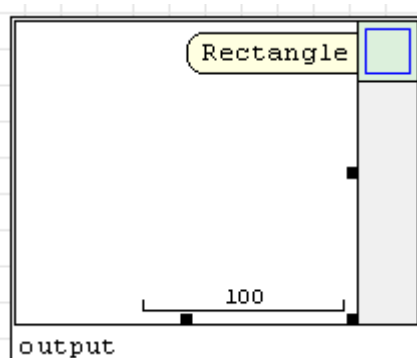
Die Vorlagenliste enthält in unserem Fall zunächst nur eine Vorlage:

```
Vorlage := {rect}
```

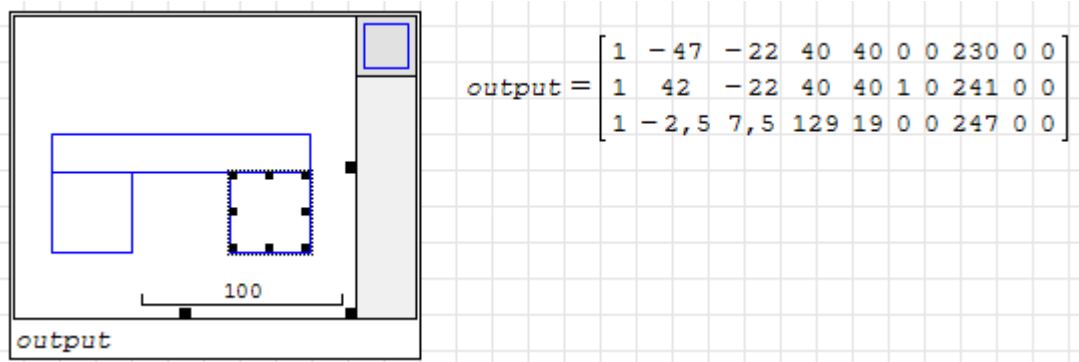
Im Kontextmenü des Modellerbereichs kann man die Liste als Vorlagenliste auswählen (angezeigt werden alle im Rechenblatt definierten Listen)



Nach Zuordnung der Vorlagenliste erscheint das Objekt in der Palette am rechten Bildrand. Ist der Bereich aktiviert, wird der Objektname angezeigt, wenn man mit dem Mauszeiger darübergeht.



Aus der Palette kann man nun das Objekt mit der Maus auf die Zeichenfläche ziehen. An den Marken kann man die Größe des Objekts ändern.



Sobald Objekte auf der Zeichenfläche erscheinen, füllt sich die Ausgabevariable. Jede Zeile steht für ein Objekt, die Spalten stehen für Vorlagennummer (Position in der Vorlagenliste), Mittelpunktskoordinaten, Breite, Höhe, Auswahlstatus und weitere Daten, mehr dazu in Abschnitt 8.4.5.

### 8.4.2. Objektdefinition (Templates)

Alle Grafikobjekte werden durch Vorlagen beschrieben, alle Vorlagen sind in einer Liste enthalten. Die Vorlagenliste wird dem Modeller-Bereich unter dem Punkt „Templates“ im Kontextmenü zugeordnet.

**Polygon-Objekte:** Zeilenvektor:

1. Name (Zeichenkette oder Variable, wird als Hilfetext in der Vorlagenliste angezeigt).
2.  $n \times 2$ -Matrix mit Punktkoordinaten, beschreibt die Randkurve der Geometrie. Optional können Linienfarbe, Strichstärke und Füllfarbe angegeben werden, zusammengefasst zu einem Zeilenvektor.
3. Schalter, ob die Vorlage in der Seitenleiste des Modeller-Bereichs angezeigt werden soll (1: ja, 0: nein).
4. Spaltenvektor mit der Voreinstellung für die Objektabmessungen beim Einfügen aus der Seitenleiste.

In einer Vorlage können auch mehrere Randkurven enthalten sein, dann ist Punkt 2 nicht ein Zeilenvektor, sondern eine Matrix, in der jede Zeile einen Linienzug definiert.

**Textobjekte:** Zeilenvektor aus

1. Name
2. Zeilenvektor

[ Text Farbe Größe ]

3. Schalter zur Anzeige der Vorlage in der Seitenleiste

Textobjekte haben ihren Einfügebepunkt oben in der Mitte.

**Zusammengesetzte Objekte:** Zeilenvektor aus

1. Name
2.  $n \times 3$ -Matrix,
  - a) 1. Spalte: Koordinatenmatrix oder Text
  - b) 2. Spalte: Farbe



- c) 3. Spalte: Strichstärke oder Schriftgröße
- 3. Schalter, ob die Vorlage in der Seitenleiste des Modeller-Bereichs angezeigt werden soll (1: ja, 0: nein).
- 4. Spaltenvektor mit der Voreinstellung für die Objektabmessungen beim Einfügen aus der Seitenleiste.

### 8.4.3. Predefined Layout (vordefinierte Anordnung)

Das ist eine Anordnung von Objekten, die nicht interaktiv verändert werden können und nicht in der Ausgabevariable erscheinen.

Format: Matrix mit den Spalten:

- 1. Vorlagennummer
- 2.  $x$ -Koordinate des Objekts
- 3.  $y$ -Koordinate des Objekts
- 4. Breite
- 5. Höhe

### 8.4.4. Dependent Layout (dynamische Anordnung)

#### Definition mit einer Liste

Zeilenvektor. Die zweite Position gibt die Strecke an, für die der Maßstab im Bild steht. Damit kann man den Zoomfaktor setzen.

[ 1 Maßstab ]

#### Definition mit einer Funktion

Damit können auch die Skalierung und Position sowie der Vorlagenkontext gesteuert werden.

$f(\text{obj}; \text{ratio}; \text{x}; \text{y}; \text{id})$  Rückgabewert ist eine Liste der statischen (nicht veränderbaren) Objekte. Die Argumente der Funktion sind optional und können innerhalb der Funktion geändert werden (Namensparameter).

$\text{obj}$  Dynamische Objektliste

$\text{ratio}$  Länge des Maßstabes bei Standardskalierung

$\text{x}, \text{y}$  Koordinaten der Bildmitte

$\text{id}$  Kontext für neue Einfügungen:

wenn positiv: ID des Objekts, welches als Container und aktueller Kontext benutzt werden soll (alle nachfolgend eingefügten Objekte sind Unterobjekte dieses Containers)

wenn negativ: Nummer der Vorlage, die als Kontext genommen wird.

### 8.4.5. Ausgabe

Die Ausgabevariable im Platzhalter des Modeller-Bereichs geschrieben und anhand der vorhandenen Grafikobjekte belegt. Geliefert wird eine Matrix mit einer Zeile für jedes Grafikobjekt, die Spalten haben die Bedeutung:

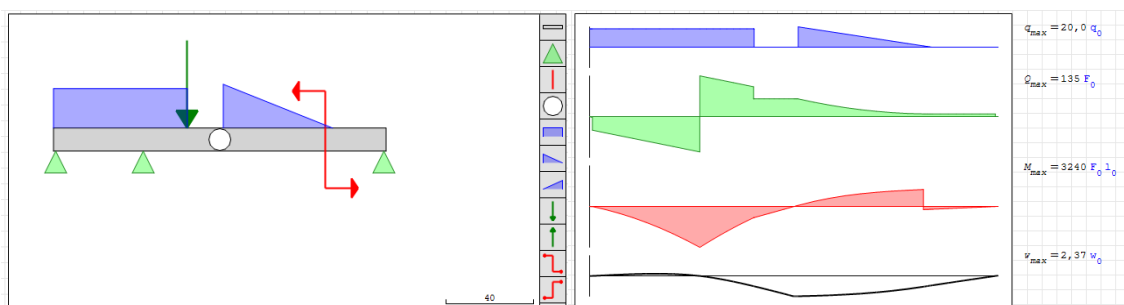
1. Vorlagennummer (Zeilennummer in der Vorlagenliste)
2.  $x$ -Koordinate des Objekts
3.  $y$ -Koordinate des Objekts
4. Breite
5. Höhe
6. Auswahlindikator (1: Objekt ist ausgewählt, 0: Objekt ist nicht ausgewählt)
7. Unbekannt
8. Objektnummer (ID)
9. Unbekannt
10. Unbekannt

### 8.4.6. Beispiel: Modelleditor für Statikberechnungen

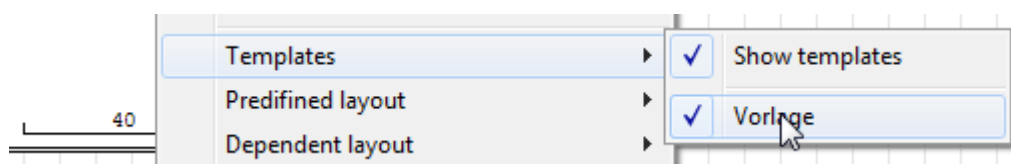
Das Beispiel **BeamFEA** ist in der Online-Galerie über den Erweiterungsmanager verfügbar und kann auch online ausprobiert werden. Der Modeller-Bereich wird dabei gleich zweifach benutzt:

- als Editor für das zu berechnende Balkensystem (Balken, Lager, Gelenke, Kräfte, Momente, Streckenlasten),
- für die Anzeige der Streckenlast, der Querkraft, des Biegemoments und der Biegelinie.

Im Modelleditor kann man Objekte von der Seitenleiste in den Modellbereich ziehen, in Größe und Position ändern und löschen. Der aktuelle Zustand wird mit der SMath-Datei abgespeichert.



Die Objekte in der Seitenleiste werden durch eine Liste definiert, deren Name im Kontextmenü ausgewählt wird:



```

beam
support
sym
circle
dist (1; 1)
Vorlage := dist (1; 0)
           dist (0; 1)
           force (1)
           force (-1)
           moment (-1)
           moment (1)

```

Die Vorlage enthält die eigentlichen Objektdefinitionen. Einige davon werden hier vorgestellt:

**beam** ist ein hellgraues Rechteck mit schwarzem Rand und hat beim Einfügen zunächst die Größe  $100 \times 10$ .

```

beam := "beam" [ [ [ [ 0 -1 ]
                  [ 1 -1 ]
                  [ 1 1 ] "black" 1 "lightgray" ] ] ] 1 [ 100
               [ 0 1 ]
               [ 0 -1 ] ]

```

**force** ist ein grüner Pfeil mit Linienstärke 2, der beim Einfügen zunächst die Größe  $10 \times 40$  hat. Der Parameter  $s$  bestimmt, ob er nach oben oder unten zeigt. Da man die Orientierung im Modeller nicht ändern kann, muss man für den Pfeil aufwärts und abwärts separate Objekte in die Vorlagenliste eintragen.

```

force (s) := "force" [ [ [ [ 0 0 ] "green" 2 "green"
                        [ 0 -1·s ]
                        [ -1 -1·s ]
                        [ 0 0 ] "green" 2 "green" ] ] ] 1 [ 10
                    0,2· [ [ 1 -1·s ]
                        [ -1 -1·s ] ] ]

```

**dist** ist ein hellblaues Trapez, dessen linke und rechte Kantenlänge mit den Parametern  $I$  und  $J$  gesteuert werden, so dass man damit bequem die Bausteine für Streckenlasten in der Vorlagenliste erzeugen kann. Das Objekt hat beim Einfügen zunächst die Größe  $50 \times 20$ . In der Seitenleiste erscheint das Objekt unter dem Namen „distributed load“

```

dist (I; J) := "distributed load" [ [ [ [ 0 0 ]
                                      [ 0 -I ]
                                      [ 1 -J ]
                                      [ 1 0 ] ] ] ] "blue" 0 "#550000ff" 1 [ 50
                                           [ 20 ] ]

```

**circle** ist ein weißer Kreis mit schwarzem Rand und hat beim Einfügen zunächst die Größe  $10 \times 10$ . Die Randkontur wird mit der Schrittweite  $30^\circ$  erzeugt. Die  $10^\circ$  extra sichern ab, dass die Randkontur tatsächlich geschlossen ist. Das Objekt erscheint als „hinge“ in der Seitenleiste.

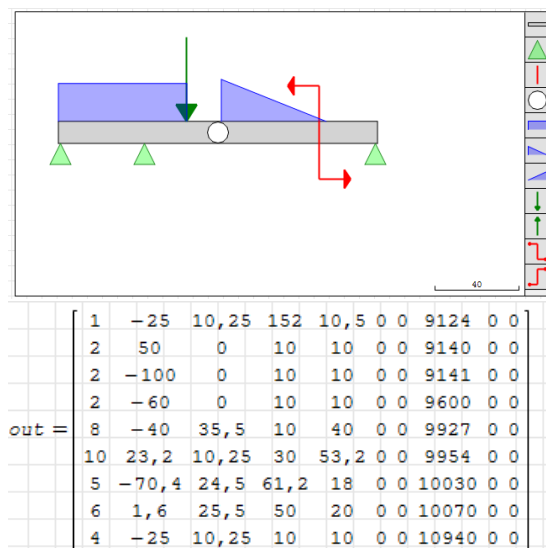
```

arc( $\varphi_1$ ;  $\varphi_2$ ) :=  $\left[ \begin{array}{l} \varphi := [\varphi_1; \varphi_1 + 30^\circ \dots (\varphi_2 + 10^\circ)] \\ \text{augment}(\overrightarrow{\cos(\varphi)}; \overrightarrow{\sin(\varphi)}) \end{array} \right]$ 

circle :=  $\left[ \text{"hinge"} \left[ \text{eval}(\text{arc}(0^\circ; 360^\circ)) \text{"black"} 0 \text{"white"} \right] 1 \begin{bmatrix} 10 \\ 10 \end{bmatrix} \right]$ 

```

Im Platzhalter des Modeller-Bereichs wird die Ausgabevariable eingetragen, die die Information über die tatsächlich angeordneten Objekte enthält. Der Platzhalter ist hier ausgeblendet.



Jede Zeile in der Matrix steht für ein Objekt. Beispielsweise enthält Zeile 1 folgende Informationen:

- 1. Spalte: Wert 1 identifiziert die Objektvorlage anhand der Position in der Liste Vorlagen. In diesem Fall ist es ein Objekt des Typs beam.
- Spalten 2 und 3 geben die Einfügeposition an (Mittelpunkt des Objekts, hier  $x = -25$ ,  $y = 10,25$ ). Das Koordinatensystem ist in der Mitte des Modellerfensters zentriert, die Skalierung kann am Maßstab rechts unten abgelesen werden.
- Spalten 4 und 5 geben die tatsächliche Größe des Objekts an, hier  $152 \times 10,5$ .

Auf Basis dieser Information wird dann das mathematische Modell aufgebaut.

### 8.4.7. Weiterführende Ressourcen

Beispiele im Erweiterungsmanager, die den Modeller-Bereich nutzen:

- **BeamFEA** Das oben beschriebene Beispiel.
- **Colors of the rainbow** Ein kleines Sortierspiel.

Forum:

- [Modeller Region<sup>4</sup>](#) Forum-Rubrik
- [Anleitung zur erweiterten Steuerung<sup>5</sup>](#)

<sup>4</sup>[http://en.smath.info/forum/yaf\\_postst2339\\_Modeller-Region.aspx](http://en.smath.info/forum/yaf_postst2339_Modeller-Region.aspx)


<sup>5</sup><http://smath.info/?file=742103>

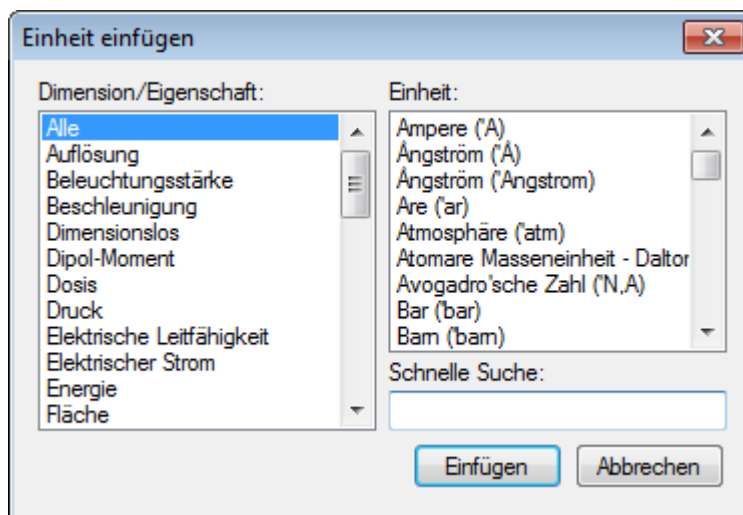
## 9. Maßeinheiten

Eine besondere Stärke von SMath ist die Fähigkeit, mit Maßeinheiten zu rechnen. Jede Zahl kann mit einer Maßeinheit versehen werden und bei Ergebnissen kann die gewünschte Maßeinheit angegeben werden.

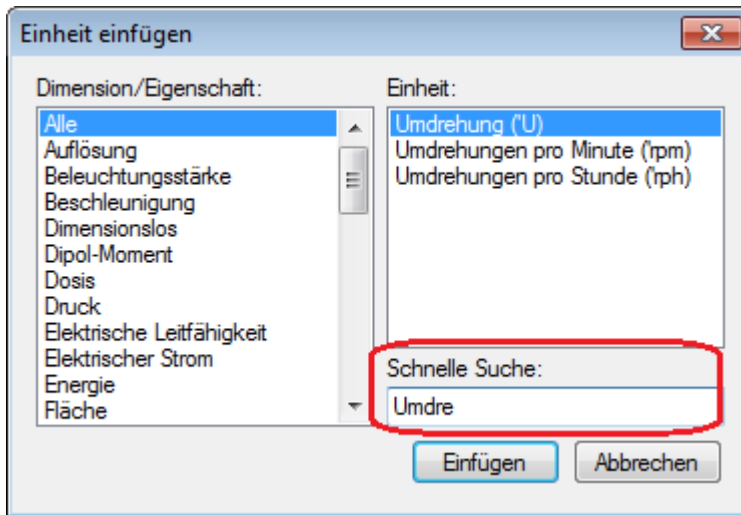
SMath prüft, ob die Maßeinheiten eines Ausdrucks zusammenpassen, nimmt also gewissermaßen eine eigene Einheitenkontrolle vor. Damit ist aber nicht garantiert, dass man nicht aufpassen muss. Besondere Aufmerksamkeit (und damit je einen Abschnitt) verdienen

- die Unterscheidung von **Temperaturen und Temperaturdifferenzen**. Mit der konsequenten Verwendung von Kelvin (K) ist man auf der sicheren Seite.
- die Unterscheidung zwischen **Winkelgeschwindigkeit und Frequenz**. Winkel haben die interne Basiseinheit Radiant (rad), eine Umdrehung entspricht  $2\pi$  rad. Aber auch die Einheiten  $^\circ$  und U (Umdrehung) können verwendet werden. Frequenzen sind Ereignisse pro Zeit, Winkelgeschwindigkeiten sind Winkeländerung pro Zeit.

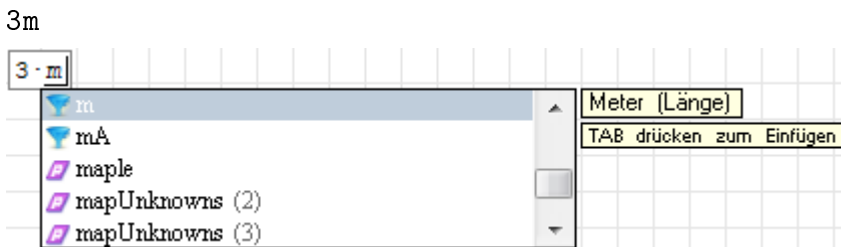
In SMath ist eine ganze Reihe von Einheiten verschiedener Dimensionen vordefiniert. Darüber hinaus sind auch einige physikalische Konstanten dabei. Einen Überblick verschafft man sich im Dialog **Einfügen > Einheit** (auch über **[Strg] W** oder mit  in der Werkzeugleiste erreichbar)



Im linken Fenster wählt man die Dimension, im rechten Fenster erscheinen dann die zugehörigen Maßeinheiten mit ihrem Namen und ihrer internen Darstellung, Im Feld „Schnelle Suche“ kann man die Liste auf die Einheiten eingrenzen, die die angegebene Zeichenkette enthalten



Die interne Darstellung beginnt immer mit einem Hochkomma (z.B. 'm für Meter).  
 Maßeinheiten können auch direkt mit der Tastatur eingegeben werden:

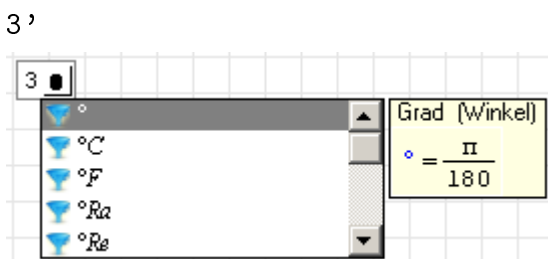


Sobald Sie einen Buchstaben in einem Formelbereich tippen, erscheint die dynamische Hilfe. Sie können weitere Buchstaben eintippen oder mit den Pfeiltasten aus der Liste auswählen. Mit  $\leftrightarrow$  (Tabulator) wird bestätigt.

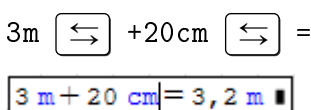


Maßeinheiten werden aufrecht und blau dargestellt.

Tippt man vor dem Einheitenamen ein Hochkomma, schränkt sich die dynamische Hilfe direkt auf Maßeinheiten ein und der Platzhalter nimmt eine runde Form an:



Hier die Summe zweier Längen:



Die interne Darstellung diese Ausdrucks ist  $3 * 'm + 20 * 'cm = 3.2 * 'm @ \#$

## 9.1. Maßeinheit in Ergebnissen

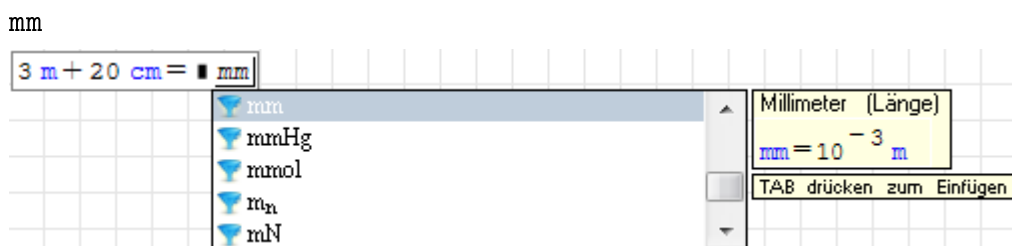
Die standardmäßig angezeigte Maßeinheit in Ergebnissen hängt von der Dimension des Ergebnisses und von der Auswertungsart (numerisch/symbolisch) ab.

- Symbolisch: Die Einheit wird in Basiseinheiten wie m, s, kg ausgedrückt.
- Numerisch: Es wird die für die jeweilige Dimension intern festgelegte Standardeinheit angezeigt.

| symbolisch  | numerisch                                  |
|---|--|
| $\frac{U}{\text{min}} = \frac{\pi}{30 \text{ s}}$ | $\frac{U}{\text{min}} = 0,1047 \text{ Hz}$ |
| $N \text{ m} = \frac{\text{kg m}^2}{\text{s}^2}$  | $N \text{ m} = 1 \text{ J}$                |

Diese Standardeinheiten sind nicht immer sinnvoll, wie beispielsweise bei Winkelgeschwindigkeiten oder Drehmomenten. Daher kann man in jedem Formelbereich separat die Maßeinheit für die Ergebnisanzeige festlegen. Dazu tippt man sie in den Platzhalter hinter dem Ergebnis ein. Weiter im obigen Beispiel:

Um das Ergebnis in mm anzuzeigen, stellen Sie die Eingabemarke auf den Platzhalter hinter dem Ergebnis und tippen



(↵) zur Bestätigung der Auswahl. Die Berechnung erfolgt erst nach Verlassen des Formelbereichs (↵) drücken oder irgendwo anders im Rechenblatt klicken, dann verschwindet der Rahmen um die Formel):

$$3 \text{ m} + 20 \text{ cm} = 3200 \text{ mm}$$

Intern rechnet SMath mit SI-Einheiten. Bei der Eingabe kann jeder Zahl eine Einheit zugeordnet werden. Diese legt die Dimension und den Vorfaktor gegenüber der SI-Einheit fest. Die Einheit mm kennzeichnet die Dimension Länge und hat den Faktor 0.001 in Bezug auf die SI-Einheit m.

Nur Größen gleicher Dimension können addiert, subtrahiert oder verglichen werden. Multiplikation und Division ist uneingeschränkt möglich.

$$1 \text{ m} + 1 \text{ kg} = \text{lastError} = \text{"Einheiten stimmen nicht überein."}$$

Bei der Ergebnisanzeige können nicht nur Maßeinheiten, sondern auch andere Faktoren, wie etwa  $\pi$ , abgespalten werden. Im linken Platzhalter erscheint dann einfach der verbleibende Restfaktor:

$$a \sin(1) = 0,5 \text{ n} \qquad a + \frac{3}{4} \cdot a = 1,75 \text{ a}$$

Als Argumente von Winkelfunktionen und Logarithmen sowie als Exponenten sind Maßeinheiten nicht definiert. Entsprechend weigert sich SMATH völlig korrekt, solche Berechnungen durchzuführen. Ausnahmen sind dimensionslose Maßeinheiten wie Grad ( $^\circ$ ) oder das Bogenmaß (rad).

```
sin(3 kg) =
lastError = "Die Berechnung kann nicht mit Einheiten durchgeführt werden."
```

Achtung: Der Vorsatz  $\mu$  (Mikro) muss als `m` `Strg-G` oder über die Palette der Seitenleiste eingegeben werden, also nicht das Zeichen `AltGr-m` eingeben. Die beiden Befehle scheinen unterschiedliche Zeichencodes zu erzeugen.

## 9.2. Eigene Maßeinheiten

Maßeinheiten können vom Anwender wie normale Variable definiert werden. Damit sie blau und aufrecht erscheinen, stellt man dem Variablennamen ein Hochkomma (') voran, das in der Anzeige dann unsichtbar ist.

```
Bier := 1
Sixpack := 6 Bier
Kasten := 20 Bier

2 Kasten + 3 Sixpack = 58 <- Ergebniseinheit nicht angegeben
2 Kasten + 3 Sixpack = 58 Bier
2 Kasten + 3 Sixpack = 9,6667 Sixpack
2 Kasten + 3 Sixpack = 2,9 Kasten
```

## 9.3. Vektoren und Matrizen mit Maßeinheiten

Elemente einer Matrix können verschiedene Einheiten haben. Matrixoperationen sind ausführbar, solange nicht Elemente unterschiedlicher Dimension addiert oder subtrahiert werden.

Bei der Anzeige ist es allerdings nicht möglich, auf die Maßeinheiten einzelner Elemente, Zeilen oder Spalten Einfluss zu nehmen. Wenn das erforderlich ist, müssen diese z.B. mit `row()`, `col()`, `submatrix()` oder Elementindex herausgegriffen und dann angezeigt werden.

Im Tabellenbereich ist es allerdings möglich, zeilenweise oder spaltenweise Anzeigeeinheiten vorzugeben.



## 9.4. Temperatureinheiten

### 9.4.1. Kurzfassung

Wenn man Temperaturen und Temperaturdifferenzen stets in Kelvin (K) angibt, braucht man hier nicht weiterzulesen.

In der Praxis ist aber die Einheit Grad Celsius ( $^{\circ}\text{C}$ ) verbreitet. Sie wird sowohl für die thermodynamische Temperatur (Zustandsgröße) wie auch für Temperaturunterschiede oder Temperaturänderungen verwendet. Aber in beiden Fällen unterscheidet sich die Umrechnung von  $^{\circ}\text{C}$  zu K und umgekehrt.

**Temperatur** Es gilt  $0^{\circ}\text{C} = 273,15\text{ K}$  und  $0\text{ K} = -273,15^{\circ}\text{C}$

**Temperaturdifferenz**  $1\text{ K} = 1^{\circ}\text{C}$

Bei Temperaturdifferenzen muss man in SMath Studio der Einheit  $^{\circ}\text{C}$  ein Delta voranstellen, also  $\Delta^{\circ}\text{C}$ , egal ob bei der Definition oder bei der Anzeige von Werten.

### 9.4.2. Details

Intern rechnet SMath immer mit K als Temperatureinheit. Wenn der Anwender also eine Größe in  $^{\circ}\text{C}$  angeben will oder ein Ergebnis in  $^{\circ}\text{C}$  angezeigt haben will, muss er SMath sagen, ob es sich um eine Temperatur oder eine Temperaturdifferenz handelt, damit SMath die richtige Umrechnung auswählen kann.

In der Handrechnung weiß man aus dem Zusammenhang, welche Umrechnung zu wählen ist, denn man weiß ja (oder sollte zumindest wissen), ob man gerade eine Temperatur oder eine Temperaturdifferenz definiert oder als Ergebnis erhält.

Nicht so in SMath. Würde man für Temperaturen und Temperaturdifferenzen in SMath  $^{\circ}\text{C}$  verwenden, wüsste SMath nicht, welche der beiden Umrechnungen jeweils anzuwenden wäre.

Als Ausweg erhält die  $^{\circ}\text{C}$  bei der Verwendung für Temperaturdifferenzen den Vorsatz  $\Delta$ , also ist die Einheit dann  $\Delta^{\circ}\text{C}$ .

|                                     |   |
|-------------------------------------|---|
| Temperatur                          | $20^{\circ}\text{C} = 293\text{ K}$                                   |
|                                     | $300\text{ K} = 26,8^{\circ}\text{C}$                                 |
| Temperatur + Differenz = Temperatur | $10^{\circ}\text{C} + 20\text{ K} = 30^{\circ}\text{C}$               |
|                                     | $10^{\circ}\text{C} + 20\Delta^{\circ}\text{C} = 30^{\circ}\text{C}$  |
|                                     | $10^{\circ}\text{C} + 20^{\circ}\text{C} = 303^{\circ}\text{C}$       |
| Temperatur - Temperatur = Differenz | $10^{\circ}\text{C} - 20^{\circ}\text{C} = -10\text{ K}$              |
|                                     | $10^{\circ}\text{C} - 20^{\circ}\text{C} = -10\Delta^{\circ}\text{C}$ |
|                                     | $10^{\circ}\text{C} - 20^{\circ}\text{C} = -283^{\circ}\text{C}$      |

Die Beispiele in rot zeigen, was passiert, wenn man den Unterschied in SMath nicht beachtet!

Auch Temperaturgradienten sind Temperaturdifferenzen, nämlich Temperaturänderungen bezogen auf eine Länge. Hier ein Beispiel: Wärmestrom durch eine Wand:

|                    |   |  |
|--------------------|---|--|
| Innentemperatur    | $T_i := 20 \text{ }^\circ\text{C}$                                |  |
| Außentemperatur    | $T_a := 0 \text{ }^\circ\text{C}$                                 |  |
| Wanddicke          | $w := 0,2 \text{ m}$  |  |
| Wärmeleitfähigkeit | $\lambda := 0,18 \frac{\text{W}}{\text{m K}}$                     |  |
| Temperaturgradient | $\nabla T := \frac{T_i - T_a}{w} = 100 \frac{\text{K}}{\text{m}}$ | $\nabla T = 54,9 \frac{\Delta^\circ\text{F}}{\text{ft}}$ |
| Wärmestrom         | $P_A := \lambda \cdot \nabla T = 18 \frac{\text{W}}{\text{m}^2}$  |  |

Temperaturskalen, die vom absoluten Nullpunkt der thermodynamischen Temperatur ausgehen, heißen absolut. Dazu zählen Kelvin und Rankine. Bei nicht-absoluten Einheiten, also solchen mit Nullpunktverschiebung, wie z.B. Celsius, Fahrenheit oder Reamur ist der Vorsatz  $\Delta$  bei Verwendung für Temperaturdifferenzen erforderlich.

|                 | absolut                         | nicht-absolut                                   | Differenz  |
|-----------------|---------------------------------|---|--|
| Kelvin          | $0 \text{ K} = 0$               |   | $1 \text{ K} = 1 \text{ K}$  |
| Grad Rankine    | $0 \text{ }^\circ\text{Ra} = 0$ |   | $1 \text{ }^\circ\text{Ra} = \frac{5 \text{ K}}{9}$  |
| Grad Celsius    |                                 | $0 \text{ }^\circ\text{C} = 273,15 \text{ K}$   | $\Delta^\circ\text{C} = 1 \text{ K}$   |
| Grad Fahrenheit |                                 | $0 \text{ }^\circ\text{F} = 255,3722 \text{ K}$ | $\Delta^\circ\text{F} = \frac{5 \text{ K}}{9}$<br>$\Delta^\circ\text{F} = 1 \text{ }^\circ\text{Ra}$ |
| Grad Reamur     |                                 | $0 \text{ }^\circ\text{Re} = 273,15 \text{ K}$  | $\Delta^\circ\text{Re} = \frac{5 \text{ K}}{4}$  |

**Vorsicht ist bei negativen Zahlenwerten in nicht-absoluten Einheiten geboten!**

$$-10 \text{ }^\circ\text{C} = -556 \text{ }^\circ\text{C}$$

Was ist hier passiert? Auf der linken Seite steht genau genommen: Null minus zehn Grad Celsius. Wegen Punktrechnung vor Strichrechnung rechnet SMATH zunächst  $10^\circ\text{C} = 283,15 \text{ K}$ , und wendet dann das Minuszeichen an. Die linke Seite wird also völlig korrekt ausgewertet zu  $-(10^\circ\text{C}) = -(283,15 \text{ K})$ . Bei der Anzeige in  $^\circ\text{C}$  müssen vom Kelvin-Zahlenwert wieder 273,15 abgezogen werden, was dann den Wert von -556,3 ergibt.

Die Irritation ergibt sich daraus, dass das negative Vorzeichen (der unäre Minus-Operator) normalerweise höhere Priorität als alle anderen Rechenoperationen hat, in SMATH aber als normale Strichrechnung betrachtet wird.

Das gewohnte Verhalten muss man daher mit Klammern erzwingen:

$$(-10) \text{ }^\circ\text{C} = -10 \text{ }^\circ\text{C}$$

Seit 2019 kann man den Zahlenwert einer Größe auch bei nicht-absoluten Temperaturskalen mittels Division durch die Einheit gewinnen. Vorher musste man explizit die Nullpunktverschiebung rückgängig machen.

$$\begin{array}{l}
 T := 100 \text{ } ^\circ\text{C} \quad T = 212 \text{ } ^\circ\text{F} \\
 \frac{T}{^\circ\text{C}} = 100 \quad \frac{T}{^\circ\text{F}} = 212 \quad \text{Neu 2019} \\
 \frac{T - 0 \text{ } ^\circ\text{C}}{\Delta^\circ\text{C}} = 100 \quad \frac{T - 0 \text{ } ^\circ\text{F}}{\Delta^\circ\text{F}} = 212
 \end{array}$$

## 9.5. Winkel und Umdrehungen

Die Basiseinheit für **Winkel** ist der Radiant, das dimensionslose Verhältnis  $u/r$  zwischen Kreisbogenlänge  $u$  und Radius  $r$ . Dieses ist bei einem Vollkreis mit  $u = 2\pi r$  gleich  $2\pi$ . Dazu gibt es noch weitere vordefinierte Winkeleinheiten.

|   |                          |   |
|---|--------------------------|---|
| $1 \text{ rad} = 57,2958 \text{ } ^\circ$ | Standardeinheit: Radiant |   |
| $1 \text{ } ^\circ = 1 \text{ } ^\circ$   | Grad                     |   |
| $1 \text{ Gon} = 0,9 \text{ } ^\circ$     | Neugrad, Gon             | $\frac{\pi}{2} \text{ Gon} = 1 \text{ rad}$ |
| $1 \text{ U} = 360 \text{ } ^\circ$       | Umdrehung                | $1 \text{ U} = 2 \cdot \pi \text{ rad}$     |
| $1 ' = \frac{1}{60} \text{ } ^\circ$      | Bogenminute              |   |
| $1 '' = \frac{1}{3600} \text{ } ^\circ$   | Bogensekunde             |   |

Winkel pro Zeit ist eine **Winkelgeschwindigkeit oder Kreisfrequenz**. SMATH hält standardmäßig alles, was Dimension 1/Zeit hat, für eine Frequenz. Daher wird auch bei Winkelgeschwindigkeiten im Ergebnis (bei numerischer Auswertung) zunächst die Einheit Hz (Hertz) angezeigt, was natürlich falsch ist.

$$60 \frac{\text{U}}{\text{min}} = 6,2832 \text{ Hz}$$

Bei Ergebnissen, die Winkelgeschwindigkeiten sind, muss auch eine Maßeinheit für Winkelgeschwindigkeiten angegeben werden!

$$60 \frac{\text{U}}{\text{min}} = 2 \cdot \pi \frac{\text{rad}}{\text{s}} \quad \frac{\text{U}}{\text{min}} = 1 \text{ rpm} \quad 60 \frac{\text{U}}{\text{min}} = 1 \frac{\text{U}}{\text{s}}$$

Hier ein Beispiel. Die Leistung  $P$  ist das Produkt aus Drehmoment  $M$  und Winkelgeschwindigkeit  $\omega$ .

$$\begin{array}{l}
 \omega := 3000 \frac{\text{U}}{\text{min}} \quad M := 100 \text{ Nm} \quad P := \omega \cdot M = 31,4 \text{ kW} \\
 \omega = 314 \frac{\text{rad}}{\text{s}}
 \end{array}$$

Wird diese Definition von Umdrehungen verwendet, dann ist es natürlich, die Drehzahl als Winkelgeschwindigkeit oder Kreisfrequenz aufzufassen.

Alternativ kann man die **Drehzahl** als dimensionsloses Ereignis pro Zeit angeben, dann ist die Ausgabe in Hz korrekt:

$$\text{min}^{-1} = 0,0167 \text{ Hz}$$

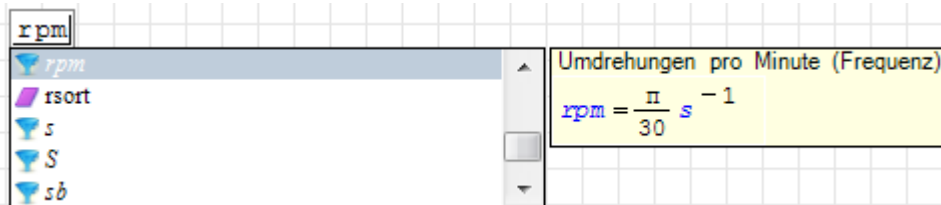
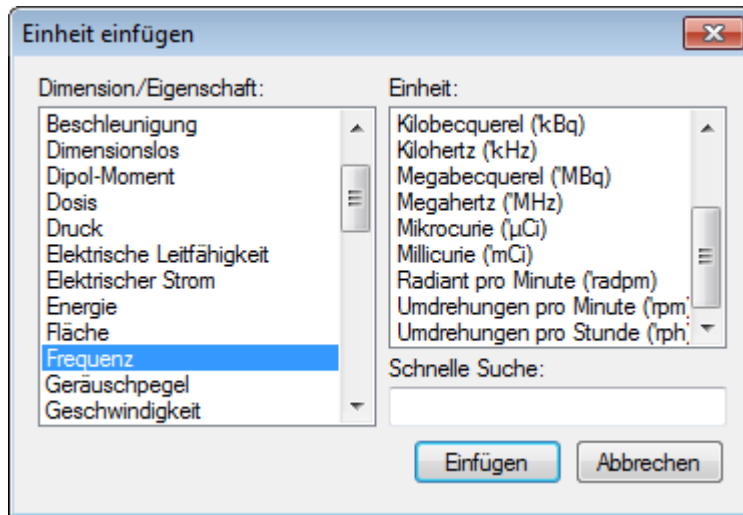
Beim Rechnen damit muss man dann aber auch die richtigen Formeln nehmen (Faktor  $2\pi$  bei der Leistungsberechnung beachten)!

$$n := 3000 \text{ min}^{-1} \quad M := 100 \text{ Nm} \quad P := 2 \cdot \pi \cdot n \cdot M = 31,4 \text{ kW}$$

$$n = 50 \text{ Hz}$$

SMath hintertreibt das Verständnis des Gesagten an zwei Stellen (ein entsprechender Verbesserungsvorschlag findet sich [hier](#)<sup>1</sup>):

1. Einheiten, die sich auf Winkelgeschwindigkeiten beziehen, werden im Auswahldialog und in der dynamischen Hilfe der Dimension „Frequenz“ zugeordnet



Das könnte man beheben durch Einführung einer Dimension Winkelgeschwindigkeit.

2. Bei numerischer Auswertung wird als Standardeinheit Hz angegeben. Das ist bei Winkelgeschwindigkeiten (und auch bei anderen dimensionslosen Größen pro Zeit, z.B. Dehnrates) falsch. Besser wäre hier die Angabe in  $1/s$ , wie es bei symbolischer Auswertung auch gemacht wird.

## 9.6. Winkel in Grad, Minuten, Sekunden

In manchen Anwendungsfällen werden Winkel in Grad, Minuten und Sekunden angegeben, also z.B.  $55^\circ 20' 30''$ . SMATH Studio kennt diese Einheiten:

$$1' = \frac{1}{60}^\circ \quad \text{Strg+K} \quad ' =$$

$$1'' = \frac{1}{3600}^\circ \quad \text{Strg+K} \quad '' =$$

$$\alpha := 55^\circ + 20' + 30'' = 55,342^\circ$$

<sup>1</sup><http://smath.info/bts/Issues/IssueDetail.aspx?id=2395>

Beachte: Die Einheiten ' (Minute) und " (Sekunde) müssen als Sonderzeichen eingegeben werden gebildet, damit es keine Konflikte mit den entsprechenden ASCII-Zeichen gibt, die in SMATH besondere Bedeutung haben. Mit der Eingabe über die dynamische Hilfe oder den Dialog „Einheit einfügen“ ist man auf der sicheren Seite.

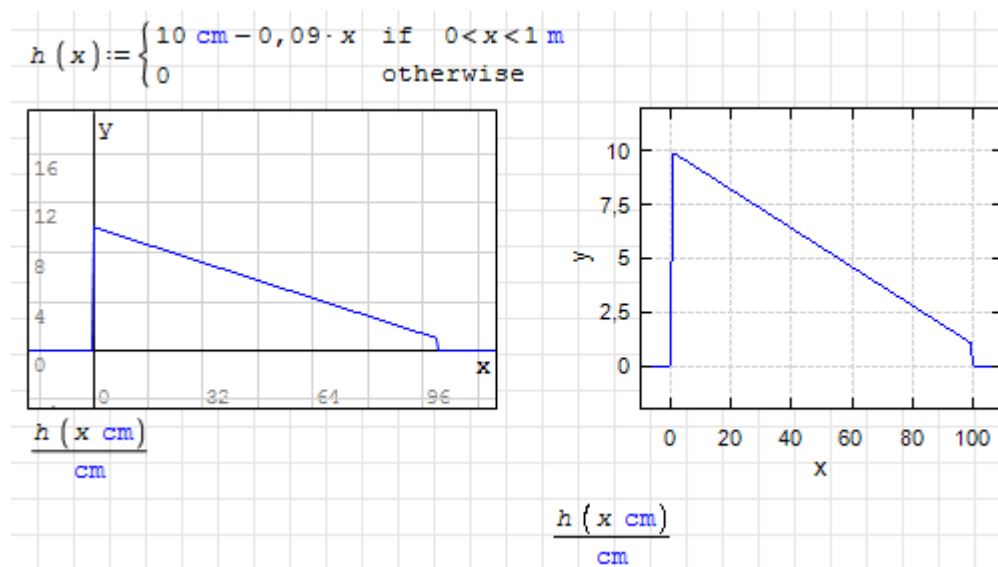
Für die Anzeige im Grad-, Minuten- und Sekundenformat gibt es keine eingebauten Hilfsmittel. Eine anwenderdefinierte Funktion, die dies leistet, kann beispielsweise so aussehen:

$$\begin{aligned}
 \text{DMS}(\varphi) &:= \begin{cases} d := \text{trunc}\left(\frac{\varphi}{\circ}\right) \\ m := \text{trunc}\left(\frac{\varphi}{'} - \frac{d}{\circ}\right) \\ s := \frac{\varphi}{''} - \frac{d}{\circ} - \frac{m}{'} \\ [d \ m \ s] \end{cases} & \alpha_{\text{DMS}} := \text{DMS}(\alpha) = [55 \ 20 \ 30] \\
 & \alpha_{\text{DMS}}^T \cdot \begin{bmatrix} \circ \\ ' \\ '' \end{bmatrix} = 55,342 \circ
 \end{aligned}$$

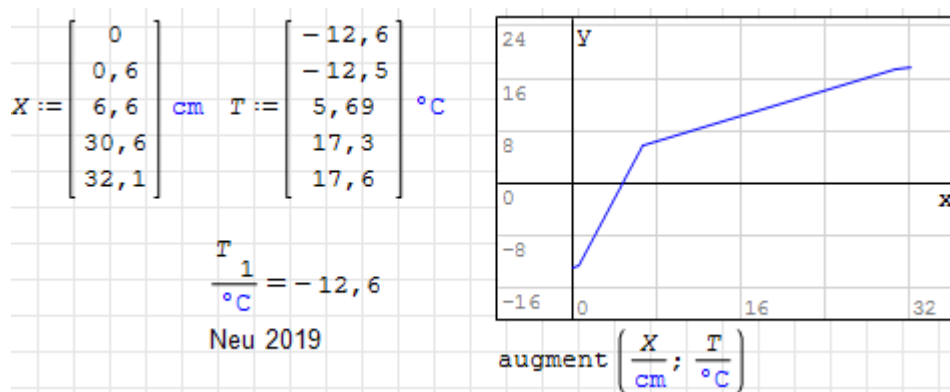
## 9.7. Plotten mit Maßeinheiten

Die unabhängige Variable  $x$  in 2D-Plots (und  $y$  in 3D-Plots) ist dimensionslos. Die Plotfunktion erwartet dimensionslose Werte vom zu plottenden Ausdruck. Soll eine dimensionsbehaftete Funktion einer dimensionsbehafteten Größe geplottet werden, so multipliziert man das Argument mit der gewünschten Einheit der  $x$ -Achse und teilt das Ergebnis durch die gewünschte Einheit der  $y$ -Achse.

Links ist ein Standard-2D-Diagrammbereich und rechts ein X-Y Plot-Bereich zu sehen.



Seit 2019 funktioniert das auch für nicht-absolute Temperatureinheiten. Vorher musste man die Nullpunktverschiebung explizit rückgängig machen:



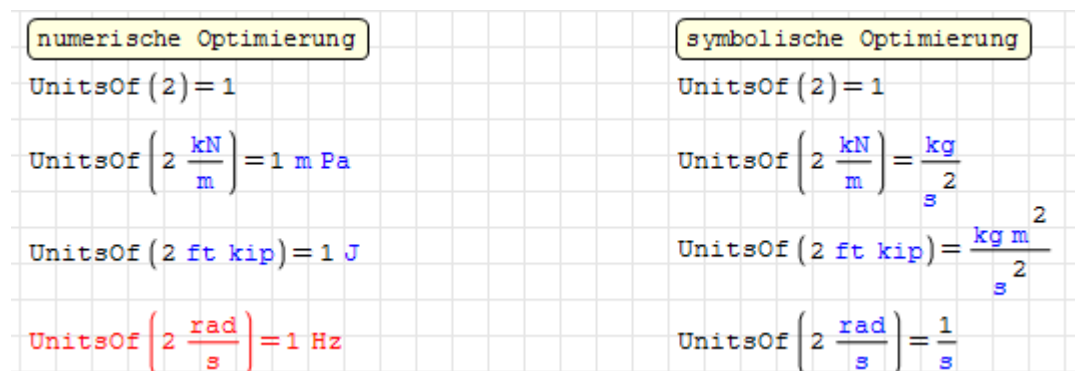
Das funktioniert natürlich auch mit mehreren Funktionen. Diese werden in einer Liste zusammengefasst und separat oder insgesamt auf die richtige Maßeinheit gebracht. Beim X-Y Plotbereich wird es genauso gemacht.

Die Maxima-Diagrammfunktionen Draw2D und Draw3D verhalten sich analog.

## 9.8. Funktionen zum Umgang mit Maßeinheiten

Die Funktion `UnitsOf(Ausdruck)` liefert die zur Dimension des Ausdrucks passende Basiseinheit.

Bei skalaren Ergebnissen (keine Matrizen oder Listen) ist die angezeigte Einheit von der Auswertungsart abhängig. Bei symbolischer Auswertung wird das Ergebnis in Basiseinheiten ausgedrückt, bei numerischer Auswertung wird offenbar versucht, die passende Spezialeinheit zu verwenden. Das birgt Fehlerpotenzial, da z.B. die Einheit Hz bei Winkelgeschwindigkeiten zu Missverständnissen führen kann.



Bei Listen oder Matrizen als Argument wird das Ergebnis immer durch Basiseinheiten ausgedrückt.

|                         |  |
|-------------------------|--|
| numerische Optimierung  | $\text{UnitsOf} \left( \begin{array}{c} 2 \quad 2 \frac{\text{kN}}{\text{m}} \\ 2 \text{ ft lbf} \quad 2 \frac{\text{rad}}{\text{s}} \end{array} \right) = \begin{array}{c} 1 \quad 1 \frac{\text{kg}}{\text{s}^2} \\ 1 \frac{\text{kg m}^2}{\text{s}^2} \quad 1 \frac{1}{\text{s}} \end{array}$ |
| symbolische Optimierung | $\text{UnitsOf} \left( \begin{array}{c} 2 \quad 2 \frac{\text{kN}}{\text{m}} \\ 2 \text{ ft lbf} \quad 2 \frac{\text{rad}}{\text{s}} \end{array} \right) = \begin{array}{c} 1 \frac{\text{kg}}{\text{s}^2} \\ \frac{\text{kg m}^2}{\text{s}^2} \quad \frac{1}{\text{s}} \end{array}$             |

Mit dieser Funktion kann man beispielsweise Größen vorübergehend dimensionslos machen, etwa zur Anwendung von numerischen Verfahren, siehe z.B. Abschnitt 6.10.3 zur numerischen Integration).

## 9.9. Vordefinierte Konstanten

Konstanten und Maßeinheiten werden in der Datei *Installationsverzeichnis/Entries/Units.xml* definiert. Dort kann man auch eigene xml-Dateien einstellen, die Maßeinheitendefinitionen enthalten. In Vorbereitung ist ein Mechanismus, mit dem man eigene Einheitsdefinitionen in der Online-Galerie als Erweiterung veröffentlichen kann. Vorläufig ist es für die Portabilität besser, solche Definitionen im SMATH-Rechenblatt selbst vorzunehmen.

Die verfügbaren Maßeinheiten sind gut im Dialog „Einheit einfügen“ zu finden, daher wird in diesem Handbuch keine entsprechende Liste gepflegt. An dieser Stelle seien nur die vordefinierten physikalischen Konstanten aufgeführt, da nicht alle auch unter dieser Kategorie im Einfüge-Dialog zu finden sind (einige zählen als Maßeinheit, wie z.B. die Lichtgeschwindigkeit).

|   |                                |
|---|--------------------------------|
| $c = 2,9979 \cdot 10^8 \frac{\text{m}}{\text{s}}$                     | Lichtgeschwindigkeit im Vakuum |
| $g_e = 9,8066 \frac{\text{m}}{\text{s}^2}$                            | Erdbeschleunigung              |
| $G_N = 6,6743 \cdot 10^{-11} \frac{\text{m}^2}{\text{kg}^2} \text{N}$ | Gravitationskonstante          |
| $m_p = 1,673 \cdot 10^{-27} \text{kg}$                                | Protonenmasse                  |
| $m_n = 1,675 \cdot 10^{-27} \text{kg}$                                | Neutronenmasse                 |
| $u = 1,66 \cdot 10^{-27} \text{kg}$                                   | Atomare Masseneinheit          |
| $m_e = 9,109 \cdot 10^{-31} \text{kg}$                                | Elektronenmasse                |
| $e = 1,602 \cdot 10^{-19} \text{C}$                                   | Elementarladung                |

|  |  |
|--|--|
| $h = 6,6261 \cdot 10^{-34} \frac{\text{J}}{\text{s}}$      | Plancksches Wirkungsquantum                          |
| $k = 1,3807 \cdot 10^{-23} \frac{\text{J}}{\text{K}}$      | Boltzmann-Konstante                                  |
| $\mu_0 = 1,2566 \cdot 10^{-6} \frac{\text{m T}}{\text{A}}$ | Magnetische Feldkonstante, Permeabilität des Vakuums |
| $R_m = 8,3145 \frac{\text{J}}{\text{mol K}}$               | Gaskonstante   |
| $N_A = 6,0221 \cdot 10^{23} \frac{1}{\text{mol}}$          | Avogadro-Zahl (Teilchenzahl pro mol)                 |



# 10. Programmierung

Man kann in SMath Studio auch programmieren. Das wird unterstützt durch

- benutzerdefinierte Funktionen,
- Funktionen zur Ablaufsteuerung (Schleifen, Verzweigungen, Anweisungsblöcke),
- Funktionen für Zeichenketten,
- Schrittweise Ausführbarkeit, Haltepunkte und Inspektion der definierten Variablen im Kontext,
- Zeitmessung mit der Funktion `time()`.

Für die Programmierung relevante Eigenschaften von SMath Studio:

- Unterschied zwischen symbolischer und numerischer Auswertung
- Der Datentyp von Ausdrücken ergibt sich durch ihren Inhalt.
- Der Namensraum (die Gesamtheit der sichtbaren definierten Variablen) heißt Kontext.

## 10.1. Benutzerdefinierte Funktionen

Diese Darstellung basiert auf einer [Anleitung von Davide Carpi](#)<sup>1</sup>.

Benutzerdefinierte Funktionen werden eingesetzt, um Ausdrücke oder ganze Berechnungsabläufe mehrfach zu verwenden und zu gliedern.

**Symbolische Variable** (*dependent variable*) Name, der auf einen symbolischen Ausdruck verweist. Beim Aufruf wird der Ausdruck unter Einsetzung bekannter Variablenwerte aus dem Rechenblatt vereinfacht. Deren Verwendung wurde bereits in Abschnitt 6.6 behandelt. Im Abschnitt 10.1.2 werden sie als Spezialform der benutzerdefinierten Funktion betrachtet.

**Funktion** (*inline function*) Name mit Argumenten, der auf einen symbolischen Ausdruck verweist. Dabei können die Werte der Argumente beim Aufruf (Verwendung) der Funktion angegeben werden.

**Prozedur** (*procedure*) Name mit Argumenten, der auf einen Anweisungsblock verweist. Die Argumente können auch für die Rückgabe von Werten genutzt werden.

---

<sup>1</sup>[http://en.smth.info/forum/yaf\\_postst7286\\_User-defined-Functions-in-SMath-Studio.aspx](http://en.smth.info/forum/yaf_postst7286_User-defined-Functions-in-SMath-Studio.aspx)

Symbolische Variable

$$m := V \cdot \rho \qquad V := 1 \text{ l} \qquad \rho := 2 \frac{\text{g}}{\text{cm}^3} \qquad m = 2 \text{ kg}$$

Einfache Funktion

$$y(x) := a \cdot x^2 + b \cdot x + c \qquad y(2) = c + 2 \cdot (b + 2 \cdot a)$$

Prozedur

$$w(f(1); n) := \begin{cases} \text{Clear}(x) \\ \text{for } j \in [1..n] \\ \quad x_{j-1} := f(j) \\ x \end{cases} \qquad w(\sin(x); 4) = \begin{bmatrix} 0,841 \\ 0,909 \\ 0,141 \\ -0,757 \end{bmatrix}$$

### 10.1.1. Einfache Funktionen

Die Definition einer einfachen Funktion besteht aus dem Funktionsnamen, Argumentnamen in Klammern, dem Zuweisungsoperator := (eingetragen als :) und einem Ausdruck auf der rechten Seite, in dem die Argumente vorkommen.

$$f(x) := x^3 + \frac{1}{x}$$

Für die Namen der Funktionen und deren Argumente gelten die gleichen Regeln wie für alle Namen in SMATH. Es sind alle Zeichen erlaubt, solange das erste Zeichen keine Ziffer ist. Gegebenenfalls muss man bei der Eingabe mit **Strg+K** in den Sonderzeichenmodus umschalten.

Die dynamische Hilfe kennt alle Definitionen und zeigt beim Eintippen von Namen die möglichen Ergänzungen an.

Hier wird nur **f** angezeigt, es gibt bisher nur die eine Funktion mit diesem Namen:


Beim Aufruf werden für die Argumente Ausdrücke übergeben, mit denen die Argumentnamen im Ausdruck der rechten Seite ersetzt werden. Nach dieser Ersetzung wird der Ausdruck der rechten Seite vereinfacht und der berechnete Wert wird als Ergebnis zurückgegeben.

$$f(2) = 8,5$$

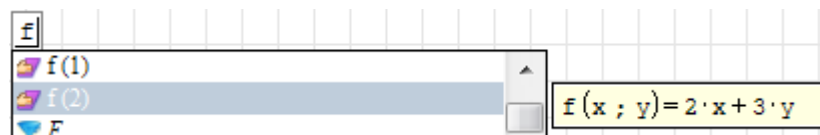
Groß- und Kleinschreibung zählt, **F(x)** ist etwas anderes als **f(x)**:

$$F(x) := x + 3 \qquad f(2) = 8,5 \qquad F(2) = 5$$

Funktionen mit gleichem Namen, aber unterschiedlicher Zahl von Argumenten sind verschiedene Objekte. Mehrere Argumente werden mit dem Argumenttrennzeichen getrennt. Dies ist das Semikolon (denn das ebenfalls geeignete Komma wird in Deutschland üblicherweise als Dezimaltrennzeichen verwendet).

```
f(x; y) := 2 · x + 3 · y  f(x; y  : 2x+3y
f(1; 2) = 8
f(2) = 8,5
```


Die dynamische Hilfe zeigt jetzt zwei Objekte,  $f(1)$  und  $f(2)$ , entsprechend der Zahl der Argumente, die diese Funktionen erwarten.



Für die Argumentnamen in Funktionsdefinitionen gelten die gleichen Regeln wie für sonstige Namen, insbesondere dürfen sie nicht mit einer Ziffer anfangen. Ein paar Beispiele, die funktionieren:

```
f(x3) := x3 + 4
f(2) = 6,000

f(_3) := _3 + 5
f(2) = 7,000

f(#3) := #3 + 6  f(3  #  : 3  #  +6
f(2) = 8,000
```

Ein Negativbeispiel: Zahlen sind keine gültigen Argumentnamen in Definitionen.

```
f(2) := a + b + 3 · i
f(1,5) := a + b + 3 · i lastError = "Syntaxfehler."
```

Wenn Sie den Funktionswert an einer bestimmten Stelle separat vorgeben wollen, dann verwenden Sie Fallunterscheidungen bei der Definition.

Der Buchstabe  $i$  sollte als Argumentname vermieden werden, da dessen Ersetzung durch einen Wert beim Funktionsaufruf Nebenwirkungen auf Rechnungen mit komplexen Zahlen haben kann. SMATH gestattet das aber formal.

```
f(i) := a + b + 3 · i
f(3) = 9 + a + b
```

Wird ein Argument in der Funktionsdefinition nicht verwendet, ist es eigentlich sinnlos. Wenn man aber aus irgendwelchen Gründen trotzdem eine Funktion definieren will, deren Argument ignoriert wird, dann kennzeichnet man das am besten durch einen leeren Platzhalter. Beim Aufruf muss aber in jedem Fall ein Argument eingesetzt werden

```
f() := x + 9
f(2) = 9 + x
f(" ") = 9 + x
f() = ) lastError = "Nicht alle Platzhalter sind gefüllt."
```

**Funktionen als Argumente:** Argumentnamen können auch Funktionen sein, dabei muss in der Definition die Zahl der Argumente angegeben werden (entweder durch eine Liste von Argumenten, deren Namen belanglos sind, oder durch eine Zahl).

Beispiel: Die Funktion  $f$  hat zwei Argumente: eine Funktion mit einem Argument sowie einen Ausdruck. Die Variable  $x$  muss auf dem Rechenblatt undefiniert sein, damit das funktioniert.

The screenshot shows a grid-based calculator interface with the following text:

$$f(g(x); y) := 2 \cdot g(0) + 3 \cdot g(y)$$

$$f\left(\sin(x); \frac{\pi}{2}\right) = 3 \qquad 2 \cdot \sin(0) + 3 \cdot \sin\left(\frac{\pi}{2}\right) = 3$$

$$f(\exp(x); 1) = 10,2 \qquad 2 \cdot \exp(0) + 3 \cdot \exp(1) = 10,2$$

$$\text{Clear}(F(1)) = 1$$

$$f(F(a); a) = 2 \cdot F(0) + 3 \cdot F(a)$$

**Globale und lokale Variablen:** Die rechte Seite von Funktionsdefinitionen wird zum Zeitpunkt der Definition wie jeder andere Ausdruck auf dem Rechenblatt berechnet und gespeichert. Alle bekannten Definitionen werden eingesetzt und der Ausdruck möglichst symbolisch vereinfacht.

Ausnahmen: Formalargumente (Namen, die in der Argumentliste der Definition stehen) werden nicht ersetzt, egal ob es im Rechenblatt dafür bereits Definitionen gibt.

Im gespeicherten Ausdruck können also zwei Sorten Variablen vorkommen:

- Formalargumente, die beim Aufruf durch die übergebenen Argumentwerte ersetzt werden. Das sind im Sinne der Programmierung **lokale Variablen**, ihr Namensraum ist das Innere der Funktion.
- Alle anderen Variablen, die beim Aufruf durch ihren Wert im rufenden Kontext (Rechenblatt oder aufrufende Funktion) ersetzt werden. Das sind also **globale Variablen**.

Der gleiche Ausdruck in einer Funktionsdefinition kann zu verschiedenen gespeicherten Ausdrücken führen, abhängig davon, ob die darin vorkommenden Namen im Rechenblatt bereits definiert sind oder nicht.

Die Prüfung der gespeicherten Definition mit der dynamischen Hilfe ist also Pflicht!

Im folgenden Beispiel ist  $x$  eine lokale Variable,  $a$  hat im Kontext einen Wert,  $b$  ist nicht definiert.

The screenshot shows a grid-based calculator interface with the following text:

$$a := 5$$

$$f(x) := a + b + 3 \cdot x$$

A yellow box highlights the definition of  $f(x)$  with the text  $f(x) = 5 + b + 3 \cdot x$ .

$$b := 2 \qquad f(2) = 13$$

$$b := 3 \qquad f(2) = 14$$

$$a := 10 \qquad f(2) = 14$$

Das gilt so übrigens auch für Funktionen. Im folgenden Beispiel ist  $F(x)$  zum Zeitpunkt der Definition von  $f(x)$  schon bekannt,  $G(x)$  jedoch nicht. Entsprechend wird nur  $G(x)$  als globaler Name in der Definition gespeichert. Das ist auch sehr gut in der dynamischen Hilfe zu erkennen.

```

F(x) = 3 + x
f(x) := F(x) + G(x)
f(x) = x + 3 + G(x)
F(x) := 5
G(x) := 2 * x
G(x) := 3 * x
f(1) = 4 + G(1)
f(1) = 4 + G(1)
f(1) = 6
f(1) = 7

```

**Funktionen für Matrix-Argumente.** Manche Ausdrücke sind nur für skalare Argumente definiert, z.B. die Betragsfunktion `abs()`:

```

M := [ 1  2  3
      -1 -2 -3 ]
f(x) := |x| - 1
f(M) = ■      lastError = "Argument muss ein Skalar sein."
f(2) = 1

```

Man kann nun die Funktion `f(x)` elementweise auf die Matrix anwenden, dazu dient die Funktion `vectorize()`:

```

vec → f(M) = [ 0 1 2
              0 1 2 ]
vec → f(2) = 1

```

Wie man sieht, geht das auch mit skalaren Argumenten, auch wenn es da nicht nötig wäre. Will man eine Funktion definieren, die gleichermaßen mit Skalaren und Matrizen (elementweise angewendet) zurechtkommt, kann man `vectorize()` ins Innere der Funktion verlegen, also zu einem Teil der Definition machen:

```

f(x) := |vec(x)| - 1
f([ [ 1  2  3
     -1 -2 -3 ] ]) = [ 0 1 2
                     0 1 2 ]
f(2) = 1

```

### 10.1.2. Symbolische Variable

Die Verwendung symbolischer Variablen wurde im Abschnitt 6.6 bereits behandelt. Man kann sie auch als Funktionen betrachten, bei denen die Wertübergabe ausschließlich über globale Variable erfolgt, die also keine formalen Argumente besitzen.

Der Funktionsaufruf besteht einfach in der Abfrage des Wertes, wobei die gerade gültigen Werte aus dem Kontext in den gespeicherten symbolischen Ausdruck eingesetzt werden.

```

y := 2 * x + 3
x := 2   y = 7
x := 3   y = 9

```

Bei Integration oder Ableitung werden eventuelle Werte der angegebenen unabhängigen Variablen ignoriert.

$$\frac{d}{d x} y = 2 \quad \int_0^2 y d x = 10$$

Genau wie bei einfachen Funktionen wird der Ausdruck bei der Definition unter Verwendung aller bekannten Werte für die darin vorkommenden Variablen vereinfacht. Was bei der Definition tatsächlich abgespeichert wird, ist also vom Kontext abhängig.

Das kann verwirrend sein, aber die dynamische Hilfe lässt keine Zweifel offen. Im folgenden Beispiel hat  $b$  bereits einen Wert, welcher bei der Definition von  $x$  eingesetzt wird.

```

b := 5
x := 2 · z + b + c = 5 + c + 2 · z
|x = 2 · z + 5 + c
c := 2    x = 7 + 2 · z
c := 4    x = 9 + 2 · z

```

In der Definition können natürlich auch Funktionen enthalten sein:

```

x := b + d + G(z)
|x = 5 + d + 3 · z
z := 3  d := 3    x = 17
        d := 2    x = 16
z := 2                    x = 13
                        b := 3  x = 13

```

### 10.1.3. Prozeduren

Prozeduren sind spezielle Funktionen,

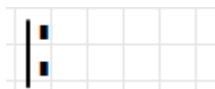
- deren Argumente sowohl zur Übergabe von Eingangsdaten als auch zur Rückgabe von Ergebnissen dienen können,
- deren rechte Seite zur Definitionszeit nicht unter Verwendung des Kontexts vereinfacht wird und
- deren Inneres einen separaten Namensraum bildet, in dem der globale Kontext sichtbar ist und in dem lokale Definitionen möglich sind.

Eine Prozedur kann beim Aufruf auf zwei Wegen Werte zurückgeben:

- den Funktionswert selbst oder
- durch Veränderung des Werts von Variablen, die beim Aufruf als Argumente verwendet werden.

Sie werden wie einfache Funktionen definiert, außer dass die rechte Seite durch einen Anweisungsblock gekapselt ist.

Anweisungsblöcke werden mit der `line()`-Funktion gebildet, erreichbar mit „line“ auf der Palette „Programmierung“ der Seitenleiste.



In jedem Platzhalter kann eine SMath-Anweisung stehen. Wie bei Listen oder Vektoren kann man durch Eintippen des Argumenttrennzeichens (Semikolon) Zeilen einfügen und durch  $\leftarrow$  Zeilen löschen.

Im folgenden Beispiel wird die Funktion  $f(x)$  definiert. Rückgabewert ist der Ausdruck in der letzten Zeile des Anweisungsblocks. Die Variable  $a$  erhält innerhalb des Blocks beim Aufruf der Funktion den Wert 13, das ist aber außerhalb des Blocks nicht zu sehen (der Wert ist vorher und hinterher 10).

```
a = 10
f(x) := | a := 3 · x + 4   f(3) = 13   a = 10
        | a
```

Das Besondere an Prozeduren ist, dass ihre Argumente auch für die Ausgabe verwendet werden können.

**Beispiel 1:** Im folgenden Beispiel wird die im Rechenblatt bereits definierte Variable  $a$  durch die Verwendung als Argument in  $f(x)$  im Inneren der Funktion als  $x$  sichtbar und kann dort verwendet und verändert werden. Das nennt man *call by name*, Namensübergabe). Die dynamische Hilfe macht die Zuweisung an  $a$  beim Aufruf der Funktion explizit sichtbar.

```
f(x) := | x := x + 1
        | out := x2
        | out
a := 2
f(a) = 9      Rückgabewert der Funktion
f(a) = 16    Wert von a wurde in der Funktion
a = 4        verändert.
| a = 4
```

Naturgemäß funktioniert die Namensübergabe nur mit Namen, nicht mit Ausdrücken oder z.B. Zahlen. An diese kann kein Wert zurückgegeben werden. Die Namen müssen dabei genau wie bei jeder anderen Zuweisung nicht vorher definiert sein.

**Beispiel 2:** Hier ein etwas komplexeres Beispiel für die Benutzung von Ausgabeargumenten:

```
a := 1 while a < 10   b = 100   a = 10
      b := f(a)
| a = 10
| b = 100
```

1. Iteration:  $a = 1$  und  $b = f(1) = (1 + 1)^2 = 4$ .  $a$  wird beim Funktionsaufruf um 1 erhöht.
2. Iteration:  $a = 2$  und  $b = f(2) = (2 + 1)^2 = 9$  und wieder wird  $a$  um 1 erhöht.

Das geht so weiter, bis bei  $a = 10$  die Eintrittsbedingung in die Schleife nicht mehr erfüllt ist.

**Beispiel 3:** Hier wird das Ergebnis von  $f(a)$  wieder auf  $a$  zugewiesen.

```
a := 1 while a < 10 a = 25
      a := f(a)
```

1. Iteration:  $a = 1$ , Auswertung der Funktion liefert  $f(a) = f(1) = (1 + 1)^2 = 4$  und als Nebenwirkung wird  $a = 2$  gesetzt. Allerdings geht dieser Wert gleich wieder verloren, weil  $a$  mit dem Funktionswert überschrieben wird, also ist  $a = 4$ .

2. Iteration:  $a = 4$ ,  $f(a)$  liefert  $f(4) = (4 + 1)^2 = 25$ . Als Nebenwirkung wird  $a$  hier kurzzeitig auf 5 gesetzt und dann aber gleich mit dem Funktionswert 25 überschrieben.

Damit ist die Schleife beendet, weil  $25 > 10$  ist. Die Änderung des Arguments innerhalb der Funktion bleibt hier wirkungslos.

**Beispiel 4:** Hier wird beim Funktionsaufruf der vorher nicht definierte Name `message` als Argument verwendet. Die Funktion weist diesem Argument einen Wert zu und definiert damit den Namen auf dem Rechenblatt.

```
f(x; msg) := | a := x^2 - 10
              | if a < 0
              |   msg := "result not allowable"
              | else
              |   msg := "ok"
              | a
```

```
f(5; message) = 15
message = "ok" | message = "ok"
```

Will man in Prozeduren sicherstellen, dass die Argumente beim Aufruf nicht verändert werden, darf man ihnen im Inneren der Funktion halt keine Werte zuweisen. Im Zweifelsfall kopiert man das Argument auf eine lokale innere Variable.

**Beispiel 5:** Hier bleibt der Funktionsaufruf  $f(a)$  für  $a$  ohne Nebenwirkungen.

```
f(x) := | xt := x
        | xt := xt + sqrt(xt)/2 + xt^2
        | xt
        | a := 2
        | f(a) = (1 + 6*sqrt(2))/sqrt(2)
        | a = 2
```

Die Definition von Prozeduren wird immer so abgespeichert, wie sie geschrieben ist. Anders als bei einfachen Funktionen und symbolischen Variablen werden zur Definitionszeit bekannte Variablenwerte (der Kontext) im Rechenblatt nicht berücksichtigt.

Beim Funktionsaufruf werden alle Argumentnamen durch die übergebenen Werte oder Ausdrücke ersetzt. Dann werden die Anweisungen nacheinander abgearbeitet. Immer wenn ein im Inneren undefinierter Name auftritt, wird geprüft, ob er außerhalb der Funktion definiert ist und wenn ja, verwendet. Anderenfalls bleibt er unbekannt.

**Beispiel 6:** Hier ist  $c_1$  im Rechenblatt bekannt, wird aber in der Definition von  $f(x)$  nicht ersetzt.

```
c1 := 2 f(x) := | out := x + 3*c1
                | out
                | f(x) = | out := x + 3*c1
                | out
```



Beim Aufruf  $f(2)$  wird zunächst in der Definition  $x$  durch 2 ersetzt. Dann wird `out` berechnet und dabei der extern vorhandene Wert für  $c_1$  verwendet. Das Ergebnis ist  $2+3\cdot 2 = 8$ .

Bei anderen Werten für  $c_1$  kommt beim Aufruf  $f(2)$  entsprechend ein anderes Ergebnis heraus.

```

out := 0
c1 := 2  f(2) = 8  out = 0
a := 2  f(a) = 8  out = 0
c1 := 6  f(a) = 20  out = 0
c1 := 7  f(a) = 23  out = 0

```

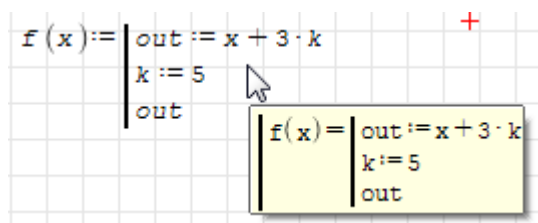
Im Inneren wird die Variable `out` definiert, diese ist aber außen nicht zu sehen. Das gilt auch bei Variablen, die außerhalb bereits definiert sind. Deren Werte sind innerhalb der Funktion solange sichtbar, wie sie nicht innerhalb überschrieben werden. Aber auch da wird die Überschreibung nicht nach außen sichtbar.

**Beispiel 7:** Neben dem Argument  $x$  hat die folgende Funktionsdefinition in der ersten Anweisung noch die globale Variable  $k$  als Eingangskanal. In der zweiten Anweisung wird  $k$  aber überschrieben.

```

f(x) := | out := x + 3 * k
      | k := 5
      | out

```



Es ist vielleicht nicht auf ersten Blick zu erkennen, aber auf diese Art kann man mit globalen Variablen auf Funktionen Einfluss nehmen und innerhalb der Funktionen für diese Variablen Standardwerte angeben, falls die globale Variable nicht definiert ist.

Hier ist  $k$  nicht definiert und `out` hat zunächst den Wert  $2 + 3k$ . Dann wird  $k = 5$  gesetzt und `out` zu  $2 + 3 \cdot 5 = 17$  ausgewertet:

```

[k] = ■      lastError = "k - nicht definiert."
f(2) = 17
[k] = ■      lastError = "k - nicht definiert."

```

Im folgenden Aufruf ist global  $k = 4$  und `out` bekommt in der ersten Anweisung der Funktion den Wert  $2 + 3 \cdot 4 = 14$ . Daran ändert auch die folgende Überschreibung  $k = 5$  nichts mehr. Zudem ist die Überschreibung nach außen nebenwirkungsfrei.

```

k := 4  f(2) = 14  k = 4

```

Überlebensregel: lokale Variablen immer vor der ersten Verwendung definieren, sonst könnten sie einen globalen Wert haben!

Beachten Sie, dass bei Feldern (Matrizen, Listen) elementweise Definitionen nicht als Initialisierung ausreichen! Im folgenden Beispiel wird in  $f(x)$  die Matrix  $M$  nicht richtig initialisiert. Entsprechend haben globale Werte für  $M$  massiven Einfluss auf das Ergebnis. Bei  $g(x)$  wird hingegen ein sauberer Anfangszustand hergestellt und das Ergebnis hängt nur vom übergebenen Argument ab.

| Zerbrechliche Definition                    |   | Robuste Definition durch Initialisierung |   |
|---|---|--|---|
| $f(x) :=$                                   | $\begin{cases} M_{11} := x \\ M_{12} := 2 \cdot x \\ M \end{cases}$ | $g(x) :=$                                | $\begin{cases} M := 0 \\ M_{11} := x \\ M_{12} := 2 \cdot x \\ M \end{cases}$ |
| $M := [1\ 1]$                               | $f(2) = [2\ 4]$   | $g(2) = [2\ 4]$                          |   |
| $M := [1\ 1\ 1]$                            | $f(2) = [2\ 4\ 1]$  | $g(2) = [2\ 4]$                          |   |
| $M := \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ | $f(2) = \begin{bmatrix} 2\ 4 \\ 1\ 0 \end{bmatrix}$                 | $g(2) = [2\ 4]$                          |   |

**Funktionen als Argumente.** Auch bei Prozeduren können die Argumente selbst Funktionen sein, deren Argumentzahl allerdings bei der Definition festgelegt wird, genau wie bei einfachen Funktionen. Als Ausgabekanal sind allerdings nur Variablen geeignet, also eine als Argument übergebene Funktion kann nicht als Nebenwirkung umdefiniert werden.

## 10.2. Ablaufsteuerung

Das sind Anweisungen, die den Ablauf des Programms steuern. Dazu gehören

- Verzweigung
- Schleifen
- Abfangen von Fehlern

### 10.2.1. Verzweigung

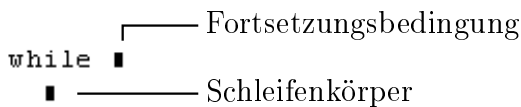
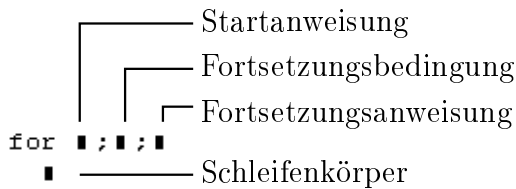
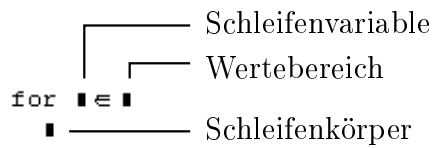
Die Verzweigung dient der Ausführung von verschiedenen Programmteilen abhängig vom Wert logischer Ausdrücke. Das wird mit der `if`-Anweisung realisiert:

```
if Bedingung
  Programmteil wenn Bedingung erfüllt ist.
else
  Programmteil wenn Bedingung nicht erfüllt ist.
```

Die Bedingung ist ein logischer Ausdruck, 0 entspricht falsch, jeder andere Wert entspricht wahr.

### 10.2.2. Schleifen

Schleifen dienen der wiederholten Ausführung von Anweisungen. Eine häufige Anwendung ist die elementweise Bearbeitung von Listen (Matrizen).



**Mengenschleife**, bei der die Schleifenvariable nacheinander alle Werte des Wertebereichs annimmt. Der Wertebereich ist meist ein Vektor von Zahlen, der mit der Funktion `range` erzeugt wird.

**Kopfgesteuerte Schleife**. Zuerst wird die Startanweisung ausgeführt. Ist die Fortsetzungsbedingung erfüllt, wird der Schleifenkörper ausgeführt und anschließend die Fortsetzungsanweisung ausgeführt und die Fortsetzungsbedingung erneut geprüft.

**Kopfgesteuerte Schleife**. Die Schleife wird durchlaufen, solange die Fortsetzungsbedingung erfüllt ist.

Wenn eine Schleifenvariable  $i$  ganzzahlige Werte von 1 bis  $n$  annehmen soll, dann schreibt man unter Verwendung der Funktion `range(2)`

```
n := 5 for i ∈ [1..n]
      |
      | fo [Tab] i [↔] rang [Tab]
```

So kann man beispielsweise einen Vektor der Kubikzahlen erzeugen. Beachten Sie, dass die Variable  $c$  nicht als Matrix vordefiniert werden muss. Sie wird dynamisch genau so groß dimensioniert, dass das Element mit dem höchsten Index noch reinpasst.

```
n := 5 for i ∈ [1..n]
      |
      | c_i := i^3
      |
      | [c[i Leer]:i^3]
      |
      | c = [ 1
            8
            27
            64
            125 ]
```

Als Wertebereich der Schleifenvariable kann jeder Vektor oder jede Matrix angegeben werden. Matrizen werden dabei mit nur einem Index indiziert, wobei die Elemente zeilenweise abgearbeitet werden (lineare Indizierung). Im folgenden Beispiel ist  $v$  die Schleifenvariable, die nacheinander die Werte aus der Matrix  $M$  annimmt. Die Variable  $i$  ist der Index für den Ergebnisvektor  $c$ , in den die Werte der Schleifenvariable nacheinander geschrieben werden, so dass die Bearbeitungsreihenfolge (zeilenweise) sichtbar wird.

```
M := [ 1, 2 2, 4
       3, 1 4, 3
       5, 8 6, 7 ]
i := 0 for v ∈ M
      | i := i + 1
      | c_i := v
      | c = [ 1, 2
            2, 4
            3, 1
            4, 3
            5, 8
            6, 7 ]
```

Bei der folgenden Schleife kommt es aber auf die Bearbeitungsreihenfolge nicht an.

```
M := [ 1, 2 2, 4
       3, 1 4, 3
       5, 8 6, 7 ]
s := 0 for v ∈ M
      | s := s + v
      | s = 23, 5
```

Häufig kann man die Benutzung von Schleifen vermeiden, zum Beispiel mit der elementweisen Funktionsanwendung, mit Indexvektoren oder mit der Summenfunktion:

$$\begin{array}{c} \longrightarrow \\ \left( [1 \dots n] \right)^3 = \begin{bmatrix} 1 \\ 8 \\ 27 \\ 64 \\ 125 \end{bmatrix} \end{array}
 \quad
 c := M [1 \dots \text{length}(M)] = \begin{bmatrix} 1, 2 \\ 2, 4 \\ 3, 1 \\ 4, 3 \\ 5, 8 \\ 6, 7 \end{bmatrix}
 \quad
 \sum M = 23, 5$$

Beachten Sie, dass die elementweise Anwendung und Indexvektoren für den Leser Ihres Rechenblatts eventuell erklärungsbedürftig sind.

### 10.2.3. Angabe der Schrittweite

Gerade wenn man aus Funktionen Grafikobjekte erzeugen will, ist es oft bequemer, der Schleifenvariable direkt die  $x$ -Werte zuzuweisen. Dazu dient die Funktion `range(3)`, die einen Vektor durch Angabe des ersten, des zweiten und des letzten Arguments definiert. Das folgende Beispiel zeigt, wie es funktionieren soll:

$$\begin{array}{c} x_1 := 0 \quad x_2 := 10 \quad x_n := 50 \\ \text{rang} \downarrow \text{Tab} \quad x.1 \rightarrow x.n \rightarrow x.2 \\ [x_1; x_2 \dots x_n] = \begin{bmatrix} 0 \\ 10 \\ 20 \\ 30 \\ 40 \\ 50 \end{bmatrix} \end{array}$$

Aus dem ersten und dem zweiten Wert wird das Inkrement (die Schrittweite) bestimmt. Die Schleife wird beginnend mit dem ersten Wert abgearbeitet. Dann wird die Schleifenvariable um das Inkrement erhöht. Wenn damit der Endwert noch nicht überschritten wurde, wird der Durchlauf wiederholt.

### 10.2.4. Abfangen von Fehlern

Bei der Ausführung des Rechenblatts können Fehler auftreten. Abhängig von den Einstellungen unter **Berechnung** > **Im Fehlerfall** wird die Ausführung fortgesetzt, abgebrochen oder angehalten (und kann im Debug-Modus fortgesetzt werden). Der Text der Fehlermeldung wird dabei in die Variable `lasterror` geschrieben.

Die Kontrollstruktur `try/on error` erlaubt es, beliebige, auch unerwartete Fehler innerhalb des Programms abzufangen, so dass es auf keinen Fall zu einem Abbruch kommt:

$$\begin{array}{c} \text{try} \\ \quad \text{Programnteil, wo der Fehler auftreten könnte.} \\ \text{on error} \\ \quad \text{Programnteil, der ausgeführt wird, wenn es} \\ \quad \text{zu einem Fehler gekommen ist.} \end{array}$$

Hier ein Beispiel.  $f(x)$  berechnet den Kehrwert des Arguments. Im Fall  $x = 0$  führt das auf ein nicht definiertes Ergebnis und SMATH erkennt einen Fehler. Insbesondere in

Schleifen oder Vektoren ist es ärgerlich, wenn man überhaupt kein Ergebnis bekommt, obwohl nur für einzelne Werte Probleme auftreten. Das kann man durch Abfangen des Fehlers vermeiden.

```

a := [-2..2] →  $\frac{1}{a}$       lastError = ""
f(x) := try
  res := 1/x
on error
  res := lastError
res

```

$f(a) = \begin{bmatrix} -0,5 \\ -1 \\ \text{"Division durch Null."} \\ 1 \\ 0,5 \end{bmatrix}$

Es liegt in der Verantwortung des Anwenders, sicherzustellen, dass ein abgefangener Fehler das Rechenergebnis nicht ungültig macht. Daher kann es sinnvoll sein, die in `lastError` gespeicherte Fehlermeldung zu prüfen. Beachten Sie, dass der Text von Fehlermeldungen von der Spracheinstellung abhängig ist.

Mit `error(Fehlermeldung)` kann man auch ein eigenes Fehlerereignis mit eigener Fehlermeldung setzen.

## 10.3. Fehlersuche

Bei der Fehlersuche in Programmen wird der Anwender durch folgende Funktionen in SMath unterstützt:

- die dynamische Hilfe: Sie zeigt für alle definierten Objekte die Belegung an.
- die Protokollfunktion `trace()` und das Ausgabefenster,
- der Debugger. Das ist ein Werkzeugsatz zur Untersuchung des Rechenablaufs. Damit können Sie
  - die Berechnung an beliebig wählbaren Punkten und jederzeit anhalten,
  - die Berechnung nach einem Halt schrittweise oder bis zum nächsten Haltepunkt fortsetzen,
  - jederzeit die Werte der definierten Variablen und Funktionen (den Kontext) inspizieren,

Die Funktion `trace()` kann in Anweisungsblöcke integriert werden, um zur Laufzeit die Belegung von Variablen anzuzeigen. Die Ausgabe erfolgt im Ausgabefenster, welches mit **Ansicht** ein- und ausgeschaltet werden kann.

Die Verwendungsmöglichkeiten und die Ausgabeformatierung werden im [SMath Wiki](#) erläutert. Im einfachsten Fall schreibt man die anzuzeigenden Variablen oder Ausdrücke als Argumente.

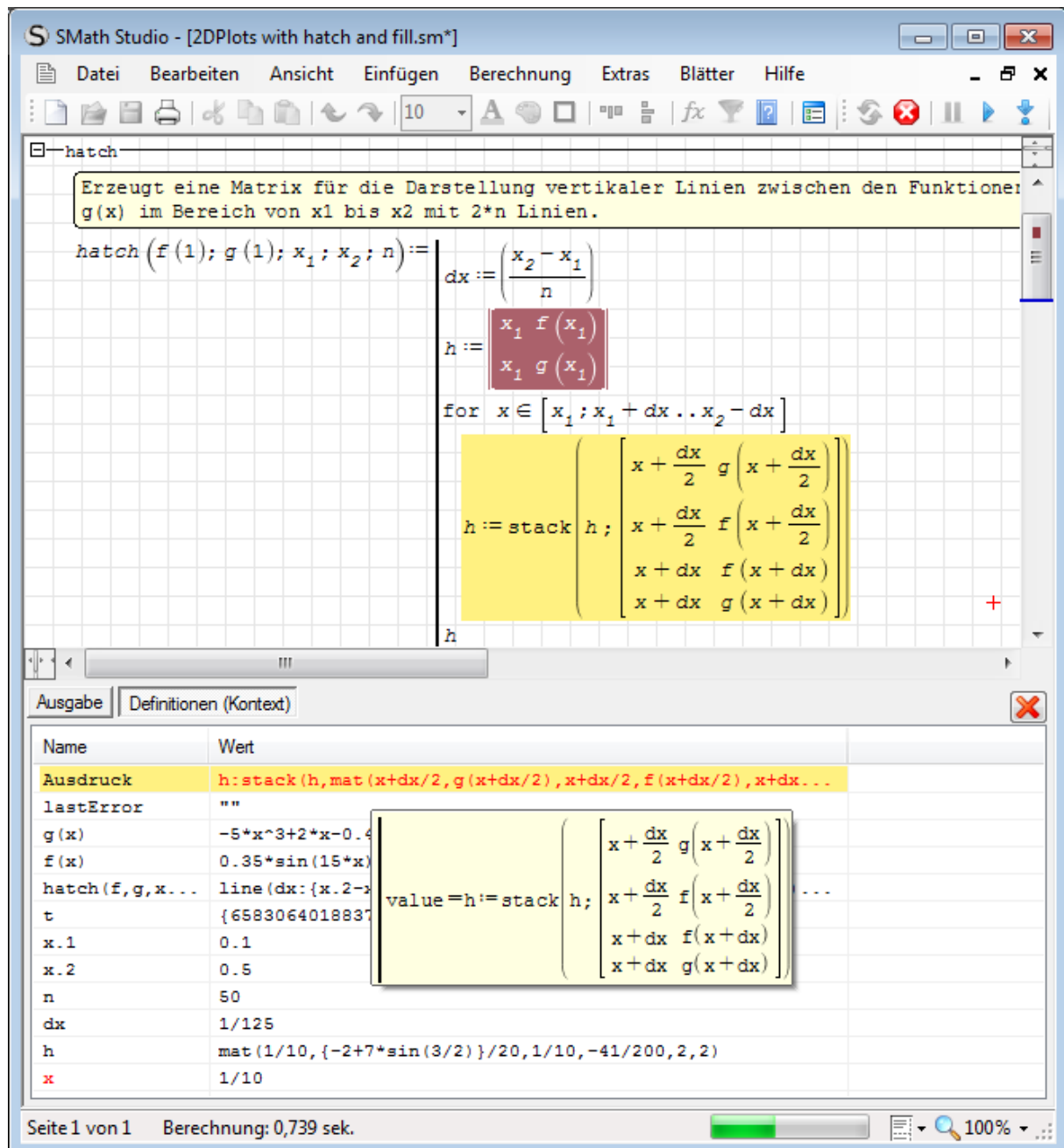








Abbildung 10.1.: SMath Studio im Debug-Modus. Unten das Debug-Fenster mit den definierten Namen und ihren Werten. Lässt man den Mauszeiger darüber schweben, wird dynamisch eine formatierte Version des Werts angezeigt. Rot markiert sind die Namen, deren Werte sich im letzten Schritt geändert haben. Haltepunkte sind braun, die aktuelle Anweisung ist gelb markiert.

Ein typischer Arbeitsablauf mit dem Debugger sieht so aus:

- Sie setzen einen **Haltepunkt** an einer Stelle, wo Sie den Stand der Berechnung sichten wollen.
  - Ausdruck markieren
  - **Kontextmenü** > **Haltepunkt ein/aus**
  - Der markierte Ausdruck wird braun hinterlegt.

- Starten der Berechnung mit  oder  (letzteres arbeitet einen Einzelschritt ab)
- Beim ersten Haltepunkt hält die Berechnung an. Sie sind nun im Debug-Modus und können die definierten Namen mit den zugehörigen gespeicherten Ausdrücken (den Kontext) sichten. Öffnen Sie dazu das Debug-Fenster mit **Ansicht > Debug-Fenster** (Abbildung 10.1).
- Sie haben nun die folgenden Optionen:
  -  oder `[ESC]` Abbruch der Berechnung und Verlassen des Debug-Modus (Inhalt des Debug-Fensters wird nicht mehr aktualisiert)
  -  Fortsetzung der Berechnung (bis zum Ende oder zum nächsten Haltepunkt oder bis die Pause-Taste gedrückt wird)
  -  Weiter zur nächsten Anweisung.
-  Dauert die Rechnung länger, können Sie auch ohne gesetzte Haltepunkte jederzeit mit der Pause-Taste in den Debug-Modus wechseln.

## 10.4. Funktionen für Zeichenketten

SMath Studio verfügt über einige eingebaute Zeichenkettenfunktionen. Viele der im Folgenden dargestellten Funktionen werden durch das Plugin *Custom Functions* bereitgestellt.

### Zusammenfügen

```
concat("abc"; "def"; "ghi") = "abcdefghi"
strjoin(", "; { "bread", "milk", "sugar", ... }) = "bread, milk, sugar, ..."
```

### Teilketten herausgreifen

Startposition und Länge können angegeben werden.

```
substr("abcdefghijkdef"; 5) = "efghijkdef"
substr("abcdefghijkdef"; 5; 3) = "efg"
```

Aufteilen anhand von Trennzeichen:

```

strsplit("the QUICK bRoWn FoX..."; " ") = [
  "the"
  "QUICK"
  "bRoWn"
  "FoX..."
]

strsplit("the QUICK bRoWn FoX..."; "o") = [
  "the QUICK bR"
  "Wn F"
  "X..."
]

```

### Teilketten ersetzen oder löschen

```

strrep("abcdefghijkldef"; "def"; "_DEF_") = "abc_DEF_ghijk_DEF_"
strrep("abcdefghijkldef"; "def"; "") = "abcghijk"

```

### Teilketten suchen

Die Funktion `findstr()` liefert einen Vektor mit den Anfangspositionen der Teilkette in einer gegebenen Zeichenkette. Wenn die Teilkette nicht vorkommt, liefert die Funktion den Wert -1.

```

findstr("abcdefghijkldef"; "def") = [ 4
                                     12 ]
findstr("abcdefghijkldef"; "ab") = [ 1 ]
findstr("abcdefghijkldef"; "xy") = -1

```

### Länge bestimmen

```

strlen("12345") = 5

```

### Prüfen, ob eine Zeichenkette vorliegt:

```

IsString("abc") = 1
IsString((sqrt(2))) = 0
IsString({ "abc"
           "1" }) = 0   Ausdruck ist vom Typ "Liste" (system)

```

### Ausdruck in Zeichenkette umwandeln

Die Ausdrücke werden symbolisch ausgewertet und in die interne Textdarstellung umgewandelt. Man kann kein Format für die Zahlen vorgeben.



```

num2str(2,3) = "23/10"
num2str(a+b) = "a+b"
num2str({ "a"
          pi }) = "sys("a", pi, 2, 1) "
num2str([ 1 2
          3 4 ]) = "mat(1,2,3,4,2,2) "
num2str(int(x^2, x)) = "int(x^2, x) "

```

Die Ausgabe von Ergebnissen mit Angabe der Nachkommastellen erfolgt mit `var2str()`:

```

var2str(ss) = "ss"           Liefert den Namen undefinierter Variablen
                             als Zeichenkette
x := pi
var2str(x) = "3,14159265358979" Liefert den Wert als Zeichenkette, Zahl der
var2str(x; 2) = "3,14"      Nachkommastellen kann angegeben werden.

```

### Zeichenketten in Ausdrücke umwandeln

`str2num()` ist die Umkehrfunktion zu `num2str()`.

```

num2str({ "abc"
          sqrt(2) }) = "sys("abc", sqrt(2), 2, 1) "
symbolische Optimierung str2num(num2str({ "abc"
                                           sqrt(2) })) = { "abc"
                                                         sqrt(2) }
symbolische Optimierung str2num("sqrt(x^2+y^2)") = sqrt(x^2 + y^2)

```

### Groß- und Kleinschreibung

Diese Funktionen sind im Plug-in *Custom Functions* enthalten. Anders als die Standardfunktionen für Zeichenketten können sie auch auf Matrizen und Listen angewendet werden, die Zeichenketten enthalten.

Alles in Kleinbuchstaben umwandeln:

```

strtolower("the QUICK bRoWn FoX...") = "the quick brown fox..."
strtolower([ "the QUICK bRoWn FoX..."
              sqrt(2)
              "another sTRING" ]) = [ "the quick brown fox..."
                                      sqrt(2)
                                      "another string" ]

```

Alles in Großbuchstaben umwandeln:

```

strtoupper("the QUICK bRoWn FoX...") = "THE QUICK BROWN FOX..."
strtoupper([ "the QUICK bRoWn FoX..."
              sqrt(2)
              "another sTRING" ]) = [ "THE QUICK BROWN FOX..."
                                      sqrt(2)
                                      "ANOTHER STRING" ]

```

Anfangsbuchstaben der Zeichenketten in Großbuchstaben umwandeln:

```
ucfirst("the QUICK bRoWn FoX...") = "The QUICK bRoWn FoX..."
ucfirst([[ "The QUICK bRoWn FoX..." ],
         [ "another sTRING" ]]) = [[ "The QUICK bRoWn FoX..." ],
                                   [ "Another sTRING" ]]
```

Anfangsbuchstaben aller Wörter in Zeichenketten in Großbuchstaben umwandeln:

```
ucwords("the QUICK bRoWn FoX...") = "The QUICK BRoWn FoX..."
ucwords([[ "The QUICK bRoWn FoX..." ],
         [ "another sTRING" ]]) = [[ "The QUICK BRoWn FoX..." ],
                                   [ "Another sTRING" ]]
```

### Beschreibungstext von Definitionen abfragen

Nähere Erläuterungen zu Beschreibungstexten sind in in Abschnitt 5.13 zu finden.

```
Hier wird a definiert
a := 1, 32
description(a) = "Hier wird a definiert"

Vier Definitionen auf einmal
[ a := 1 b := 2 ]
[ c := 3 d := 4 ]
description(a) = "Vier Definitionen auf einmal"
description(b) = "Vier Definitionen auf einmal"
```

### Zeichenkodierung

SMath Studio kann Unicode-Zeichen aus 4-stelligen Hexadezimalzahlen generieren. Allerdings werden keine Unicode-Zeichen sondern die SMath-spezifische Ersatz-Zeichenkette gespeichert, so dass die korrekte Anzeige so erzeugter Zeichen auf Formelbereiche beschränkt ist:

```
a := concat("\", "005E", "\") = "^"
Formatierte Textanzeige      Tabellenbereich
\005E\                       \005E\
a                              Table...
                               [ a ]
```

Maxima bietet Funktionen für die Umwandlung von ASCII-Code und Unicode in Zeichen und umgekehrt:

```
(M) (ascii(97)) = "a"
(M) (cint("a")) = 97
```

```

M ( unicode ( 945 ) ) = "α"
M ( cint ( "α" ) ) = 945

```

Die folgenden Funktionen wandeln Listen von Zahlencodes in Zeichenketten um und umgekehrt:

```

M ( string_to_octets ( "abc" ) ) = { 97
                                   98
                                   99 }
M ( octets_to_string ( { { 100 }
                        { 101 } } ) ) = "de"

```

## 10.5. Code-Bausteine

Code-Bausteine, in der englischen Version *snippets*, sind normale Rechenblätter (.sm-Dateien), die zum Einfügen in andere Dateien vorgesehen sind, um dort bestimmte Funktionen bereitzustellen. Das geschieht in den Schritten:

- Schreiben des gewünschten Inhalts in eine .sm-Datei.
- Festlegen eines Kurznamens für den späteren Aufruf: **Datei** > **Eigenschaften: Dateiattribute**: Feld „Baustein-Tastaturkürzel“. Dieser Name wird zusammen mit der Beschreibung in der dynamischen Hilfe angezeigt.
- Ausfüllen der Titel, Beschreibung, Autor und Firma werden in der Erweiterungsverwaltung (siehe Abschnitt 12.1) angezeigt. Zudem zeigt die dynamische Hilfe die Beschreibung an, wenn man das Tastaturkürzel eingibt.
- .sm-Datei in den Ordner *extensions/snippets* der SMath-Installation verschieben.
- Beim nächsten Neustart ist die Datei dem SMath-System bekannt.
- Bausteine können als Blattbereich eingefügt werden. Er ist standardmäßig eingeklappt ist und hat Namen des Bausteins als Titel.

Diese Funktion wurde in SMath 0.90 eingeführt. Sehen Sie hier das [Ankündigungsvideo](#)<sup>2</sup>.

Eingefügte Bausteine sind völlig gleichwertig zu anderweitig eingefügten Anweisungen im Rechenblatt, es gibt also keinen getrennten Namensraum und keine Eingabe- und Ausgabeargumente. Das muss von den Funktionsdefinitionen im Baustein geregelt werden.

Code-Bausteine kann man auch aus der Online-Galerie laden oder dort einstellen, dies wird im Kapitel „Erweiterungen“, Abschnitt 12.3 erklärt.

### 10.5.1. Erzeugung eines Bausteins

Die Schritte werden anhand einer Funktion zum Schraffieren der Fläche zwischen zwei Kurven in 2D-Diagrammen erläutert. Das kann man beispielsweise folgendermaßen realisieren:

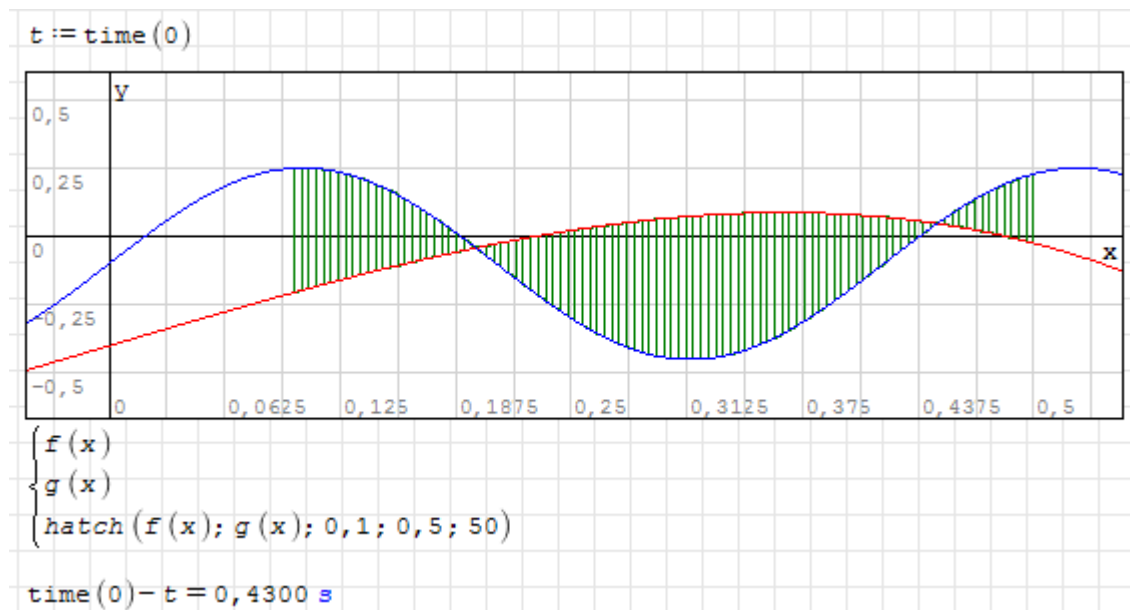
<sup>2</sup>[http://en.smath.info/forum/yaf\\_postst639\\_Snippets-feature.aspx](http://en.smath.info/forum/yaf_postst639_Snippets-feature.aspx)

$$f(x) := 0,35 \cdot \sin(15 \cdot x) - 0,1 \quad g(x) := -5 \cdot x^3 + 2 \cdot x - 0,4$$

Erzeugt eine Matrix für die Darstellung vertikaler Linien zwischen den Funktionen  $f(x)$  und  $g(x)$  im Bereich von  $x_1$  bis  $x_2$  mit  $2 \cdot n$  Linien.

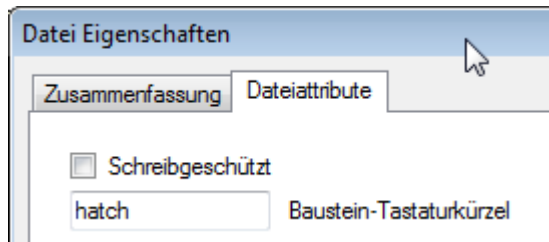
```
hatch(f(1); g(1); x1; x2; n) :=
  dx := (x2 - x1) / n
  h := [
    [x1 f(x1)]
    [x1 g(x1)]
  ]
  for x ∈ [x1; x1 + dx .. (x2 - dx)]
    h := eval stack h; [
      [x + dx/2 g(x + dx/2)]
      [x + dx/2 f(x + dx/2)]
      [x + dx f(x + dx)]
      [x + dx g(x + dx)]
    ]
  h
```

Es wird ein Polygonzug erzeugt, der abwechselnd der einen und der anderen Funktion folgt und dazwischen hin- und herspringt. Die vertikalen Verbindungslinien erzeugen dann die Schraffur.

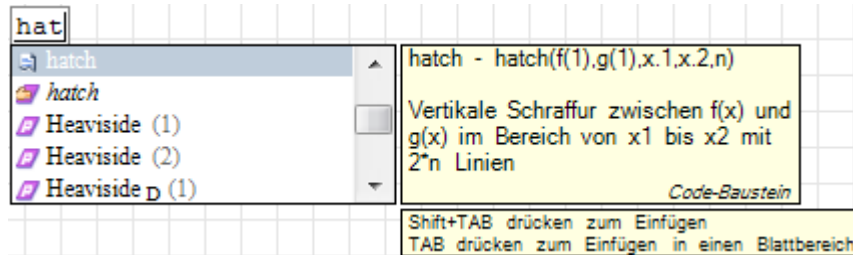


Grundsätzlich reicht es zunächst, die Funktion `hatch()` in eine `.sm`-Datei beliebigen Namens zu schreiben und diese im Ordner `extensions\snippets` der SMath-Installation abzuliegen. Dann findet sie die Code-Baustein-Verwaltung beim nächsten Aufruf und von dort kann man den Baustein ins aktuelle Rechenblatt einfügen.

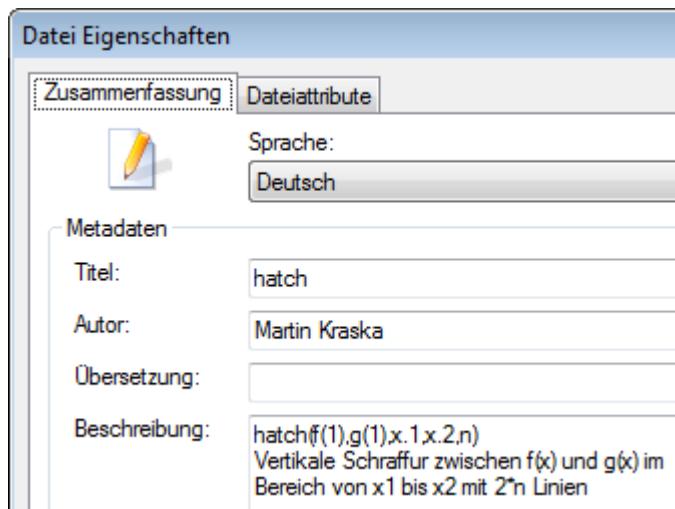
Gibt man in den Dateieigenschaften das „Tastaturkürzel“ an,



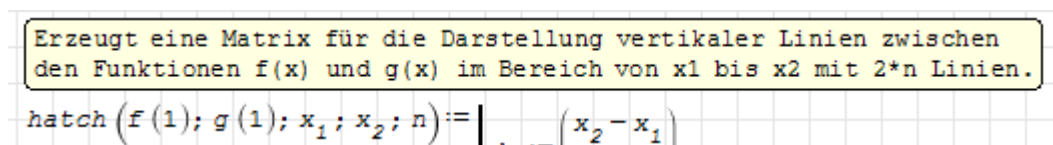
dann kennt auch die dynamische Hilfe den Baustein:



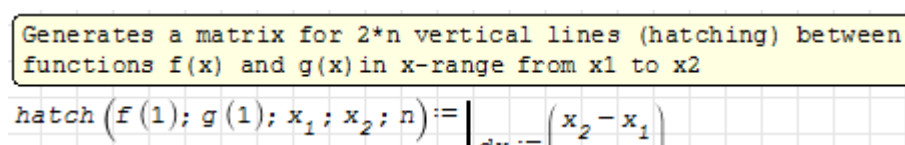
Der Hilfetext wird dabei den Dateieigenschaften (Beschreibung) entnommen. Diesen sollte man mindestens auch auf Englisch hinterlegen.



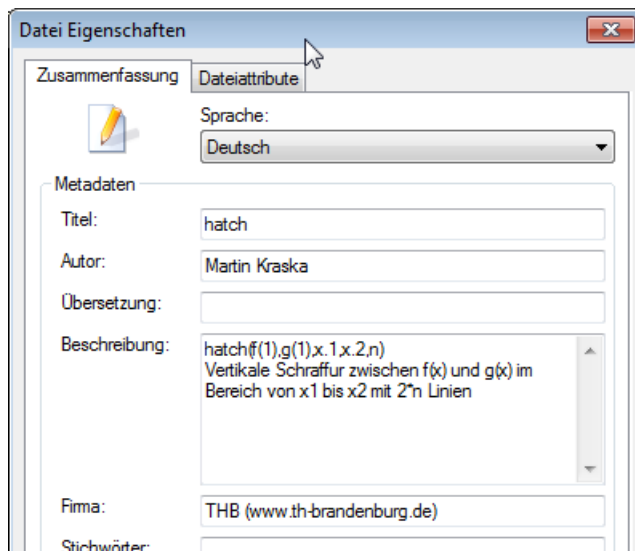
Zusätzlich kann und sollte man auch die Funktionen im Baustein mit Beschreibungen versehen. In unserem Fall also beispielsweise so:



Bausteine, die Sie im SMath-Forum veröffentlichen wollen, sollten mindestens noch eine englische Beschreibung erhalten:



Die Datei-Eigenschaften sollte man stets (gegebenenfalls auch mehrsprachig) auszufüllen.



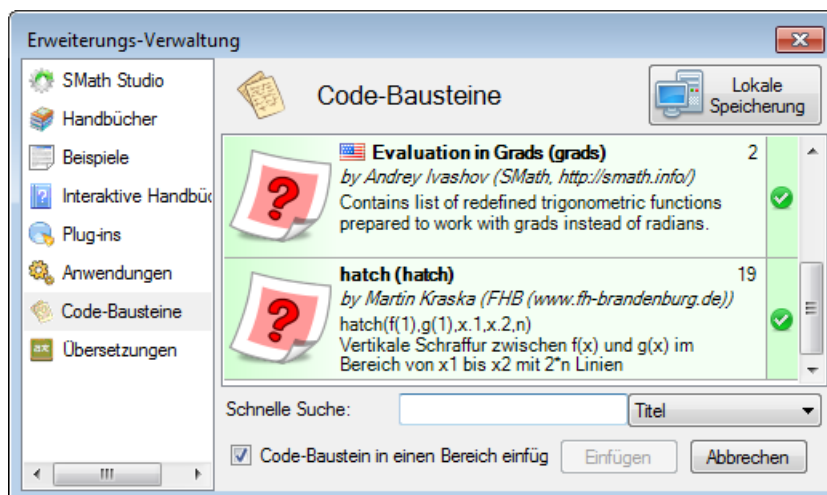
In der Baustein-Verwaltung werden der Titel, die Beschreibung, das Baustein-Tastaturkürzel, der Autor und die Firma angezeigt, siehe nächster Abschnitt.

Wie Sie Code-Bausteine installieren und veröffentlichen können, wird in Kapitel 12 erläutert.

### 10.5.2. Verwendung von Bausteinen

Zunächst muss der Baustein ins Rechenblatt eingefügt werden. Dafür gibt es zwei Optionen:

1. Durch Eintippen des Namens an der gewünschten Stelle des Rechenblatts. Dann erscheint die dynamische Hilfe. Der Baustein-Name kann dann mit  $\leftrightarrow$  bestätigt werden, damit wird der Baustein in einem ausgeblendeten Blattbereich eingefügt.
2. **Extras** > **Code-Baustein-Verwaltung** ruft den Dialog „Erweiterungs-Verwaltung“ mit aktiviertem Bereich „Code-Bausteine“ auf. Dort kann der Baustein ausgewählt und per **Einfügen** ins aktuelle Rechenblatt eingefügt werden.



Wird in der dynamischen Hilfe mit  $\leftrightarrow$  bestätigt oder in der Baustein-Verwaltung der Haken bei „Code-Baustein in einem Bereich einfügen“ stehen gelassen, so wird der Inhalt des Bausteins in einen ausgeblendeten Blattbereich eingefügt. Der Titel wird gemäß Dateieigenschaften gesetzt.

⊞ hatch

Alternativ wird der Inhalt direkt ins Arbeitsblatt eingefügt (ohne ausblendbaren Blattbereich).

In jedem Fall sind die enthaltenen Definitionen mit ihren Beschreibungen nun der dynamischen Hilfe bekannt und können verwendet werden.

hatch

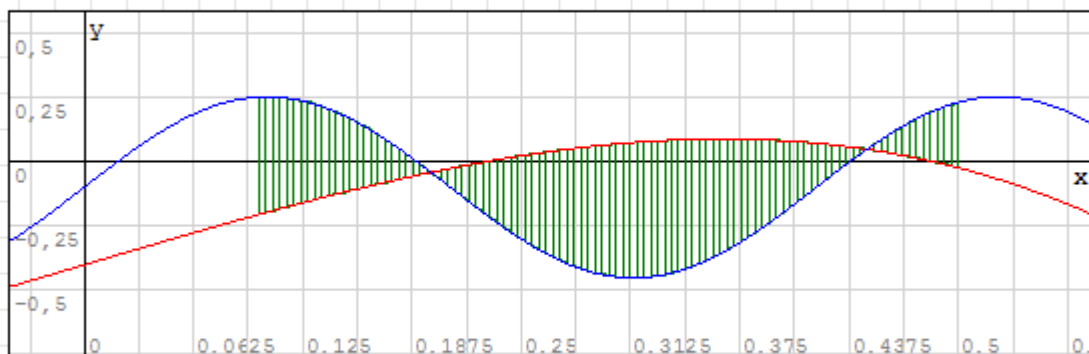
- Heaviside (1)
- Heaviside (2)
- Heaviside\_D (1)
- Heaviside\_D (2)

Erzeugt eine Matrix für die Darstellung vertikaler Linien zwischen den Funktionen  $f(x)$  und  $g(x)$  im Bereich von  $x_1$  bis  $x_2$  mit  $2 \cdot n$  Linien.

$$\text{hatch}(f(x); g(x); x_1; x_2; n) = \begin{cases} dx := \frac{x_2 - x_1}{n} \\ h := \begin{bmatrix} x_1 & f(x_1) \\ x_1 & g(x_1) \end{bmatrix} \\ \text{for } x \in [x_1; x_1 + dx .. (x_2 - dx)] \\ h := \text{eval} \left( \text{stack} \left( h, \begin{bmatrix} x + \frac{dx}{2} & f\left(x + \frac{dx}{2}\right) \\ x + \frac{dx}{2} & g\left(x + \frac{dx}{2}\right) \end{bmatrix} \right) \right) \\ h \end{cases}$$

$$f(x) := 0,35 \cdot \sin(15 \cdot x) - 0,1 \quad g(x) := -5 \cdot x^3 + 2 \cdot x - 0,4$$

⊞ hatch



```
{ f(x)
  g(x)
  hatch(f(x); g(x); 0,1; 0,5; 50)
```

# 11. Der SMath Viewer

Der Inhalt dieses Kapitels basiert auf einem [Forum-Beitrag](#)<sup>1</sup> des Entwicklers Andrey Ivashov.

Seit der Version 0.96 kann SMath das aktuelle Rechenblatt als interaktive Anwendung exportieren. Die Bereiche des Rechenblatts werden in Bedienelemente umgewandelt, in denen der Anwender Zahlenwerte ändern kann, nicht aber das dahinter liegende Rechenblatt.

SMath Studio dient dazu, technische Berechnungen durchzuführen und nachvollziehbar zu dokumentieren. Wenn die Berechnungen durch Dritte ausgeführt werden soll, die dafür nicht die volle Dokumentation benötigen, ist der Export als ausführbare Anwendung eine sinnvolle Option. Vorteile dieses Vorgehens sind:

- Der Anwender benötigt keine SMath-Installation, somit können auch keine Versionskonflikte entstehen.
- Der Anwender sieht nur die Ein- und Ausgabefelder sowie genau die Inhalte, die der Entwickler dem Anwender sichtbar machen will. Die Berechnungslogik kann vor dem Anwender verborgen werden.
- Der Entwickler kann die Rechnung in SMath Studio erstellen und dokumentieren (was in dieser Form z.B. in Excel nicht möglich ist).
- Die Anwendung kann direkt aus dem Rechenblatt exportiert werden.
- Der Anwender benötigt keine Erfahrung mit SMath.
- Der Anwender kann die Berechnung nicht versehentlich beschädigen, es wird immer das ausgeführt, was Sie ausgeliefert haben.
- Es kann im Einzelfall zu Geschwindigkeitsvorteilen kommen, da bei Änderungen immer nur der Inhalt der gerade angezeigten Registerkarte neu berechnet wird.
- Trotz des komplett enthaltenen SMath-Systems sind die exe-Dateien vergleichsweise kompakt (ab 0,5 MB, je nach Zahl und Größe der benutzten Plug-ins).

SMath Viewer ist ein Auslieferungsweg für Arbeitsblätter, und SMath Studio die zugehörige Entwicklungsumgebung.

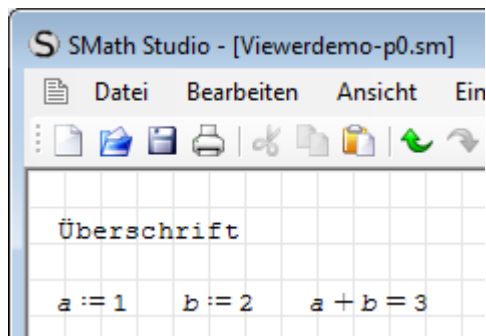
## 11.1. Einführungsbeispiel

Erzeugen Sie ein beliebiges Arbeitsblatt mit ein paar Variablen, Formeln und etwas Text in SMath Studio und wählen Sie im Hauptmenü **Datei** > **Speichern unter**

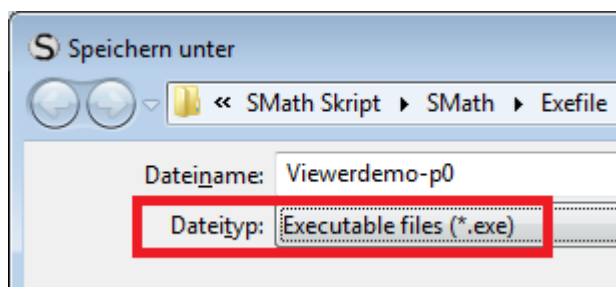
---

<sup>1</sup>[http://en.smith.info/forum/yaf\\_postsm9315\\_SMath-Viewer.aspx#post9315](http://en.smith.info/forum/yaf_postsm9315_SMath-Viewer.aspx#post9315)

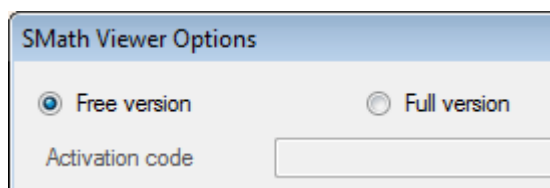




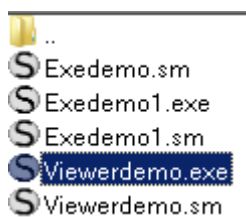
Wählen Sie im Dialog „Speichern unter“ als Dateityp „Executable files (\*.exe)“ aus und drücken Sie .



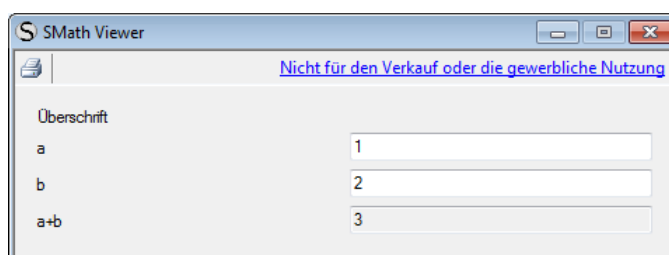
Beim Abspeichern hat man die Wahl zwischen der freien und der Vollversion (siehe Abschnitt 11.4).



Nach Bestätigung im Auswahldialog erscheint in dem von Ihnen gewählten Verzeichnis die einsatzbereite ausführbare Anwendung.



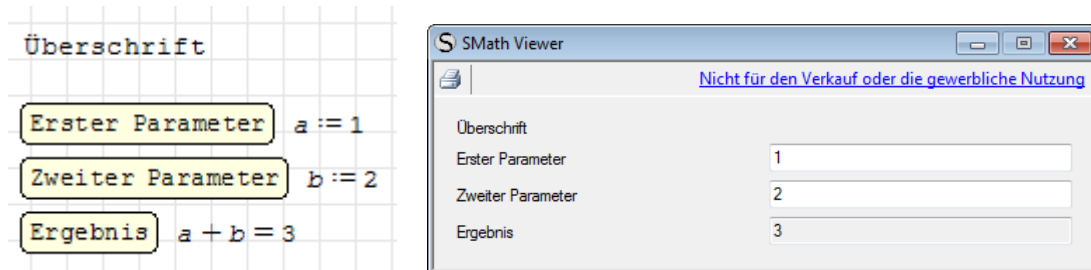
Starten Sie diese mit Doppelklick und Sie sehen die SMath Viewer-Oberfläche mit betriebssystemtypischen Bedienelementen.



## 11.2. Gestaltungsmöglichkeiten

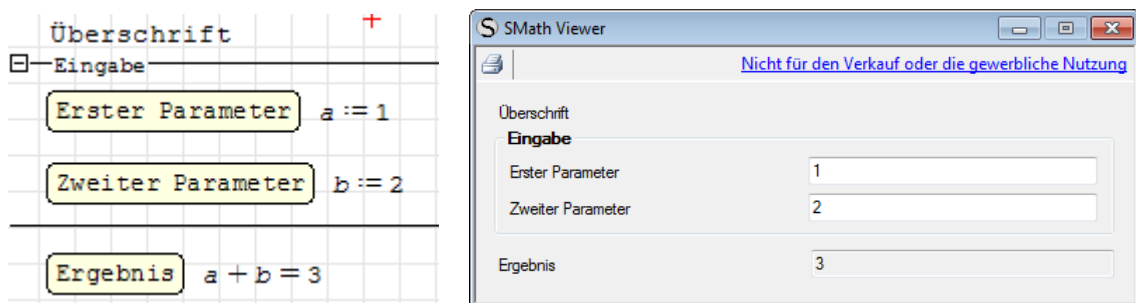
Damit das Programm besser bedienbar wird, brauchen wir einige Gestaltungselemente, die SMath Studio auch bereithält.

Einige Bereichstypen können mit Beschreibungstexten (siehe Abschnitt 5.13) versehen werden. Diese werden im Viewer anstelle der Formel oder des Variablennamens angezeigt:

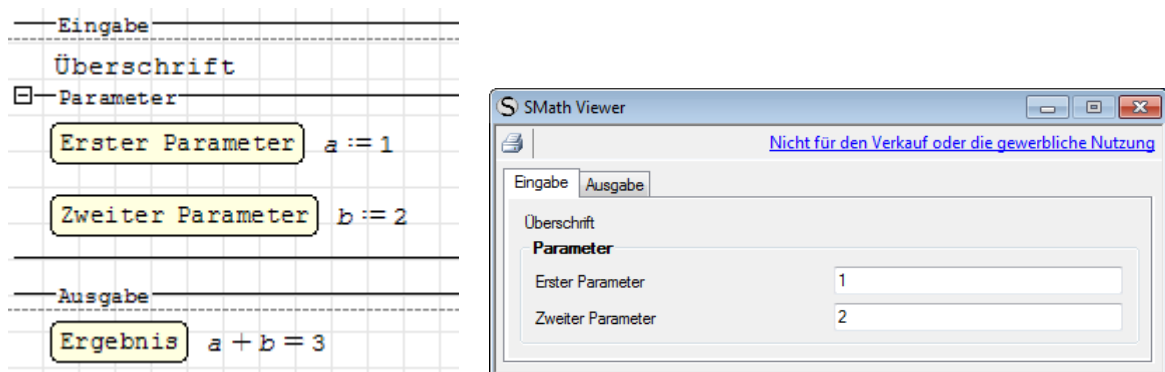


Von den Beschreibungstexten werden dabei die ersten 26 Zeichen und nur die jeweils erste Zeile benutzt.

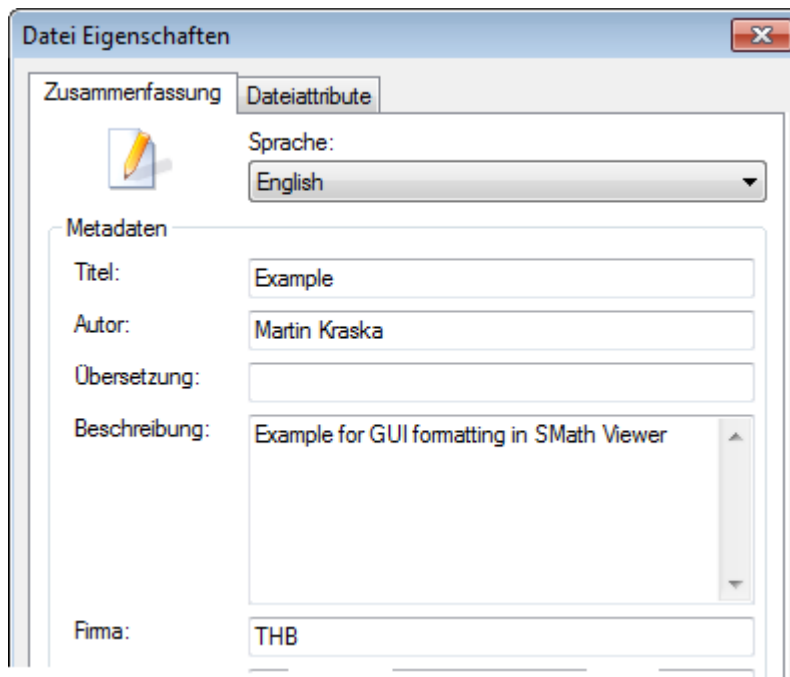
Benutzen Sie Blattbereiche, um deren Inhalt in der Vieweransicht zu gruppieren, die Gruppe kann auch einen Namen bekommen:



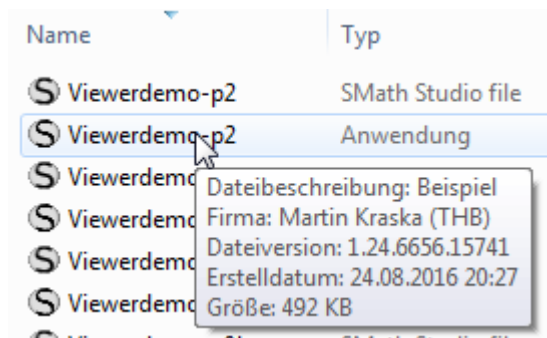
Mit Trennlinien können Sie Ihre Anwendung in mehrere Registerkarten gliedern. Trennlinientitel werden zu Namen der Registerkarten. Fehlt der Titel, werden Nummern vergeben.



Unter **Datei > Eigenschaften** können Sie Angaben zu Ihrem Dokument in verschiedenen Sprachen machen.

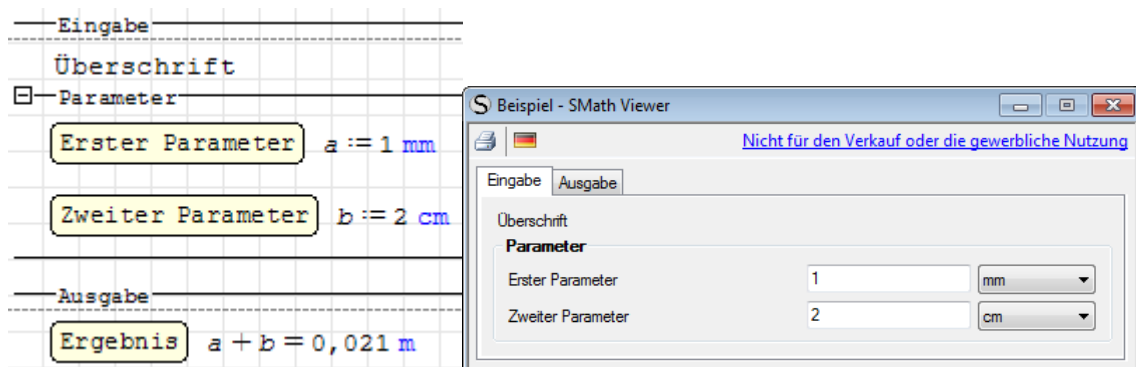


Im Dateimanager werden die Einträge entsprechend der Windows-Spracheinstellung angezeigt.



Der Fenstertitel zeigt den Titel aus den Dateieigenschaften in der eingestellten Sprache an.

Maßeinheiten werden unterstützt. Sowohl bei Ein- als auch bei Ausgabe kann die Maßeinheit aus einer Liste gewählt werden. Diese ist beschränkt auf die dimensionsmäßig passenden Einheiten.



## 11.3. Anwendungshinweise

### Plug-in-Unterstützung

Die exportierte Viewer-Anwendung enthält alle in der Rechnung benötigten Plug-ins. Das Verhalten erzeugter Bereiche (z.B. Plots) kann aber vom Verhalten im Rechenblatt abweichen.

Das Maxima-Plug-in wird leider nicht unterstützt.

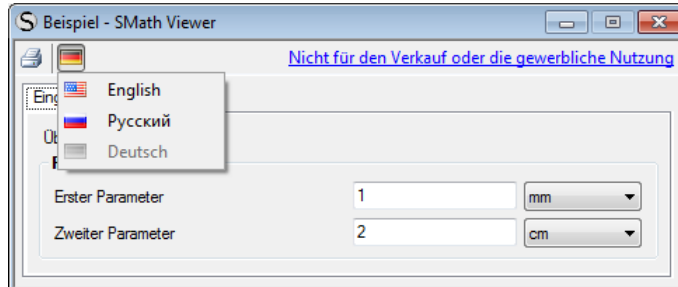
Werden keine externen Plug-ins benutzt, bleibt die ausführbare Datei unter 0,5 MB.

### Sprachanpassung

Sie können Ihre Anwendung mehrsprachig machen, indem Sie bei allen Texten alternative Sprachvarianten angeben.

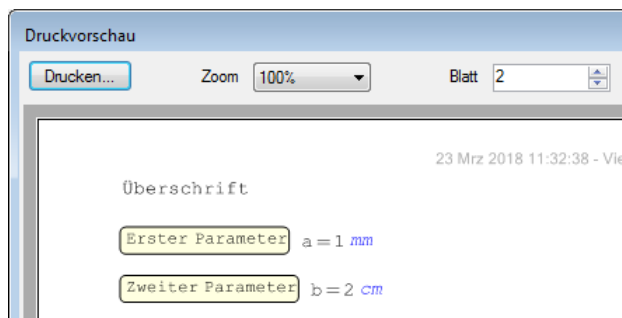
In der Regel kann die voreingestellte Sprache im Kontextmenü des jeweiligen Bereichs oder der Beschreibung geändert werden, so dass die Eingabe alternativer Texte möglich ist. Bei umfangreichen Übersetzungen kann es vorteilhaft sein, vorübergehend die Spracheinstellung in SMath zu ändern. Siehe auch Abschnitt 5.15 für mehrsprachige Arbeitsblätter.

Der Viewer zeigt die Anwendung standardmäßig passend zur Systemkonfiguration des Benutzers an. In der Werkzeugleiste kann man aber auch die anderen im Rechenblatt verfügbaren Sprachen anwählen.



### Druckansicht

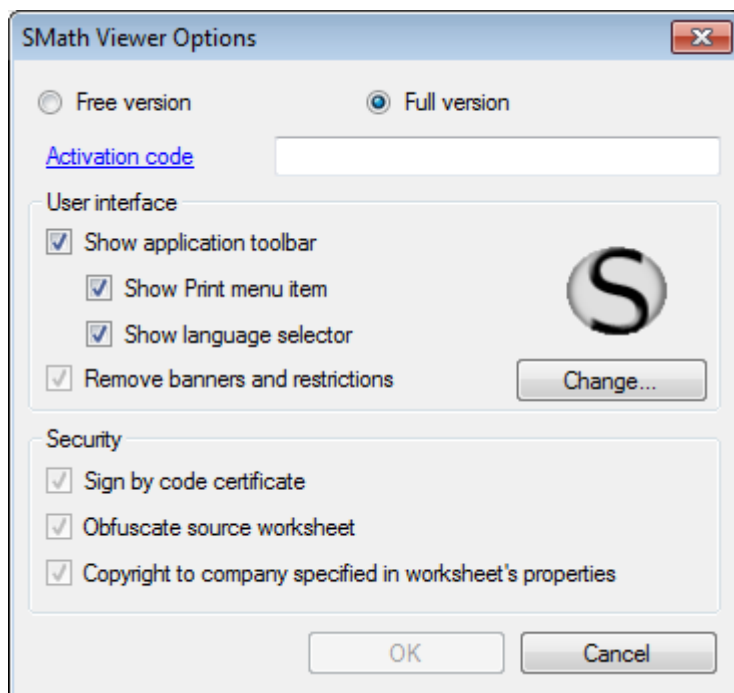
In der Anwendung steht eine Druckansicht zur Verfügung, die der ursprünglichen Rechenblattansicht entspricht. Seitenumbrüche durch Trennlinien erzeugen auch hier neue Seiten.



## 11.4. Lizenz

Der SMath Viewer ist gegenwärtig (März 2019) für die private Nutzung und für Bildungszwecke kostenlos. Dabei sind der Verkauf oder die kommerzielle Nutzung der erzeugten Anwendungen nicht gestattet, ein entsprechender Hinweis findet sich auf der Benutzeroberfläche.

Eine kommerzielle Version (*Full version*) befindet sich in der Erprobung. Diese erfordert einen Freischaltcode, den man unkompliziert über den Link im Auswahldialog beim Exportieren bekommen kann.



Besonderheiten bei kommerziellen Lizenzen:

- Signierung durch ein Code-Zertifikat. Dadurch wird die Datei gegen Modifikation durch Viren oder Hackerangriff geschützt und die sichere Ausführung gewährleistet.
- Das Rechenblatt wird verschleiert, so dass es nicht möglich ist, aus der exe-Datei das ursprüngliche Rechenblatt zu extrahieren.
- Das Anwendungs-Icon kann in jedes gewünschte geändert werden.
- Die Kompilation (Export als Anwendung) erfolgt online auf dem SMath.info-Server. Dadurch werden die Verschleierung des Rechenblatts und die Signierung ermöglicht. Das schließt allerdings auch Plug-ins von Drittanbietern aus.
- Verkauf und kommerzielle Nutzung von Viewer-Anwendungen sind erlaubt (Beschränkungshinweis in der Oberfläche entfällt).
- Die Oberfläche kann durch Unterdrückung einzelner Elemente angepasst werden.

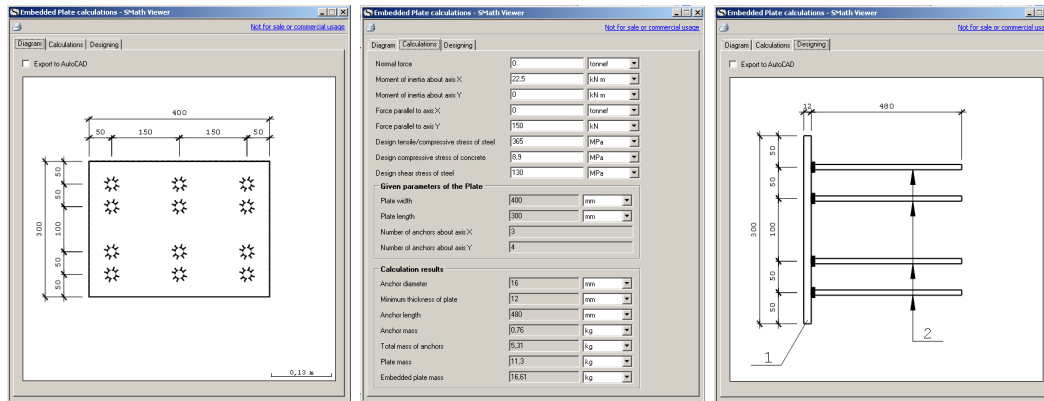
## 11.5. Veröffentlichung in der SMath-Online-Galerie

SMath-Rechenblätter können als Viewer-Datei (Anwendung) in der Online-Galerie von SMath Studio veröffentlicht werden. Sie werden auf dem SMath-Server kom-

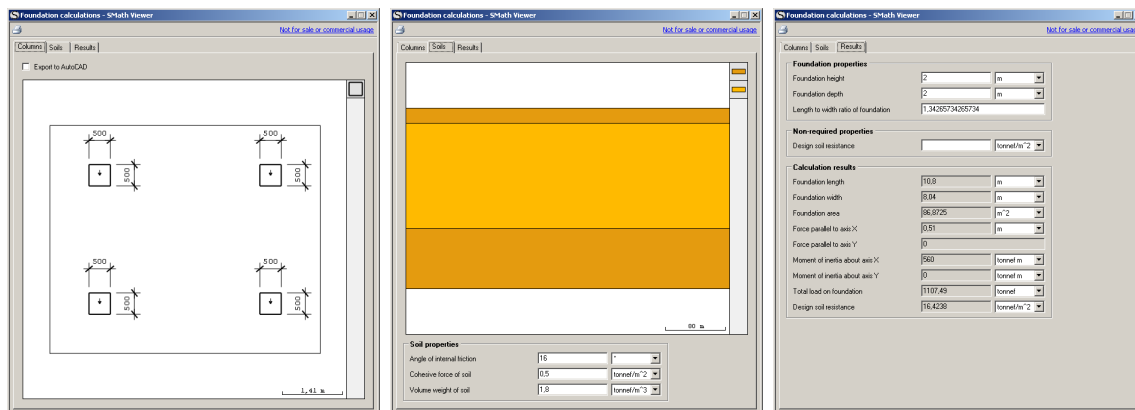
piliert. Anwender können sich die ausführbare Datei über die Erweiterungsverwaltung herunterladen und lokal ausführen.

Hier einige Beispiele:

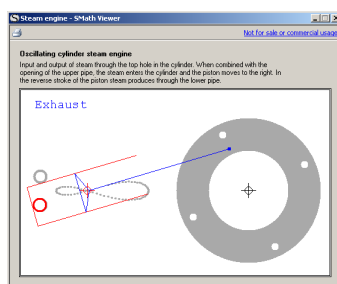
**EmbeddedPlate<sup>2</sup>** (ENG, RUS). Hier wird das Modeller-Plug-in benutzt.



**Foundation<sup>3</sup>** (ENG, RUS). Die erste Registerkarte enthält interaktive Bedienelemente.



**Oscillating.exe<sup>4</sup>** (ENG, RUS) Beispiel mit Animation.



<sup>2</sup><http://smath.info/?file=740169>


<sup>3</sup><http://smath.info/?file=740186>

<sup>4</sup><http://smath.info/?file=740185>

## 12. Erweiterungen

Anwender können auf verschiedenen Wegen zur Erweiterung und Verbesserung von SMath beitragen. Diese Beiträge können auf [smath.info](https://en.smath.info) registriert und hochgeladen werden und sind dann für alle Nutzer aus der SMath Studio-Anwendung heraus verfügbar. Dieser Mechanismus nennt sich Online-Galerie.

**Handbücher** sind einzelne Dateien im PDF- oder Windows-Hilfe-Format.

**Interaktive Handbücher** sind Gruppen miteinander verlinkte SMath-Dokumente, die mit **Werkzeuggestreife**  aufrufbar sind und in einem speziellen Sichtfenster geöffnet werden.

**Beispiele** Das sind einzelne .sm-Dateien, die als normale Rechenblätter geöffnet werden.

**Plug-ins** (Anwendungserweiterungen). Das sind dynamisch ladbare Funktionsbibliotheken (dll), die den Funktionsumfang von SMath Studio erweitern.

**Anwendungen.** Das sind Rechenblätter, die als eigenständige Programme (SMath Viewer) exportiert wurden (.exe-Dateien).

**Code-Bausteine** (sogenannte „snippets“). Das sind einzelne .sm-Dateien, deren Inhalt in das aktuelle Rechenblatt eingefügt werden kann, z.B. um Daten oder Funktionen bereitzustellen.

**Übersetzungen.** Die Benutzeroberfläche von SMath Studio ist mehrsprachig. Wenn in Ihrer Sprachversion Übersetzungen fehlen oder falsch sind, können Sie das selbst ändern.

SMath Studio enthält einen Erweiterungsmanager (siehe nächster Abschnitt), mit dem man Erweiterungen auf dem eigenen Rechner installieren und verwalten kann.

Auf der [SMath-Internetseite](https://en.smath.info)<sup>1</sup> kann man die verfügbaren Erweiterungen sichten.

Das Registrieren und Hochladen von Erweiterungen unterscheidet sich je nach Typ:

- Plugins werden als C#-Projekt auf den SVN-Server hochgeladen und über den **Release Manager** auf dem SMath-Server kompiliert und für die Online-Galerie freigegeben.
- Handbücher (PDF oder Windows Hilfe), Interaktive Handbücher, Beispiele und Code-Bausteine werden über den **Extension Manager** hochgeladen.

### 12.1. Erweiterungsmanager in SMath Studio

Der Erweiterungsmanager kann auf verschiedenen Wegen aufgerufen werden. Sie unterscheiden sich nur in der ausgewählten Kategorie, in der man landet:

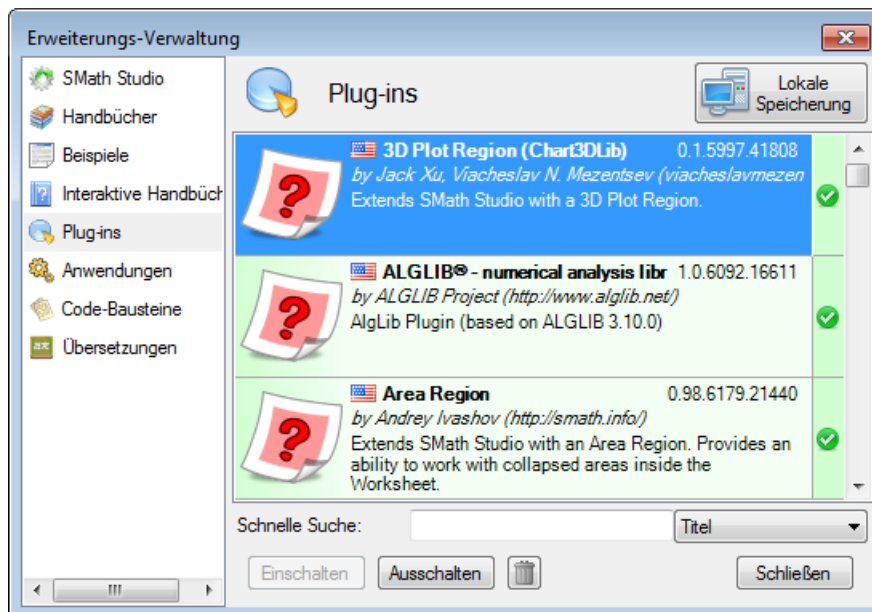
---

<sup>1</sup><https://en.smath.info/view/SMathStudio/summary>

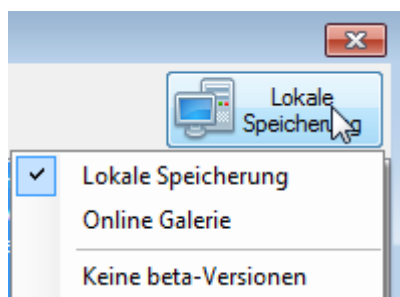
Extras > Plug-ins

Extras > Code-Baustein-Verwaltung


Hilfe > Beispiele...




Der Erweiterungsmanager zeigt nach Aufruf die lokal auf Ihrem Rechner gespeicherten Erweiterungen an. Für den Zugriff auf die online verfügbaren Erweiterungen schalten Sie auf die Online-Galerie um (Knopf rechts oben):



Sie können wählen, ob auch Erweiterungen im Beta-Stadium (vom Entwickler als noch nicht gebrauchsfertig eingestuft) angezeigt werden sollen.

Die heruntergeladenen Erweiterungen werden im Einstellungsverzeichnis gespeichert (siehe Abschnitt 4.5). Sie sind dort unter kryptisch bezeichneten Unterverzeichnissen zu finden. Unter  > **Handbuch** > **Anwendungserweiterungen (Plug-ins)** finden Sie eine Seite mit aktiven Links zu den Plug-in-Verzeichnissen. Normalerweise gibt es aber keinen Grund, diese Verzeichnisse zu besuchen.

Zusätzlich zu Download/Installation und Update können Sie je nach Erweiterungstyp weitere Einstellungen vornehmen oder Aktionen auslösen:

- SMATH Studio: Automatische Update-Abfrage ein-/ausschalten.
- Handbücher: Dokument öffnen.
- Beispiele: In SMATH Studio öffnen. Ein Link führt auf eine Registrierungsseite für Beispiele.
- Interaktive Handbücher: Als Standard für den -Knopf setzen und Handbuch öffnen.



- Plug-ins: Einschalten, Ausschalten, Löschen.
- Anwendungen: Ausführen.
- Code-Baustein: Ins aktuelle Rechenblatt einfügen (wahlweise als Blattbereich).
- Übersetzungen: Link zum Online-Übersetzungsportal

## 12.2. Erweiterungsarten

### 12.2.1. Handbücher (Handbook)

Dies sind Dateien, die vom Anwender aus SMath Studio heraus geöffnet werden können.

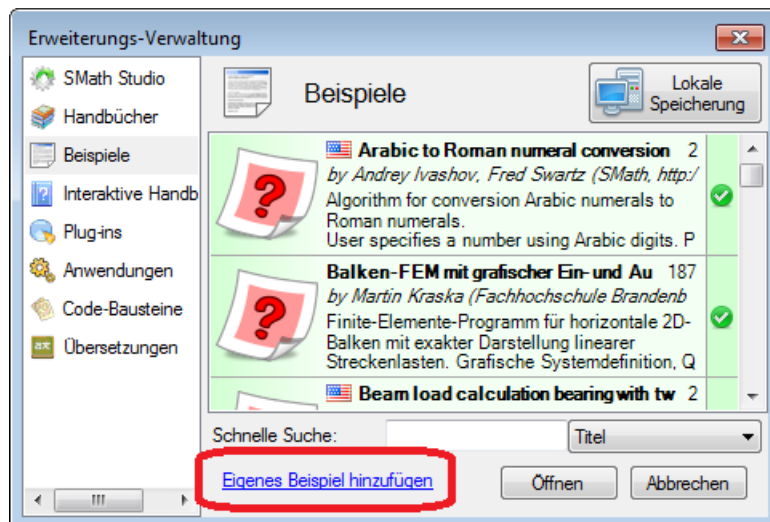
|                |  |
|----------------|--|
| Registrierung  | Allgemeines Registrierungsportal (siehe Abschnitt 12.3)  |
| Dateiformat    | SMath Studio, PDF oder Windows Hilfe   |
| Metadaten      | Dokumentsprache (Auswahlliste)<br>Dokumenttitel (erscheint in der Galerie)<br>Dokumentbeschreibung (unterer Textteil in der Galerie-Anzeige, wird umgebrochen)<br>Autor (wird nicht umgebrochen) |
| Identifikation | GUID, erzeugt aus dem Dateinamen des Dokuments. Updates müssen daher den gleichen Namen haben.   |
| Version        | JJJJ.MM.TT.SS (Jahr.Monat.Tag.Stunde) beim Hochladen.  |

### 12.2.2. Beispiele (Examples)

Beispiele sind Rechenblätter, die in der Online-Galerie bereitgestellt werden und aus dem Erweiterungsmanager heraus in SMath Studio geöffnet werden können.

|                |  |
|----------------|--|
| Registrierung  | <b>Beispiel-Registrierung</b> oder allgemeines Registrierungsportal (siehe Abschnitt 12.3)   |
| Dateiformat    | Rechenblatt (.sm oder .smz)  |
| Metadaten      | <i>Titel</i> , <i>Autor</i> und <i>Beschreibung</i> unter <b>Datei</b> > <b>Eigenschaften</b> in SMath Studio in mindestens einer Sprache ausfüllen. |
| Identifikation | <b>id</b> im <b>identity</b> -Block in der .sm-Datei sowie eindeutiger (und bei jedem Update gleicher) Dateiname.                                    |
| Version        | <b>revision</b> im <b>identity</b> -Block in der .sm-Datei (wird bei jeder Speicherung hochgezählt)  |

Anwender können selbst Beispiele für die Veröffentlichung vorschlagen.



Der Link „Eigenes Beispiel hinzufügen“ führt auf das [Beispiel-Registrierungsportal](#) mit Instruktionen und der Möglichkeit, Dateien hochzuladen. Sie werden von einem Administrator freigegeben, das geschieht also nicht sofort. Spätere Aktualisierungen erfolgen dann über das allgemeine Registrierungsportal, ohne dass eine weitere Freigabe nötig ist.

Beispiele, die mit den Standard-Plug-Ins auskommen, können auch direkt von der [Homepage](#)<sup>2</sup> aus in der Online-Version von SMATH (also ohne lokale Installation) geöffnet werden.

### 12.2.3. Interaktive Handbücher (Book)

Interaktive Handbücher sind Sätze von miteinander verlinkten SMATH-Dokumenten. Ein Beispiel dafür ist das interaktive Handbuch des Autors (siehe Abschnitt 4.3).

|                |  |
|----------------|--|
| Registrierung  | Allgemeines Registrierungsportal (siehe Abschnitt 12.3)  |
| Dateiformat    | <b>Unkomprimierte</b> .zip-Datei mit beliebigen Ordnern und SMATH-Dateien, <i>contents.sm</i> muss im Wurzelverzeichnis liegen.            |
| Metadaten      | Dateieigenschaften unter <b>Datei</b> > <b>Eigenschaften</b> für <i>contents.sm</i> in SMATH Studio in mindestens einer Sprache ausfüllen. |
| Identifikation | <b>id</b> im <b>identity</b> -Block in <i>contents.sm</i>  |
| Version        | <b>revision</b> im <b>identity</b> -Block in <i>contents.sm</i>  |

Im Erweiterungsmanager kann man die Handbücher herunterladen und eines zur Standardreferenz erklären. Dann ist es unter **Werkzeuggestreife** > aufrufbar.

Die Links erzeugt man am besten als Textbereiche mit der Funktion **Einfügen** > **Textbereich (utilities)** > **add new link to a worksheet**.

Solche Links können mit Strg-Klick angewählt werden und öffnen sich in der gleichen SMATH-Instanz anstelle des gerade offenen Dokuments.

Links in Writer-Bereichen hingegen werden in einer neuen SMATH-Instanz geöffnet. Das bläht den Desktop auf und dauert lange.

Pro und Contra zur Verwendung des Writer-Bereichs in interaktiven Handbüchern:

<sup>2</sup><http://en.smath.info/>

- Pro
  - Mehrere separate Links in einem Writer-Bereich möglich
  - Flexiblere Formatierung (bis hin zu einzelnen Zeichen)
- Contra
  - Man muss erst den Bereich anklicken und kann dann Links darin folgen.
  - Links werden in neuen SMath-Instanzen geöffnet, das dauert lang.
  - Die alternativen Texte in anderen Sprachen werden nicht als gelbe Sprechblase angezeigt. Das erschwert das mehrsprachige Editieren.

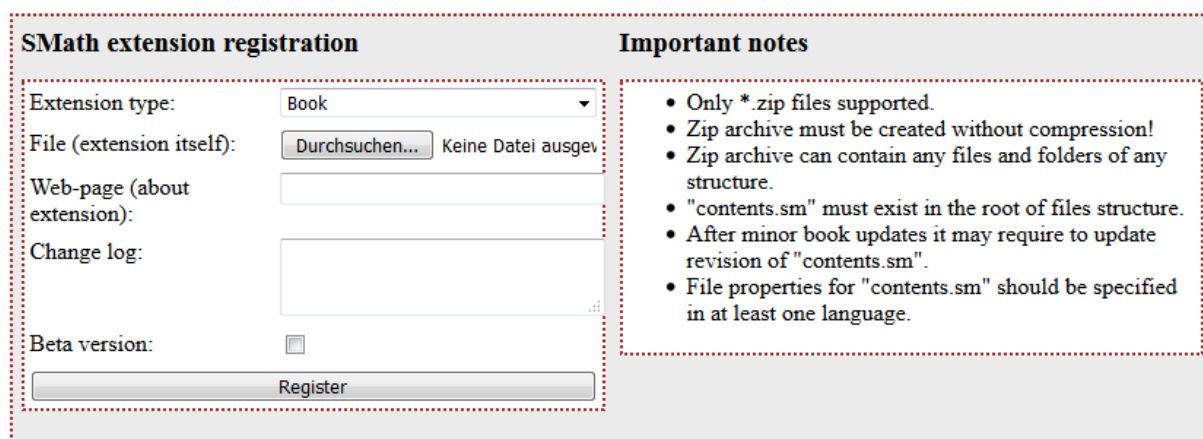


Abbildung 12.1.: Registrierung von interaktiven Handbüchern

### 12.2.4. Plug-ins

Plug-ins sind Programmergänzungen, die als kompilierte Anwendungserweiterung (dynamisch anbindbare Bibliothek, .dll) bereitgestellt werden und den Funktionsumfang von SMath ergänzen. Sie können über den Erweiterungsmanager installiert werden.

|                |  |
|----------------|--|
| Registrierung  | Release Manager  |
| Dateiformat    | Visual Studio-Projekt im <a href="#">SVN-Server</a>            |
| SMath-Version  | Auswahlliste, erforderliche SMath-Programmversion              |
| Metadaten      | Eigenschaften in der Entwicklungsumgebung für die .dll setzen. |
| Identifikation | Wird in der Entwicklungsumgebung gesetzt                       |
| Version        | Wird in der Entwicklungsumgebung gesetzt                       |


Der Autor dieses Handbuchs hat eine inoffizielle portable [Distribution](#)<sup>3</sup> zusammengestellt, die sowohl unter Linux als auch unter Windows funktioniert und in der viele Plug-ins vorinstalliert sind.

Was können Plug-ins bieten:

- Zusätzliche Funktionen. Diese werden in der dynamischen Hilfe aufgelistet, in der Regel mit einer Kurzbeschreibung. Bei allen Funktionen findet man in diesem Hilfetext den Hinweis, in welchem Plug-in sie bereitgestellt werden. Somit kann man den Adressaten für Fragen und Fehlermeldungen ermitteln.

<sup>3</sup><https://en.smath.com/wiki/SMath%20with%20Plugins.ashx>

- Zusätzliche Bereichstypen. Diese sind dann unter **Einfügen** verfügbar.
- Zusätzliche Exportformate. Diese erscheinen als mögliche Dateitypen unter **Datei> Speichern unter**

Informationen zu den einzelnen Plug-ins finden Sie im Interaktiven Handbuch ( > **Anwendungserweiterungen**) oder im Forum, wo es im **Subforum „Extensions“**<sup>4</sup> zu den Plug-ins meist auch Diskussionen gibt. Im vorliegenden Handbuch werden die Plug-ins nicht explizit vorgestellt, da dies zum Interaktiven Handbuch redundant wäre. Wenn die im Handbuch vorgestellten Funktionen Plug-ins erfordern, werden diese genannt.

#### 12.2.4.1. Upload of Plug-ins

To create a plug-in, follow the instructions laid out in the step-by-step description by Davide Carpi, available in the Extension Manager.

To publish the plug-in (new or updated):

- Upload your solution files to the SVN repository
- Go to the **Release Manager** and select tab „Add new“
- Specify the SVN path to the SVN directory (the one that contains your solution (.sln) file).
- Press „Build“. Your solution is compiled on the server. You can download a zip file with the generated files and a log of the compilation process.
- Press „Publish“. In the pop-up dialog
  - make sure to have the „Is approved“ box checked
  - check the „Is beta“ box according to your judgement
  - Add some text to the change log (copy it from your SVN log or from the announcement in the forum)
  - Press [OK]
- If all went right, you can find the new version in the Extension manager and on the **website**.

#### 12.2.5. Anwendungen (Applications)

Das sind SMath Viewer-Anwendungen (siehe Kapitel 11), die auf dem SMath-Server aus hochgeladenen .sm-Rechenblättern kompiliert werden. Dabei steht nur eine Untermenge der existierenden Plug-ins zur Verfügung.

|                |  |
|----------------|--|
| Dateiformat    | Rechenblatt (.sm oder .smz)  |
| Metadaten      | Dateieigenschaften unter <b>Datei&gt; Eigenschaften</b> in SMath Studio in mindestens einer Sprache ausfüllen. |
| Identifikation | <b>id</b> im <b>identity</b> -Block in der .sm-Datei   |
| Version        | <b>revision</b> im <b>identity</b> -Block in der .sm-Datei   |

<sup>4</sup>[http://en.smith.info/forum/yaf\\_topics32\\_Extensions.aspx](http://en.smith.info/forum/yaf_topics32_Extensions.aspx)

### 12.2.6. Code-Bausteine (Snippets)

Das sind SMath-Dokumente, die man in andere SMath-Dokumente einfügen kann, z.B. für häufig benötigte Definitionen. Ihre Erstellung und Verwendung wird in Abschnitt 10.5 erläutert.

|                |   |
|----------------|---|
| Dateiformat    | Rechenblatt (.sm oder .smz)   |
| Metadaten      | <i>Titel</i> , <i>Autor</i> und <i>Beschreibung</i> unter <b>Datei</b> > <b>Eigenschaften</b> in SMath Studio in mindestens einer Sprache ausfüllen. Dabei muss das <i>Tastaturkürzel</i> angegeben werden. Unter diesem Namen ist der Baustein in der dynamischen Hilfe bekannt. |
| Identifikation | id im <code>identity</code> -Block in der .sm-Datei   |
| Version        | revision im <code>identity</code> -Block in der .sm-Datei   |

## 12.3. Das Registrierungsportal

Auf dem [Registrierungsportal](https://en.smath.info/upload/extensions.aspx)<sup>5</sup> können Handbücher (PDF oder Windows Hilfe), Interaktive Handbücher, Beispiele und Code-Bausteine registriert und hochgeladen werden. Ablauf:

1. Erweiterung in das erforderliche Format bringen und gegebenenfalls die Metadaten bereithalten.
2. Registrierung im Portal <https://en.smath.info/upload/extensions.aspx>. Dazu müssen Sie ein Nutzerkonto mit Name und Passwort anlegen.
  - a) Erweiterungstyp wählen. Abhängig davon erscheinen rechts Hinweise. Diese sind unbedingt zu beachten.
  - b) Metadaten ausfüllen (typabhängig).
  - c) Datei wählen.
  - d) Gegebenenfalls Passwort für den Zugriff über den Erweiterungsmanager setzen.
  - e) Gegebenenfalls als Beta-Version kennzeichnen.
  - f) Registrierung und Hochladen starten mit .
3. Bei neuen Erweiterungen: Freigabe durch den Entwickler (Andrey Ivashov) abwarten. Updates benötigen keine Freigabe.
4. Die Erweiterung ist in der Online Galerie des Erweiterungsmanagers oder unter <https://en.smath.info/view/SMathStudio/summary> zu finden.

|  |                        |
|--|------------------------|
| <b>SMath extension registration</b>  | <b>Important notes</b> |
| Extension type: <input style="width: 150px;" type="text" value="Select type"/> | Select type to start.  |

Please, note: everything uploaded will be distributed under the following license:



Abbildung 12.2.: Das Registrierungsportal <https://en.smath.info/upload/extensions.aspx>

<sup>5</sup>[en.smath.info/upload/extensions.aspx](https://en.smath.info/upload/extensions.aspx)

The screenshot displays a web interface for managing uploaded files. It is divided into three main sections:

- Uploaded files:** A table showing a single file named "SMath Einführung.pdf" with a size of 5.02 MB and a download icon.
- Information:** A box containing system and user data:
 

|                  |                            |
|------------------|----------------------------|
| Your IP address: | 91.66.197.0                |
| Your login is:   | mkraska ( <b>Log out</b> ) |
| Your browser:    | Mozilla Firefox 27.        |
| Your OS:         | Windows                    |
| Current time:    | 07:34:57                   |
| Generated in:    | 0 sek. 124 mc.             |
- Create file:** A form with a "Choose file:" button labeled "Durchsuchen...", a text input for "Description:", and an "Upload file to server" button. The status "Keine Datei ausgewählt" is shown.
- Create folder:** A form with a "Folder name:" text input, a "Description:" text input, and an "Add folder" button.

Abbildung 12.3.: Portal zum Verwalten der eigenen hochgeladenen Erweiterungen (z.B. Löschen oder Download-Statistik ansehen) <http://smath.info/upload/default.aspx>

### 12.3.1. Versionierung von SMath-Dokumenten

Bei der Registrierung von Dateien gibt es je nach Dateityp unterschiedliche Wege, eine eindeutige Identität und Version zu erkennen.

Jedes Rechenblatt hat eine eigene Identifikationsnummer (`id`) und Version (`revision`). In einem Texteditor (hier mit XML-Syntax-Hervorhebung) kann man diese Daten in der `.sm`-Datei finden.

```
<regions>
  <settings>
    <identity>
      <id>94874e93-4882-4a0c-b0b6-4aba69703565</id>
      <revision>5</revision>
    </identity>
```

Die ID wird für neu erzeugte Dateien beim erstmaligen Abspeichern vergeben.

Achtung: Dateien, die durch **Datei > Speichern unter** oder im Betriebssystem kopiert werden, haben die gleiche ID!

**Neue, eindeutige ID erzeugen:**

1. Datei mit einem Editor öffnen, ganzen `<identity>...</identity>` Block löschen und Datei speichern.
2. Datei in SMath Studio laden und wieder speichern. Der Identifizierungsblock wird neu angelegt.

Die Version wird bei jedem Speichern in SMath Studio hochgezählt.

## 12.4. Quellcode auf dem SVN-Server

Der Quellcode für die externen Plug-ins und das Interaktive Handbuch liegen auf dem SMath **SVN-Server**<sup>6</sup>. SVN ist ein Quellcode-Verwaltungssystem, mit dem verschiedene Entwickler an einem Projekt gleichzeitig arbeiten können.

Will man eigene Plug-ins bereitstellen, muss deren Quelltext auf dem SMath SVN-Server liegen.

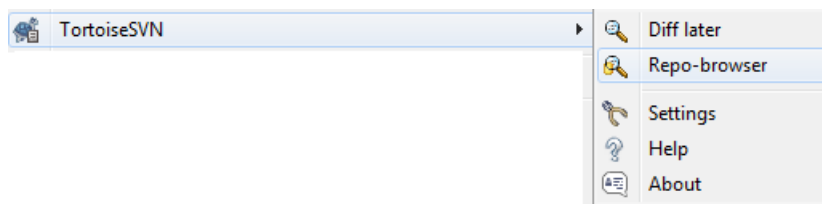
Mit einem entsprechenden Programm (SVN-Client) kann man Inhalt und Entwicklungsgeschichte sichten, Dateien herunterladen und hochladen.

Ein SVN-Client unter Windows ist **Tortoise SVN**<sup>7</sup>. Nach Download und Installation zeigt sich das Programm als Shell Extension, d.h. die Funktionen hängen sich ins Kontextmenü des Dateimanagers.

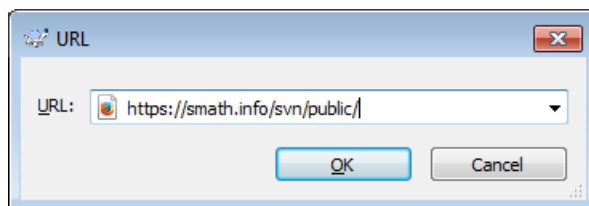
### Sichtung des SVN-Repositories

Für den Lesezugriff benötigt man keine Zugangsdaten zum Server. Wenn TortoiseSVN installiert ist, funktioniert der Zugriff so:

- Öffnen Sie mit Rechtsklick an irgendeiner Stelle im Dateimanager das Kontextmenü und wählen Sie **TortoiseSVN > Repo Browser**



- Geben Sie die Adresse des SVN-Servers ein:



- Dann erscheint ein Fenster zur Sichtung des Serverinhalts mit den Verzeichnissen:
  - handbooks** enthält ein Verzeichnis „Activebook“, mit den Dateien des Interaktiven Handbuchs,
  - plugins** enthält Unterverzeichnisse für alle Plug-ins

<sup>6</sup><https://smath.info/svn/public/>

<sup>7</sup><https://tortoisesvn.net/index.de.html>

# A. Glossar

**Bereich** (*region*) Grundelement, aus dem SMath-Rechenblätter zusammengesetzt werden. Sie werden frei auf dem Blatt angeordnet. Die verfügbaren Bereichstypen werden im Kapitel 5 erläutert.

**Bildbereich** (*picture region*) Bereich mit statischen, aber skalierbaren Rastergrafiken, die in der Regel aus der Zwischenablage eingefügt werden.

**Editierbarer Bildbereich** (*image region*) Bereich, der Vektor- und Rastergrafiken darstellt, welche mit externen Programmen aus SMath heraus editiert oder erzeugt werden können.

**Erweiterter Textbereich** (*writer region*) Textbereich mit erweiterten Formatierungsfunktionen.

**Formelbereich** (*math region*) Bereich mit aktiven Rechnungen (Formeln).

**Code-Bausteine** (*snippets*) SMath-Dokumente, die über den Erweiterungsmanager bereitgestellt werden und in das aktuelle Dokument eingefügt werden können. Erstellung und Verwendung von Code-Bausteinen werden in Abschnitt 10.5 behandelt.

**Dynamische Hilfe** (*dynamic assistant*) Auswahlfenster, welches beim Eintippen von Namen in Formelbereichen erscheint.

**Erweiterungsmanager** (*extension manager*) Verwaltung verschiedener Arten von Erweiterungen für SMath Studio mit Zugriff auf die Online-Galerie (zentral bereitgestellte Erweiterungen).

**Formatierte Textausgabe** (*conditionally formatted label*), Abschnitt 5.11.

**Funktion** Berechnungsvorschrift, die einen Wert liefert, der von anderen Werten (in der Argumentliste oder Variablen im aufrufenden Namensraum) abhängig ist. Abschnitt 6.5.

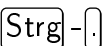






**Plug-in** Anwendungserweiterung, die zusätzliche Funktionen zum eigentlichen SMath-Programm bereitstellt. Abschnitt 12.2.4.



## B. Operatoren

Hier finden Sie eine vollständige Liste aller Operatoren in SMath Studio. Die meisten dieser Operatoren sind auch für Dritte, die SMath nicht kennen, verständlich.

Allerdings ist es häufig besonders für Anfänger schwer erkennbar, wie die Operatoren einzugeben sind. Solange die Paletten der Seitenleiste hier nicht alles abdecken oder die Funktionsnamen statt der Operatoren verwenden, ist diese Liste eine Hilfe.

| Darstellung   | Eingabe  | Bedeutung   |
|---|--|---|
| $m$   | 'm   | Maßeinheit  |
| $\bar{a}_b$   | a.b a,b  | Name mit Textindex  |
| $:=$  | :  | Zuweisung, Definition   |
| $=$   | =  | Numerische Auswertung   |
| $\overset{\text{Strg}}{=}$  |  -    | Symbolische Auswertung  |
| $\left. \begin{array}{c} \blacksquare \\ \blacksquare \end{array} \right  \blacksquare$                   | at    | Substitution (Plug-in <i>Custom Functions</i> )                         |
| $\left. \begin{array}{c} \blacksquare \\ \blacksquare \end{array} \right  \blacksquare \blacksquare$      | at  ; | Substitution (Plug-in <i>Custom Functions</i> )                         |
| $\left\{ \begin{array}{c} \blacksquare \\ \blacksquare \end{array} \right.$                               | sys   | Liste   |
| $\left[ \begin{array}{cc} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{array} \right]$ | mat   | Matrix  |
|  ( $\blacksquare$ )    | maxi  | Auswertung mit Maxima (Plug-in <i>Custom Glyphs</i> und <i>Maxima</i> ) |
| Arithmetik  |  |   |
| $\blacksquare + \blacksquare$ $\blacksquare + \blacksquare$   | +  | Positives Vorzeichen (reine Optik) oder Addition                        |
| $\blacksquare - \blacksquare$ $\blacksquare - \blacksquare$   | -  | Negatives Vorzeichen oder Subtraktion                                   |
| $\blacksquare \pm \blacksquare$ $\blacksquare \pm \blacksquare$   | Einfügen><br>Operator  | Plus/Minus  |
| $\blacksquare \mp \blacksquare$ $\blacksquare \mp \blacksquare$   | Einfügen><br>Operator  | Minus/Plus  |
| $\blacksquare \cdot \blacksquare$   | *  | Multiplikation  |
| $\blacksquare !$  | !  | Fakultät  |
| $\frac{\blacksquare}{\blacksquare}$   | /  | Division  |









|                     |                           |   |
|---------------------|---------------------------|---|
| $a^b$               | $\wedge$                  | Potenzierung  |
| $\log_a(b)$         | log $\leftrightarrow$     | Logarithmus mit angegebener Basis                           |
| $ a $               | abs $\leftrightarrow$     | Betrag einer Zahl   |
| $\sqrt[n]{a}$       | Strg- $\hat{\phantom{a}}$ | $n$ -te Wurzel  |
| $\sqrt{a}$          | \                         | Quadratwurzel   |
| $\lceil a \rceil$   | ce $\leftrightarrow$      | nächstgrößere Ganzzahl (Plug-in <i>Custom Functions</i> )   |
| $\lfloor a \rfloor$ | fl $\leftrightarrow$      | nächstkleinere Ganzzahl (Plug-in <i>Custom Functions</i> )  |
| $\bar{a}$           | conj $\leftrightarrow$    | komplex konjugierte Zahl (Plug-in <i>Custom Functions</i> ) |

Matrizen und Vektoren

|  |                                  |   |
|--|----------------------------------|---|
| $\{ \dots \}$                                  | sys $\leftrightarrow$            | Liste   |
| $\begin{bmatrix} \dots \\ \dots \end{bmatrix}$ | mat $\leftrightarrow$            | Matrix  |
| $\mathbf{a} \times \mathbf{b}$                 | Strg-8                           | Kreuzprodukt (Vektoren)                           |
| $ A $  | det $\leftrightarrow$            | Determinante einer Matrix                         |
| $A^T$  | tran $\leftrightarrow$           | Transposition einer Matrix                        |
| $A[i]$   | [                                | Elementadressierung in Feldern (Vektoren, Listen) |
| $A[i,j]$                                       | [ $\rightarrow$ ;                | Elementadressierung in Matrizen                   |
| $\sum a_i$                                     | sum $\leftrightarrow$            | Summe aller Elemente                              |
| $\sum_{i=1}^n a_i$                             | sum $\downarrow \leftrightarrow$ | Summe   |
| $\prod_{i=1}^n a_i$                            | pr $\leftrightarrow$             | Produkt   |
| $\text{rang}[a \dots a]$                       | rang $\leftrightarrow$           | Wertebereich (Vektor)                             |
| $\text{rang}[a; a \dots a]$                    | rang $\leftrightarrow$ ;         | Wertebereich, 2. Wert angebar                     |
| $\vec{f}$                                      | vec $\leftrightarrow$            | Elementweise Anwendung von Funktionen             |

Logische und Vergleichsoperatoren (Boolesch)







|          |        |                  |
|----------|--------|------------------|
| $a == b$ | $\sim$ | Logisches Gleich |
| $a < b$  | $<$    | Kleiner als      |

|                    |   |  |
|--------------------|---|--|
| <code>&gt;</code>  | <code>&gt;</code>   | Größer als                                   |
| <code>&lt;=</code> | <code>Strg-0</code>   | Kleiner oder gleich                          |
| <code>&gt;=</code> | <code>Strg-0</code>   | Größer oder gleich                           |
| <code>≠</code>     | <code>Strg-3</code>   | Ungleich                                     |
| <code>-</code>     | Panel Boolesch  | Logische Negation                            |
| <code>&amp;</code> | <code>&amp;</code>  | Logisches Und                                |
| <code> </code>     | <code> </code>  | Logisches Oder                               |
| <code>⊕</code>     | Panel Boolesch  | Exklusives Oder (XOR)                        |
| <code>lele</code>  | <code>lele</code>    | Intervall (Plug-in <i>Custom Functions</i> ) |
| <code>lelt</code>  | <code>lelt</code>    |  |
| <code>ltle</code>  | <code>ltle</code>    |  |
| <code>ltle</code>  | <code>ltle</code>    |  |
| <code>gege</code>  | <code>gege</code>    |  |
| <code>gegt</code>  | <code>gegt</code>   |  |
| <code>gtge</code>  | <code>gtge</code>  |  |
| <code>gtgt</code>  | <code>gtgt</code>  |  |

---

Integral- und Differenzialrechnung (Analysis)



---

|                                       |   |                                    |
|---------------------------------------|---|------------------------------------|
| <code>'</code>                        | <code>dif</code>     | Differentiation, 1. Ableitung      |
| <code><math>\frac{d}{d}</math></code> | <code>dif</code>  ;  | Differentiation, 1. Ableitung      |
| <code><math>\frac{d}{d}</math></code> | <code>dif</code>  ;; | Differentiation, höhere Ableitung  |
| <code>∫</code>                        | <code>int</code>     | unbestimmtes Integral              |
| <code>∫</code>                        | <code>int</code>  ;; | bestimmtes Integral                |
| <code>lim</code>                      | <code>lim</code>     | Grenzwert (Plug-in <i>Maxima</i> ) |

---

Programmierung












---

|                               |   |   |
|-------------------------------|---|---|
| <code>{ if otherwise }</code> | <code>cas</code>  ;; | Fallunterscheidung Intervall (Plug-in <i>Custom Functions</i> ) |
| <code>for ∈</code>            | <code>for</code>     | for-Schleife mit Wertebereich                                   |






|   |                                 |   |
|---|---------------------------------|---|
| <code>for █;█;█<br/>█</code>              | for $\boxed{\leftrightarrow}$ ; | for-Schleife mit Initialisierung,<br>Fortsetzungsbedingung und<br>Inkrementierung |
| <code>while █<br/>█</code>                | wh $\boxed{\leftrightarrow}$    | while-Schleife  |
| <code>if █<br/>█<br/>else<br/>█</code>    | if $\boxed{\leftrightarrow}$    | Bedingte Anweisung  |
| <code> █<br/>█</code>                     | lin $\boxed{\leftrightarrow}$   | Anweisungsblock   |
| <code>try<br/>█<br/>on error<br/>█</code> | try $\boxed{\leftrightarrow}$   | Abfangen von Fehlern  |

# C. Tastenbelegung

Hier werden nur die Tastenkürzel angegeben, die nicht direkt das erwartete Zeichen erzeugen.

| Taste         | Anweisung   | Funktion   |
|---------------|-------------|--|
| "             |             | Rechenblatt: Textbereich<br>Formelbereich: Zeichenkette  |
| ' (Hochkomma) |             | Platzhalter für vordefinierte Einheit, Funktion oder Konstante   |
| \             | sqrt()      |  Quadratwurzel  |
| [             | e1()        |  Platzhalter für Elementindex (Vektor, Matrix) erzeugen |
| ]             | line()      |  Anweisungsblock                                       |
| .             |             | Textindex an Namen   |
| ~             |             |  Platzhalter für Exponent                             |
| !             |             | Fakultät   |
| &             |             |  Logisches Und  |
|               |             |  Logisches Oder                                       |
| ~             |             |  Logisches Gleich                                     |
| @             |             | 2D-Diagramm  |
| Strg - .      |             | Symbolische Auswertung   |
| Strg - ^      | nthroot()   |  n-te Wurzel  |
| ↵             |             | Aktuellen Bereich verlassen  |
| Strg - ↵      |             | Rechenblatt oder Textbereich: Trennlinie einfügen  |
| ↑ - ↵         |             | Formelbereich: Umbruch (an +, ·, - oder =)   |
| Strg - Leert. |             | Textbereich: Zeilenumbruch   |
| ↔             |             | Dynamische Hilfe aktivieren  |
| ↔             |             | Gehe zum nächsten Bereich  |
| ↑ - ↔         |             | Gehe zum vorigen Bereich   |
| Strg - ↔      |             | Umschalten zum nächsten offenen Rechenblatt  |
| Strg - +      |             |  Logisches Gleich                                     |
| Strg - 1      | transpose() |  Transposition (Matrix)                               |
| Strg - 3      |             |  Ungleich   |

| Taste     | Anweisung | Funktion  |
|-----------|-----------|---|
| Strg-8    |           | $\times$ Kreuzprodukt (von Vektoren)  |
| Strg-9    |           | $\leq$ Kleiner oder gleich  |
| Strg-0    |           | $\geq$ Größer oder gleich   |
| Strg-A    |           | Alles markieren   |
| Strg-B    |           | Textbereich Fettschrift an/aus  |
| Strg-C    |           | Markierung in die Zwischenablage kopieren                                   |
| Strg-E    |           | Dialog „Funktion einfügen“  |
| Strg-F    |           | Suchfunktion aktivieren   |
| Strg-G    |           | vorherigen Buchstaben griechisch machen                                     |
| Strg-H    |           | Hyperlink(-bereich) einfügen  |
| Strg-I    |           | Textbereich Kursivschrift an/aus  |
| Strg-K    |           | Formelbereich: Sonderzeicheneingabe aktivieren/deaktivieren                 |
| Strg-M    | mat()     | Matrix  |
| Strg-N    |           | Neues Rechenblatt erzeugen  |
| Strg-O    |           | Dialog „Blatt Öffnen“   |
| Strg-P    |           | Dialog „Drucken“  |
| Strg-S    |           | Blatt speichern   |
| Strg-T    |           | Zeichenfläche (Bildbereich) einfügen  |
| Strg-U    |           | Textbereich Unterstreichung an/aus  |
| Strg-V    |           | Inhalt der Zwischenablage einfügen  |
| Strg-W    |           | Dialog „Einheit einfügen“   |
| Strg-X    |           | Markierung ausschneiden und in Zwischenablage kopieren                      |
| Strg-Y    |           | Wiederholen   |
| Strg-X    |           | Rückgängig  |
| Strg-↑ -P |           | $\pi$ Pi  |
| Strg-↑ -T |           | Tabelle einfügen  |
| Strg-↑ -W |           | Erweiterten Textbereich (Writer) einfügen                                   |
| Strg-↑ -Z |           | $\infty$ Unendlich  |
| Pos1      |           | Anfang des Rechenblatts, Ausdrucks oder Textbereichs                        |
| End       |           | Ende des Rechenblatts, Ausdrucks oder Textbereichs                          |
| Strg-F4   |           | Rechenblatt schließen   |
| F8        |           | gleichzeitiges Editieren eines Namens an allen Stellen eines Formelbereichs |
| F9        |           | Rechenblatt neu berechnen   |
| F11       |           | Rechenblatt schrittweise im Debug-Modus ausführen                           |
| F12       |           | Gehe zur Definition des Namens, auf dem sich die Einfügemarke befindet      |
| Alt+ ESC  |           | Textbereich: Formel einfügen  |

| Taste   | Anweisung | Funktion           |
|---|-----------|--------------------|
| In der dynamischen Hilfe  |           |                    |
|    |           | Auswahl bestätigen |
| Strg -   | Leert.    | Auswahl bestätigen |
| Strg -   |           | Auswahl bestätigen |
|  -  |           | Auswahl bestätigen |

## D. Vordefinierte Maßeinheiten und Konstanten



Konstanten und Maßeinheiten werden in der Datei *Installationsverzeichnis/Entries/Units.xml* definiert. Man könnte dort eigene Einheiten und Sprachversionen definieren. Das ist allerdings nicht zu empfehlen, da diese Definitionen in anderen SMath-Installationen nicht bekannt sind. Für die Portabilität ist es besser, solche Definitionen im SMath-Rechenblatt selbst vorzunehmen.

### Zahlenwerte

```
1 % = 0,01
1 ‰ = 0,001
1 ppm = 1 · 10-6
1 ppb = 1 · 10-9
```

### Winkel

Der Anschaulichkeit halber werden die Ergebnisse hier in ° (Grad) angezeigt, standardmäßig würde die Anzeige im Bogenmaß erfolgen. Beachten Sie, dass die Symbole für Bogenminute und Bogensekunde Sonderzeichen sind.

|  |  |
|--|--|
| 1 rad = 57,2958 °  | Standardinheit: Radiant  |
| 1 ° = 1 °  | Grad   |
| 1 Gon = 0,9 °  | Neugrad, Gon $\frac{\pi}{2} \% = 1 \text{ Gon}$  |
| 1 U = 360 °  | Umdrehung $1 \text{ U} = 2 \cdot \pi \text{ rad}$  |
| 1 ' = $\frac{1}{60}$ °   | Bogenminute  |
| 1 " = $\frac{1}{3600}$ °   | Bogensekunde   |
| 1 ' = $\frac{1}{60}$ °   |  1' <b>Strg+K</b> '=  |
| 1 " = $\frac{1}{3600}$ °   |  1' <b>Strg+K</b> " = |
| $\alpha := 55 \text{ °} + 20 \text{ '} + 30 \text{ ''} = 55,342 \text{ °}$ |  |



## Länge

|                                   |  |                                  |
|-----------------------------------|--|----------------------------------|
| 1 m = 1 m                         |  | Standardeinheit: Meter           |
| 1 pm = $1 \cdot 10^{-12}$ m       |  | Picometer                        |
| 1 nm = $1 \cdot 10^{-9}$ m        |  | Nanometer                        |
| 1 $\mu$ m = $1 \cdot 10^{-6}$ m   |  | Mikrometer                       |
| 1 mm = 0,001 m                    |  | Millimeter                       |
| 1 cm = 0,01 m                     |  | Zentimeter                       |
| 1 dm = 0,1 m                      |  | Dezimeter                        |
| 1 km = 1000 m                     |  | Kilometer                        |
| 1 bohr = $5,292 \cdot 10^{-11}$ m |  | Bohrscher Radius                 |
| 1 Angstrom = $1 \cdot 10^{-10}$ m |  | Angström                         |
| 1 in = 0,0254 m                   |  | Zoll                             |
| 1 ft = 0,3048 m                   |  | Fuß (Zollsystem)                 |
| 1 yd = 0,9144 m                   |  | Yard                             |
| 1 Smoot = 1,702 m                 |  | Smoot (Körpergröße von O. Smoot) |
| 1 furlong = 201,2 m               |  | Furlong ("Furche")               |
| 1 mi = 1609 m                     |  | Meile                            |
| 1 AE = $1,5 \cdot 10^{11}$ m      |  | Astronomische Einheit            |
| 1 LJ = $9,46 \cdot 10^{15}$ m     |  | Lichtjahr                        |
| 1 pc = $3,09 \cdot 10^{16}$ m     |  | Parsec                           |
|                                   |  | a c = $9,46 \cdot 10^{15}$ m     |

## Fläche

|   |                   |                                  |
|---|-------------------|----------------------------------|
| 1 acre = $4046,8564 \text{ m}^2$        | Acre              | 1 acre = $0,1 \text{ furlong}^2$ |
| 1 barn = $1 \cdot 10^{-28} \text{ m}^2$ | Barn (Atomphysik) |                                  |
| 1 ha = $10000 \text{ m}^2$              | Hektar            |                                  |
| 1 ar = $100 \text{ m}^2$                | Ar                |                                  |

## Volumen

|                                      |            |
|--------------------------------------|------------|
| 1 l = $0,001 \text{ m}^3$            | Liter      |
| 1 ml = $1 \cdot 10^{-6} \text{ m}^3$ | Milliliter |
| 1 gal = $0,003785 \text{ m}^3$       | Gallone    |

## Zeit

|   |                          |
|---|--------------------------|
| $1 \text{ s} = 1 \text{ s}$                         | Standardeinheit: Sekunde |
| $1 \text{ a} = 3,1557 \cdot 10^7 \text{ s}$         | Jahr                     |
| $1 \text{ week} = 6,05 \cdot 10^5 \text{ s}$        | Woche                    |
| $1 \text{ d} = 86400 \text{ s}$                     | Tag                      |
| $1 \text{ hr} = 3600 \text{ s}$                     | Stunde                   |
| $1 \text{ min} = 60 \text{ s}$                      | Minute                   |
| $1 \text{ ks} = 1000 \text{ s}$                     | Kilosekunde              |
| $1 \text{ ms} = 0,001 \text{ s}$                    | Millisekunde             |
| $1 \text{ }\mu\text{s} = 1 \cdot 10^{-6} \text{ s}$ | Mikrosekunde             |
| $1 \text{ ns} = 1 \cdot 10^{-9} \text{ s}$          | Nanosekunde              |
| $1 \text{ ps} = 1 \cdot 10^{-12} \text{ s}$         | Picosekunde              |

## Winkelgeschwindigkeit, Kreisfrequenz

Die Standardeinheit für Drehzahl, Winkelgeschwindigkeit und Kreisfrequenz ist Radiant pro Sekunde. Eine Umdrehung entspricht  $2\pi$  Radiant.

|   |                      |
|---|----------------------|
| $1 \text{ rpm} = 1 \frac{\text{U}}{\text{min}}$     | Umdrehung pro Minute |
| $1 \text{ rph} = 1 \frac{\text{U}}{\text{hr}}$      | Umdrehung pro Stunde |
| $1 \text{ radpm} = 1 \frac{\text{rad}}{\text{min}}$ | Radiant pro Minute   |

## Frequenz

|  |           |
|--|-----------|
| $1 \text{ Hz} = 1 \frac{1}{\text{s}}$        | Hertz     |
| $1 \text{ kHz} = 1000 \text{ Hz}$            | kilohertz |
| $1 \text{ MHz} = 1 \cdot 10^6 \text{ Hz}$    | Megahertz |
| $1 \text{ GHz} = 1 \cdot 10^9 \text{ Hz}$    | Gigahertz |
| $1 \text{ THz} = 1 \cdot 10^{12} \text{ Hz}$ | Terahertz |

## Geschwindigkeit

|  |                                |
|--|--------------------------------|
| $1 \text{ Knoten} = 0,5144 \frac{\text{m}}{\text{s}}$      | Knoten                         |
| $1 \text{ kmh} = 0,2778 \frac{\text{m}}{\text{s}}$         | Kilometer pro Stunde (kmh)     |
| $1 \text{ mph} = 0,447 \frac{\text{m}}{\text{s}}$          | Meilen pro Stunde              |
| $1 \text{ c} = 2,998 \cdot 10^8 \frac{\text{m}}{\text{s}}$ | Lichtgeschwindigkeit im Vakuum |

## Masse

|   |   |
|---|---|
| $1 \text{ kg} = 1 \text{ kg}$                         | Standardeinheit: Kilogramm                            |
| $1 \text{ Mg} = 1000 \text{ kg}$                      | Megagramm (Tonne)                                     |
| $1 \text{ t} = 1000 \text{ kg}$                       | Tonne   |
| $1 \text{ ton} = 907,1847 \text{ kg}$                 | Tonne (Zollsystem) $1 \text{ ton} = 2000 \text{ lb}$  |
| $1 \text{ g} = 0,001 \text{ kg}$                      | Gramm (die Erdbeschleunigung ist $g_e$ )              |
| $1 \text{ mg} = 1 \cdot 10^{-6} \text{ kg}$           | Milligramm  |
| $1 \text{ } \mu\text{g} = 1 \cdot 10^{-9} \text{ kg}$ | Mikrogramm  |
| $1 \text{ ct} = 0,0002 \text{ kg}$                    | Metrisches Karat                                      |
| $1 \text{ oz} = 0,0283 \text{ kg}$                    | Unze (Zollsystem)                                     |
| $1 \text{ lb} = 0,4536 \text{ kg}$                    | Pfund (Zollsystem) $1 \text{ lb} = 16 \text{ oz}$     |
| $1 \text{ slug} = 14,5939 \text{ kg}$                 | Slug (kohärente Masseneinheit im Zollsystem)          |
|   | $1 \text{ lbf} = 1 \frac{\text{slug ft}}{\text{s}^2}$ |

## Kraft

|  |                         |
|--|-------------------------|
| $1 \text{ N} = 1 \text{ N}$                          | Standardeinheit: Newton |
| $1 \text{ TN} = 1 \cdot 10^{12} \text{ N}$           | Teranewton              |
| $1 \text{ GN} = 1 \cdot 10^9 \text{ N}$              | Giganewton              |
| $1 \text{ MN} = 1 \cdot 10^6 \text{ N}$              | Meganewton              |
| $1 \text{ kN} = 1000 \text{ N}$                      | Kilonewton              |
| $1 \text{ mN} = 0,001 \text{ N}$                     | Millinewton             |
| $1 \text{ } \mu\text{N} = 1 \cdot 10^{-6} \text{ N}$ | Mikronewton             |

|                             |                                  |
|-----------------------------|----------------------------------|
| 1 dyn = $1 \cdot 10^{-5}$ N | Dyn                              |
| 1 lbf = 4,448 N             | Pfund Kraft (Zollsystem)         |
| 1 gf = 1 gf                 | Gramm Kraft (Pond)               |
| 1 kp = 9,807 N              | Kilogramm Kraft (Kilopond)       |
| 1 Mp = 9807 N               | Tonne Kraft (Megapond)           |
| 1 tonf = 2000 lbf           | Tonne Kraft (Zollsystem)         |
| 1 kip = 1000 lbf            | Tausend Pfund Kraft (Zollsystem) |

## Druck

|                                |  |
|--------------------------------|--|
| 1 Pa = 1 Pa                    | Standardeinheit: Pascal                  |
| 1 GPa = $1 \cdot 10^9$ Pa      | Gigapascal                               |
| 1 MPa = $1 \cdot 10^6$ Pa      | Megapascal                               |
| 1 kPa = 1000 Pa                | Kilopascal                               |
| 1 atm = $1,013 \cdot 10^5$ Pa  | Atmosphäre                               |
| 1 bar = $1 \cdot 10^5$ Pa      | Bar                                      |
| 1 Torr = 133,3 Pa              | Torr (mm Quecksilbersäule)               |
| 1 mmHg = 133,3 Pa              | mm Quecksilbersäule (Torr)               |
| 1 inHg = 3390 Pa               | Zoll Quecksilbersäule                    |
| 1 psi = 6894,7573 Pa           | Pfund Kraft pro Quadratzoll (Zollsystem) |
| 1 psf = 47,8803 Pa             | Pfund Kraft pro Quadratfuß (Zollsystem)  |
| 1 ksi = $6,8948 \cdot 10^6$ Pa | Tausend Pfund Kraft pro Quadratzoll      |
| 1 ksf = 47880,259 Pa           | Tausend Pfund Kraft pro Quadratfuß       |

## Energie

|                             |                        |
|-----------------------------|------------------------|
| 1 J = 1 J                   | Standardeinheit: Joule |
| 1 TJ = $1 \cdot 10^{12}$ J  | Terajoule              |
| 1 GJ = $1 \cdot 10^9$ J     | Gigajoule              |
| 1 MJ = $1 \cdot 10^6$ J     | Megajoule              |
| 1 kJ = 1000 J               | Kilojoule              |
| 1 mJ = 0,001 J              | Millijoule             |
| 1 nJ = $1 \cdot 10^{-9}$ J  | Nanojoule              |
| 1 pJ = $1 \cdot 10^{-12}$ J | Picojoule              |

|                             |  |                      |
|-----------------------------|--|----------------------|
| 1 cal = 4,187 J             |  | Kalorie              |
| 1 kcal = 4187 J             |  | Kilokalorie          |
| 1 BTU = 1055 J              |  | British Thermal Unit |
| 1 erg = $1 \cdot 10^{-7}$ J |  | Erg                  |

### Leistung

|                                 |  |   |
|---------------------------------|--|---|
| 1 W = 1 W                       |  | Standardeinheit: Watt                           |
| 1 GW = $1 \cdot 10^9$ W         |  | Gigawatt  |
| 1 MW = $1 \cdot 10^6$ W         |  | Megawatt  |
| 1 kW = 1000 W                   |  | Kilowatt  |
| 1 mW = 0,001 W                  |  | Milliwatt                                       |
| 1 $\mu$ W = $1 \cdot 10^{-6}$ W |  | Mikrowatt                                       |
| 1 nW = $1 \cdot 10^{-9}$ W      |  | Nanowatt  |
| 1 pW = $1 \cdot 10^{-12}$ W     |  | Picowatt  |
| 1 hp = 745,7 W                  |  | Britische Pferdestärke (Zollsystem)             |
| 1 hhp = 746,0 W                 |  | Britische Pferdestärke (hydraulisch, inkorrekt) |

### Stromstärke

|                                 |  |                         |
|---------------------------------|--|-------------------------|
| 1 A = 1 A                       |  | Standardeinheit: Ampere |
| 1 kA = 1000 A                   |  | Kiloampere              |
| 1 mA = 0,001 A                  |  | Milliampere             |
| 1 $\mu$ A = $1 \cdot 10^{-6}$ A |  | Mikroampere             |
| 1 nA = $1 \cdot 10^{-9}$ A      |  | Nanoampere              |
| 1 pA = $1 \cdot 10^{-12}$ A     |  | Picoampere              |

### Spannung

|                                 |  |                       |
|---------------------------------|--|-----------------------|
| 1 V = 1 V                       |  | Standardeinheit: Volt |
| 1 kV = 1000 V                   |  | Kilovolt              |
| 1 mV = 0,001 V                  |  | Millivolt             |
| 1 $\mu$ V = $1 \cdot 10^{-6}$ V |  | Mikrovolt             |
| 1 nV = $1 \cdot 10^{-9}$ V      |  | Nanovolt              |
| 1 pV = $1 \cdot 10^{-12}$ V     |  | Picovolt              |

### Kapazität

|   |                        |
|---|------------------------|
| $1 \text{ F} = 1 \text{ F}$                         | Standardeinheit: Farad |
| $1 \text{ mF} = 0,001 \text{ F}$                    | Millifarad             |
| $1 \text{ }\mu\text{F} = 1 \cdot 10^{-6} \text{ F}$ | Mikrofarad             |
| $1 \text{ nF} = 1 \cdot 10^{-9} \text{ F}$          | Nanofarad              |
| $1 \text{ pF} = 1 \cdot 10^{-12} \text{ F}$         | Picofarad              |

### Ladung

|   |                          |
|---|--------------------------|
| $1 \text{ C} = 1 \text{ C}$                         | Standardeinheit: Coulomb |
| $1 \text{ mC} = 0,001 \text{ C}$                    | Millicoulomb             |
| $1 \text{ }\mu\text{C} = 1 \cdot 10^{-6} \text{ C}$ | Mikrocoulomb             |
| $1 \text{ nC} = 1 \cdot 10^{-9} \text{ C}$          | Nanocoulomb              |
| $1 \text{ pC} = 1 \cdot 10^{-12} \text{ C}$         | Picocoulomb              |
| $1 \text{ e} = 1,602 \cdot 10^{-19} \text{ C}$      | Elementarladung          |

### Induktivität

|   |                        |
|---|------------------------|
| $1 \text{ H} = 1 \text{ H}$                         | Standardeinheit: Henry |
| $1 \text{ mH} = 0,001 \text{ H}$                    | Millihenry             |
| $1 \text{ }\mu\text{H} = 1 \cdot 10^{-6} \text{ H}$ | Mikrohenry             |

### Magnetische Feldstärke

|                                 |         |
|---------------------------------|---------|
| $Oe = 1 \frac{\text{G}}{\mu_0}$ | Oersted |
|---------------------------------|---------|

### Magnetischer Fluss

|   |                        |                                       |
|---|------------------------|---------------------------------------|
| $1 \text{ Wb} = 1 \text{ Wb}$               | Standardeinheit: Weber | $\text{Wb} = 1 \text{ m}^2 \text{ T}$ |
| $1 \text{ Mx} = 1 \cdot 10^{-8} \text{ Wb}$ | Maxwell                |                                       |

### Magnetische Flussdichte

|                                  |                        |
|----------------------------------|------------------------|
| $1 \text{ T} = 1 \text{ T}$      | Standardeinheit: Tesla |
| $1 \text{ G} = 0,0001 \text{ T}$ | Gauss                  |

### Widerstand

|                                   |                      |
|-----------------------------------|----------------------|
| $1 \Omega = 1 \Omega$             | Standardeinheit: Ohm |
| $1 M\Omega = 1 \cdot 10^6 \Omega$ | Megaohm              |
| $1 k\Omega = 1000 \Omega$         | Kilohm               |
| $1 m\Omega = 0,001 \Omega$        | Milliohm             |

### Elektrische Leitfähigkeit

|                               |         |
|-------------------------------|---------|
| $1 S = 1 S$                   | Siemens |
| $1 dS = 0,1 S$                |         |
| $1 mS = 0,001 S$              |         |
| $1 \mu S = 1 \cdot 10^{-6} S$ |         |
| $1 pS = 1 \cdot 10^{-12} S$   |         |

### Elektrisches Dipolmoment

|   |       |
|---|-------|
| $D = 3,3356 \cdot 10^{-30} \text{ m C}$ | Debye |
|---|-------|

### Strahlendosis

|               |                       |
|---------------|-----------------------|
| $1 Gy = 1 Gy$ | Standardeinheit: Gray |
| $1 Sv = 1 Gy$ | Sievert               |

### Lichtstärke, Lichtstrom

|                  |                          |
|------------------|--------------------------|
| $1 cd = 1 cd$    | Standardeinheit: Candela |
| $1 lm = 1 sr cd$ | Lumen                    |

### Beleuchtungsstärke

|               |                      |                           |
|---------------|----------------------|---------------------------|
| $1 lx = 1 lx$ | Standardeinheit: Lux | $1 lx = 1 \frac{lm}{m^2}$ |
|---------------|----------------------|---------------------------|

### Stoffmenge

|                                   |                      |
|-----------------------------------|----------------------|
| $1 mol = 1 mol$                   | Standardeinheit: Mol |
| $1 kmol = 1000 mol$               | Kilomol              |
| $1 mmol = 0,001 mol$              | Millimol             |
| $1 \mu mol = 1 \cdot 10^{-6} mol$ | Mikromol             |

### Lineare Dichte

(Maß für die Fadenfeinheit bei Textilien)

$$1 \text{ den} = 1,11 \cdot 10^{-7} \text{ kg} \frac{1}{\text{m}} \quad \text{Denier} \quad \frac{1 \text{ g}}{9000 \text{ m}} = 1 \text{ den}$$

### Viskosität

Dynamische Viskosität (Maßeinheit Poise) und kinematische Viskosität (Maßeinheit Stokes):

$$1 \text{ P} = 0,1 \text{ s Pa} \quad \text{Poise} \quad 1 \text{ P} = 0,1 \frac{\text{kg}}{\text{m s}}$$

$$1 \text{ St} = 1 \frac{\text{cm}^2}{\text{s}} \quad \text{Stokes}$$

### Katalytische Aktivität

$$1 \text{ kat} = 1 \text{ kat} \quad \text{Katal} \quad 1 \text{ kat} = 1 \frac{\text{mol}}{\text{s}}$$

### Auflösung

$$1 \text{ dpi} = 39,3701 \cdot \frac{1}{\text{m}} \quad \text{Punkte pro Zoll}$$

### Information

$$1 \text{ bit} = 1 \text{ bit} \quad \text{Standardeinheit: Bit}$$

$$1 \text{ B} = 8 \text{ bit} \quad \text{Byte}$$

$$1 \text{ TB} = 8 \cdot 10^{12} \text{ bit} \quad \text{Terabyte}$$

$$1 \text{ GB} = 8 \cdot 10^9 \text{ bit} \quad \text{Gigabyte}$$

$$1 \text{ MB} = 8 \cdot 10^6 \text{ bit} \quad \text{Megabyte}$$

$$1 \text{ kB} = 8000 \text{ bit} \quad \text{Kilobyte}$$

$$1 \text{ TiB} = 8,7961 \cdot 10^{12} \text{ bit} \quad \text{Tebibyte}$$

$$1 \text{ GiB} = 8,5899 \cdot 10^9 \text{ bit} \quad \text{Gibibyte}$$

$$1 \text{ MiB} = 8,3886 \cdot 10^6 \text{ bit} \quad \text{Mebibyte}$$

$$1 \text{ kiB} = 8192 \text{ bit} \quad \text{Kibibyte}$$



## Temperatur

Absoluttemperaturskalen und Differenzeinheiten:

|   |  |
|---|--|
| $1 \text{ K} = 1 \text{ K}$                         | Kelvin                                     |
| $1 \text{ }^\circ\text{Ra} = 0,5556 \text{ K}$      | Grad Rankine                               |
| $1 \text{ }^\circ\Delta\text{C} = 1 \text{ K}$      | $\Delta T$ von $1 \text{ }^\circ\text{C}$  |
| $1 \text{ }^\circ\Delta\text{F} = 0,5556 \text{ K}$ | $\Delta T$ von $1 \text{ }^\circ\text{F}$  |
| $1 \text{ }^\circ\Delta\text{Re} = 1,25 \text{ K}$  | $\Delta T$ von $1 \text{ }^\circ\text{Re}$ |

Temperaturskalen mit verschobenem Nullpunkt, diese dürfen nicht direkt für Temperaturdifferenzen verwendet werden. Verwenden Sie statt dessen absolute und Differenzeinheiten:

|  |                 |
|--|-----------------|
| $0 \text{ }^\circ\text{C} = 273,15 \text{ K}$  | Grad Celsius    |
| $0 \text{ }^\circ\text{F} = 255,37 \text{ K}$  | Grad Fahrenheit |
| $0 \text{ }^\circ\text{Re} = 273,15 \text{ K}$ | Grad Reamur     |

Temperaturdifferenzen von einem Grad auf verschiedenen Skalen:

|   |
|---|
| $1 \text{ }^\circ\text{C} - 0 \text{ }^\circ\text{C} = 1 \text{ K}$                       |
| $1 \text{ }^\circ\text{C} - 0 \text{ }^\circ\text{C} = 1 \text{ }^\circ\Delta\text{C}$    |
| $1 \text{ }^\circ\text{F} - 0 \text{ }^\circ\text{F} = 0,5556 \text{ K}$                  |
| $1 \text{ }^\circ\text{F} - 0 \text{ }^\circ\text{F} = 1 \text{ }^\circ\text{Ra}$         |
| $1 \text{ }^\circ\text{F} - 0 \text{ }^\circ\text{F} = 1 \text{ }^\circ\Delta\text{F}$    |
| $1 \text{ }^\circ\text{Re} - 0 \text{ }^\circ\text{Re} = 1 \text{ }^\circ\Delta\text{Re}$ |

## Konstanten

|  |                                |
|--|--------------------------------|
| $c = 2,9979 \cdot 10^8 \frac{\text{m}}{\text{s}}$                      | Lichtgeschwindigkeit im Vakuum |
| $g_e = 9,8066 \frac{\text{m}}{\text{s}^2}$                             | Erdbeschleunigung              |
| $G_N = 6,6743 \cdot 10^{-11} \frac{\text{m}^2}{\text{kg}^2} \text{ N}$ | Gravitationskonstante          |
| $m_p = 1,673 \cdot 10^{-27} \text{ kg}$                                | Protonenmasse                  |
| $m_n = 1,675 \cdot 10^{-27} \text{ kg}$                                | Neutronenmasse                 |
| $u = 1,66 \cdot 10^{-27} \text{ kg}$                                   | Atomare Masseneinheit          |
| $m_e = 9,109 \cdot 10^{-31} \text{ kg}$                                | Elektronenmasse                |
| $e = 1,602 \cdot 10^{-19} \text{ C}$                                   | Elementarladung                |


---

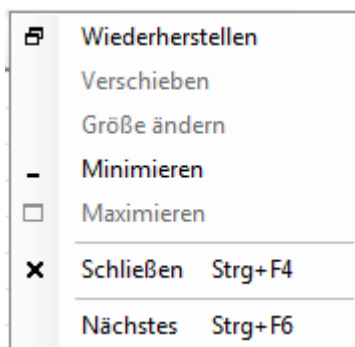
ANHANG D. VORDEFINIERTE MASSEINHEITEN UND KONSTANTEN

---

|  |  |
|--|--|
| $h = 6,6261 \cdot 10^{-34} \frac{\text{J}}{\text{s}}$      | Plancksches Wirkungsquantum                          |
| $k = 1,3807 \cdot 10^{-23} \frac{\text{J}}{\text{K}}$      | Boltzmann-Konstante                                  |
| $\mu_0 = 1,2566 \cdot 10^{-6} \frac{\text{m T}}{\text{A}}$ | Magnetische Feldkonstante, Permeabilität des Vakuums |
| $R_m = 8,3145 \frac{\text{J}}{\text{mol K}}$               | Gaskonstante   |
| $N_A = 6,0221 \cdot 10^{23} \frac{1}{\text{mol}}$          | Avogadro-Zahl (Teilchenzahl pro mol)                 |

## E. Menüstruktur

**Blatt-Menü**  Funktionen des Fenstermanagers für das aktuelle Rechenblatt, falls dieses maximiert ist. Ansonsten sind diese Funktionen durch die Titelzeile der Einzelblätter zu erreichen.

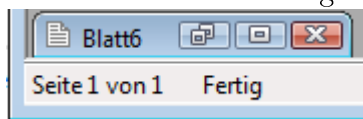


**Wiederherstellen** Alle geöffneten Blätter in separaten Fenstern anzeigen.



**Verschieben**





**Größe ändern**

**Minimieren** Das aktuell angezeigte Blatt minimieren







**Maximieren** Rechenblatt maximieren

**Schließen**  -  Blatt schließen



**Nächstes**  -  Umschalten auf das nächste offene Blatt (alternativ geht auch  -  )

### Datei

**Blatt erstellen**  -  Neues Blatt öffnen

**Öffnen**  -  Vorhandenes Blatt öffnen. Alternativ kann man auch eine SMath Datei (.sm) auf das SMath-Fenster ziehen.

**Blatt schließen**  -  aktuelles Blatt schließen

**Speichern**  -  aktuelles Blatt speichern. Damit wird die Befehlshistorie gelöscht, Rückgängig machen geht nur bis zum letzten Speicherpunkt.

**Speichern unter** Aktuelles Blatt unter wählbarem Namen und in wählbarem Format speichern. Hier verbergen sich also auch diverse Export-Funktionen, siehe dazu Abschnitt ??.

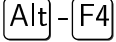

**Blatteinstellungen** Einstellungen zum Druckbild (siehe Abschnitt ??)

**Drucken**  

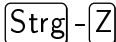

**Druckvorschau** Vorschau des Druckbildes.

**Eigenschaften** Einstellungen zu den Dateieigenschaften (Metadaten, siehe Abschnitt ??)

**Zuletzt geöffnete Dateien** Liste der zuletzt geöffneten Dateien.

**Schließen**   Alle offenen Blätter und das Programm schließen. Gegebenenfalls wird Speichern angeboten.

**Bearbeiten** Dieses Menü enthält die Standardfunktionen zum Blatt-Bearbeiten. Ist ein Bereich komplett (hellblau) markiert, dann erscheinen die Optionen des Kontextmenüs auch hier im Hauptmenü. Auf diese Optionen wird bei der Beschreibung der Bereiche eingegangen.

**Rückgängig**   Macht vorangegangene Aktionen rückgängig. Mitunter gilt das nur für die Modifikationen am aktuell geänderten Bereich (Formel). Maximal möglich bis zum letzten Speicherpunkt.

**Wiederherstellen**   Rückgängig rückgängig machen.

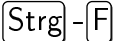

**Ausschneiden**  

**Kopieren**  

**Einfügen**  

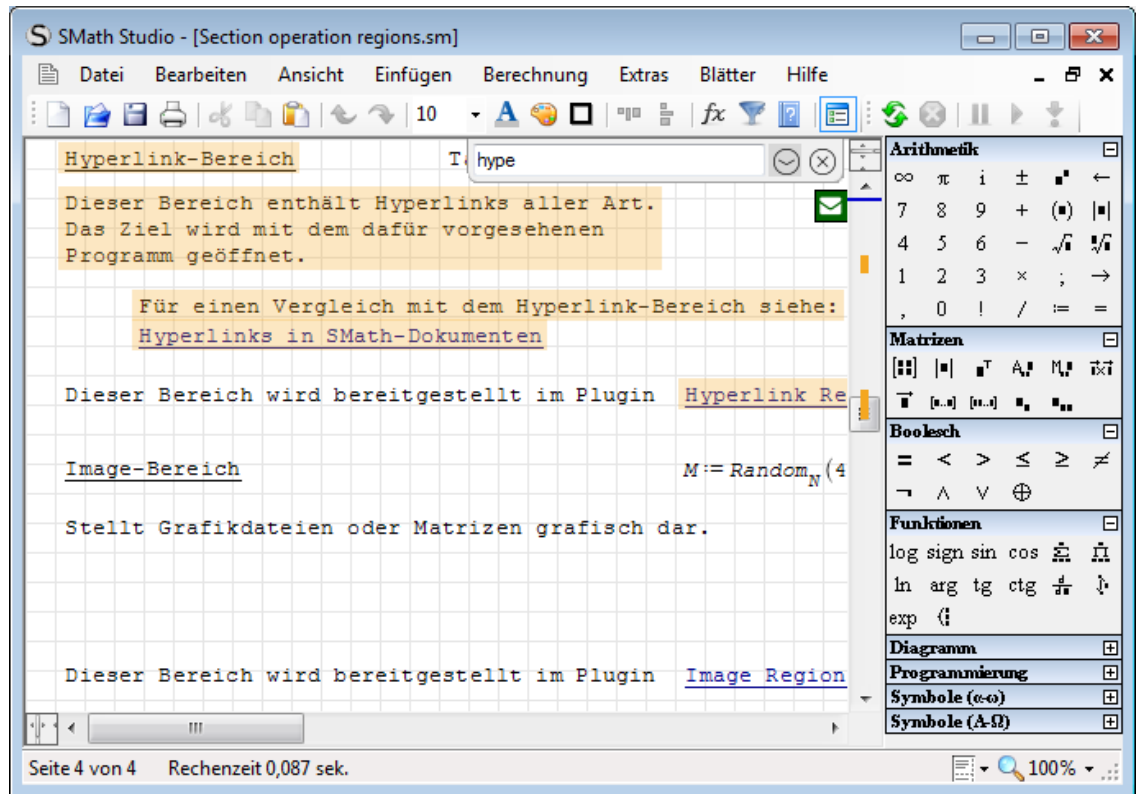
**Löschen** 

**Alles markieren**  

**Suchen**   Gesucht werden kann in allen Bereichen des Rechenblatts. Eine Funktion zum Ersetzen fehlt allerdings.

Es erscheint ein Textfeld für die Eingabe des Suchbegriffs. Bereiche, in denen der Begriff gefunden wird, werden orange gekennzeichnet und können auf der senkrechten Lauffleiste direkt angesprochen werden.

Mit der Taste  geht man zur nächsten Fundstelle, diese wird mit der Oberkante des Fensters ausgerichtet.



## Aktualisieren

**Ansicht** In diesem Menü können Anzeigeelemente ein- und ausgeschaltet werden.

**Raster** Schalter für das Kästchenraster auf dem Blatthintergrund

**Ausgabefenster** Schalter für das Ausgabefenster der Protokoll-Funktion `trace()`

**Debug-Fenster** Schalter für das Ausgabefenster des Debuggers (siehe Abschnitt 10.3). Debugger und Protokollfunktion teilen sich ein Fenster mit zwei Reitern unterhalb des Rechenblatts.

**Dynamische Hilfe** Schalter für die Anzeige der dynamischen Hilfe. Das ist einerseits eine dynamisch erzeugte Liste aller bekannten Bezeichner, die zum gerade eingetippten Wort passen, andererseits die Anzeige des gespeicherten Ausdrucks für Definitionen in Formelbereichen, wenn die Maus darüber bewegt wird. Die Abschaltung kann bei Demonstrationen am Beamer hilfreich sein, wenn man Dinge mit der Maus zeigen will, ohne dass die dynamische Hilfe aufpoppt.

**Immer im Vordergrund** Das SMATH-Fenster bleibt im Vordergrund auch wenn andere Programme aktiv sind.

**Einfügen** Hier können SMATH-Objekte in das Rechenblatt eingefügt werden. Der Inhalt der Liste ist abhängig von den installierten Anwendungserweiterungen (Plug-ins).

**Matrix**  $\text{Strg} - \text{M}$  Aufruf des Dialogs „Matrix einfügen“. Das Arbeiten mit Matrizen wird in Abschnitt 6.8 erläutert.

**Funktion**  $\text{Strg} - \text{E}$  Aufruf des Dialogs „Einfügen - Funktion“

**Operator** Aufruf des Dialogs „Einfügen - Operator“ zum Einfügen arithmetischer und boolescher Operatoren

**Einheit**  $\boxed{\text{Strg}}-\boxed{\text{W}}$  Aufruf des Dialogs „Einheit einfügen“

**Hintergrund** Aufruf des Dialogs „Hintergrund“. Damit kann man ein Hintergrundbild laden, z.B. für Formblätter.

**Bild** Bild erstellen oder aus Datei einfügen. Dient Illustrationszwecken.

**Bild erstellen**  $\boxed{\text{Strg}}-\boxed{\text{T}}$  Fügt einen Bereich ein, in dem mit der Maus freihand gezeichnet werden kann. Löschen/Radieren ist nicht möglich, daher hat diese Funktion wenig Wert. Verwenden Sie statt dessen den Image-Bereich (Abschnitt ??).

**aus Datei** Rasterbilder in verschiedenen Formaten (PNG, JPG, Icons,...) können aus der Zwischenablage oder durch Ziehen mit der Maus eingefügt werden. Der Inhalt ist statisch, wird also nicht aktualisiert, wenn der Dateiinhalt sich ändert.

**Controls** Interaktive Bedienelemente

**RadioButtonList** (Plug-in *RadioButtonList Region*) Anordnung mehrerer gegenseitig ausschließender (exklusiver) An/Aus-Schaltflächen

**CheckBoxList** (Plug-in *CheckboxList Region*) Anordnung mehrerer (auch gleichzeitig wählbarer) An/Aus-Schaltflächen

**ComboBoxList** (Plug-in *ComboBoxList Region*) Auswahlliste, der Wert einer Variable kann aus einer Liste von Werten ausgewählt werden.

**Slider** (Plug-in *Slider Region*) Schieberegler zur interaktiven Einstellung von Werten

**Numeric up-down** (Plugin *Numeric UpDown Region*) Komfortables Eingabefeld für Zahlenwerte, die Werte können mit der Maus oder den Pfeiltasten geändert werden.

**Development Tools** (Plug-in *Development Tools*) Werkzeuge für Entwickler

**SMath file recovery** Erste Hilfe, wenn sich ein Rechenblatt nicht mehr einlesen lässt (kann in seltenen Fällen vorkommen).

**Math $\Rightarrow$ Terms** Übersetzt SMath-Ausdrücke in C#-Code für Plug-ins.

**Math's visual debug** Übersetzt SMath-Ausdrücke in einen Strukturbaum.

**Network analyzer** Zeigt Verbindungsdetails für Internet-Adressen an (z.B. ob Cookies gesetzt werden).

**System informations** Zeigt SMath-relevante Systeminformationen an. Diese Info kann in die Zwischenablage übernommen werden und sollte bei Fehlermeldungen an die Entwickler immer mit angefügt werden.

**Maxima** (Plugin *Maxima*) Steuerung der Maxima-Schnittstelle

**Log** Fenster zur Sichtung des Sitzungs- und Übersetzungsprotokolls.

**Debug** Fenster zur interaktiven Sichtung und Steuerung des Übersetzungsprozesses.

**Settings** Hier wird der Pfad zu Maxima angegeben.

**Draw2D()** Maxima-2D-Diagrammbereich.

**Draw3D()** Maxima-3D-Diagrammbereich.

**Plot** Spezialdiagramme

**Pie / doughnut** (Plugin *PieChartRegion*) Torten-/Ringdiagramm mit umfangreichen Formatfunktionen. Eingabewert ist ein Vektor, die Formateinstellungen werden per Doppelklick geöffnet.

**Textbereich (utilities)** (Plugin *Text Region Utilities*) Komfortfunktionen für Standard-Textbereiche.

**add a new text choosing font** Einfügen eines Textbereichs mit Auswahl der Schriftart.

**add a new link to a worksheet** Einfügen eines Textbereichs mit Link zu einem anderen Rechenblatt.



**insert a toolbox** Einfügen einer Werkzeugbox mit Funktionen zum Editieren von Schriftart und Links, zur Umwandlung von Standardtext in erweiterte Textbereiche, und zum Suchen/Ersetzen.

**Diagramm** Diagramm erstellen

**Zweidimensional** (Tastatur @) Fügt einen 2D-Diagrammbereich ein.

**Dreidimensional** Fügt einen 3D-Diagrammbereich ein.

**Blattbereich** Fügt zwei Begrenzungslinien ein, zwischen denen das Rechenblatt ausgeblendet werden kann (Blattbereich).

**Trennlinie**  -  Fügt eine horizontale Trennlinie ein. Diese erzeugt im Ausdruck einen Seitenumbruch und in exportierten Anwendungen (siehe Kapitel 11) einen neuen Reiter (*tab*).

**Checkbox** (Plug-in *Checkbox Region*) Schaltfläche, die eine Variable durch Mausklick zwischen Null und Eins umschaltet.



**Combobox** (Plug-in *Combobox Region*) Auswahlliste, der Wert einer Variable kann aus einer Liste von Werten ausgewählt werden.

**Formatted label** (Plug-in *CFLabel Region*) Formatierte Textausgabe. Text, Textfarbe und Hintergrundfarbe können durch Variablenwerte gesetzt werden.

**Hyperlink**  -  (Plug-in *Hyperlink Region*) Fügt einen Hyperlink-Bereich ein (siehe Seite ??).

**Image** (Plug-in *Image Region*) Fügt einen Bildbereich zur Anzeige von Bilddateien oder Datenmatrizen ein (siehe Abschnitt ??). Alternativ kann das Bild mit dem Standardprogramm für Bilddateien erzeugt oder editiert werden. Das Dateiformat kann unter einer Vielzahl von Pixel- oder Vektorgrafikformaten gewählt werden.

**Modeller** (Plug-in *Modeller Region*) Bereich für interaktive parametrische Grafik, siehe Abschnitt 8.4.

**Snapshot**  -  (Plug-in *Snapshot Region*) Bereich für den automatisierten Export von Ausschnitten des Rechenblatts als Bilddatei (beispielsweise für die Erstellung von Bildern für Handbücher).

**Table** (Plug-in *Table Region*) Tabelle zur Anzeige von Daten.

**Writer region** (Plug-in *Writer Region*) Erweiterter Textbereich mit umfangreichen Formatierungsmöglichkeiten.

**Textbereich** (Tastatur ") Startet einen Textbereich.

**X-Y Plot** Erweiterter 2D-Diagrammbereich, siehe Abschnitt 8.3)

**Bereichs-Sperre** Damit kann ein ausgeblendeter Blattbereich mit einem Passwort gegen Einblenden geschützt werden.

**Sperren** Ruft den Dialog „Erstelle Passwort“ auf.

**Entsperren** Fordert zur Passworteingabe zum dauerhaften Entsperren des Gebiets auf.

## Berechnung

**Lösen** Löst eine Gleichung numerisch nach einer Variable auf. Die Variable muss vorher markiert werden.

$$x^2 = 3$$

$$\begin{bmatrix} -1,73205097195524 \\ 1,73204983012458 \end{bmatrix}$$

**Berechnen** Wertet einen markierten (Teil-)Ausdruck aus.

$$\sqrt{3} + 2$$

$$\frac{21650635094611}{12500000000000}$$

**Vereinfachen** Wertet einen markierten (Teil-)Ausdruck symbolisch aus.

$$\sqrt{3} + 2 \cdot \sqrt{3}$$

$$\frac{3}{2}$$

$$\frac{3}{3}$$

**Kehrwert** Berechnet den Kehrwert des markierten (Teil-)Ausdrucks symbolisch

$$\sqrt{3} + 2 \cdot \sqrt{3}$$

$$\frac{4,54747350886464 \cdot 10^{15}}{2,36293654902813 \cdot 10^{16}}$$

$$\frac{1}{\sqrt{3} + a \cdot \sqrt{3}}$$

$$\frac{1}{3 \cdot \frac{1}{2} \cdot (1+a)}$$

**Differenzieren** Leitet den Ausdruck nach der markierten Variable ab

$$\text{tg}(x)$$

$$\frac{1}{\cos(x)^2}$$

**Determinante** Berechnet die Determinante einer markierten Matrix

$$\begin{bmatrix} 1 & 2 \\ a & 4 \end{bmatrix}$$

$$2 \cdot 2 - a$$

**Automatische Berechnung** Schalter für die automatische Berechnung nach jeder Änderung. Abschalten kann sinnvoll sein, wenn die Rechenzeit zu lang ist und mehrere Änderungen gemacht werden müssen, bis eine neue Rechnung überhaupt sinnvoll ist.

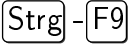



**Im Fehlerfall** Verhalten bei Auftreten eines Fehlers

**Fortsetzen** Die Rechnung wird weitergeführt, es können natürlich Folgefehler auftreten.

**Pause** Die Rechnung wird angehalten und geht in den Debug-Modus (schrittweise Ausführung mit Anzeige des Kontexts)

**Abbrechen** Die Rechnung wird angehalten

**Blatt berechnen**  Manueller Start der Neuberechnung, wenn die automatische Berechnung abgeschaltet ist. Das geht alternativ über Werkzeugleiste > .

**Prozess unterbrechen** Rechnung anhalten (wenn sie zu lange dauert).

### Extras

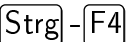
**Plug-ins** Öffnet die Seite „Plug-ins“ der Erweiterungs-Verwaltung, dort können Plug-ins (Anwendungserweiterungen) aus dem Internet geladen oder aktualisiert werden.

**Code-Baustein-Verwaltung** Öffnet die Seite „Code-Bausteine“ (*snippets*) der Erweiterungs-Verwaltung, dort können Code-Bausteine aus dem Internet geladen oder vorhandene Bausteine ins aktuelle Rechenblatt eingefügt werden.

**Einstellungen** Öffnet den Dialog „Einstellungen“ für Berechnung und Benutzeroberfläche, siehe Abschnitt ??.


**Blätter** Bietet Zugriff auf die geöffneten Blätter und darüber hinaus folgende Funktionen:

**Neues Blatt**  Neues Rechenblatt erstellen.

**Blatt schließen**  Aktuelles Blatt schließen.

### Hilfe

**Referenzmaterial** Öffnet das interaktive Handbuch. Voreingestellt ist eine kleine mathematische Formelsammlung, alternativ können andere interaktive Handbücher in der Erweiterungs-Verwaltung eingestellt werden (Abschnitt 4.3). In der THB-Distribution ist hier bereits das interaktive Handbuch des Autors dieses Buchs vorkonfiguriert.

Der bequemste Zugriff erfolgt über **Werkzeugleiste** > .

**Beispiele** Öffnet die Seite „Beispiele“ der Erweiterungs-Verwaltung. Darin kann man Beispiele aus dem Internet laden oder installierte Beispiele auswählen und als Rechenblatt öffnen.

**Nach Aktualisierungen suchen** Öffnet die Seite „SMath Studio“ der Erweiterungsverwaltung und bietet eventuell vorhandene neuere Programmversionen zur Installation an.

**Spenden** Sie erhalten die Kontodaten des Entwicklers Andrey Ivashov. Unterstützt werden verschiedene Online-Bezahlungsmöglichkeiten.

**SMath Studio Online-Version im Browser** Öffnet eine Sitzung in der Web-Version von SMath. Diese hat gegenüber der Programmversion einen eingeschränkten Funktionsumfang.

**SMath Info** Versionsinformation wird angezeigt.

## F. Portables SMath mit Maxima

Hier wird beschrieben, wie eine portable Distribution von SMath Studio zusammengestellt werden kann. Diese lässt sich auf einem USB-Speicher installieren und an jedem Windows-PC benutzen. Maxima wird dabei als Unterverzeichnis in das SMath-Installationsverzeichnis plaziert. Eine Zeit lang war dies die Konfiguration der portablen THB-Distribution. Gegenwärtig (März 2019) ist aber nur die SMath-Installation portabel und Maxima selbst ist nicht mehr Bestandteil der Distribution.

### F.1. Portables SMath Studio

Normalerweise speichert SMath Erweiterungen aus der Online-Galerie und die Programm-einstellungen in einem Windows-Systemverzeichnis. Man kann dieses mit der Funktion `SettingsDirectory()` ermitteln. Bei einer Standardinstallation kann man beispielsweise folgende Ausgabe bekommen:

```
SettingsDirectory(■) = "C:\Users\Kraska\AppData\Roaming\SMath\ "
```

Stellt man in das Installationsverzeichnis von SMath Studio eine leere Datei des Namens *portable.version* ein, so ist das Installationsverzeichnis gleichzeitig das Einstellungsverzeichnis (*settings directory*).

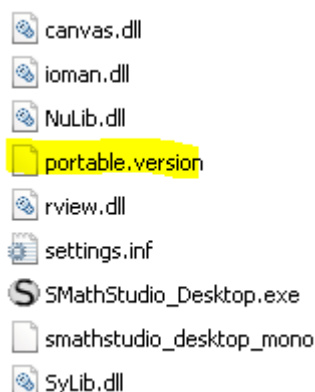
Eine portable SMath-Version kann man also auf folgende Weise erzeugen:

1. Herunterladen von der offiziellen SMath-Studio-Seite. Die aktuelle offizielle Version findet man unter **SMath Studio Forum** > **Download SMath Studio**. Dort lohnt eventuell ein Blick in das Unterforum **Nightly builds**, welches Entwicklungsversionen enthält, die häufig für die praktische Arbeit stabil genug sind und die meist auch von den aktuellen Plug-ins unterstützt werden.

Sie benötigen die Linux-Version (ja, wirklich, auch wenn Sie unter Windows arbeiten wollen!):



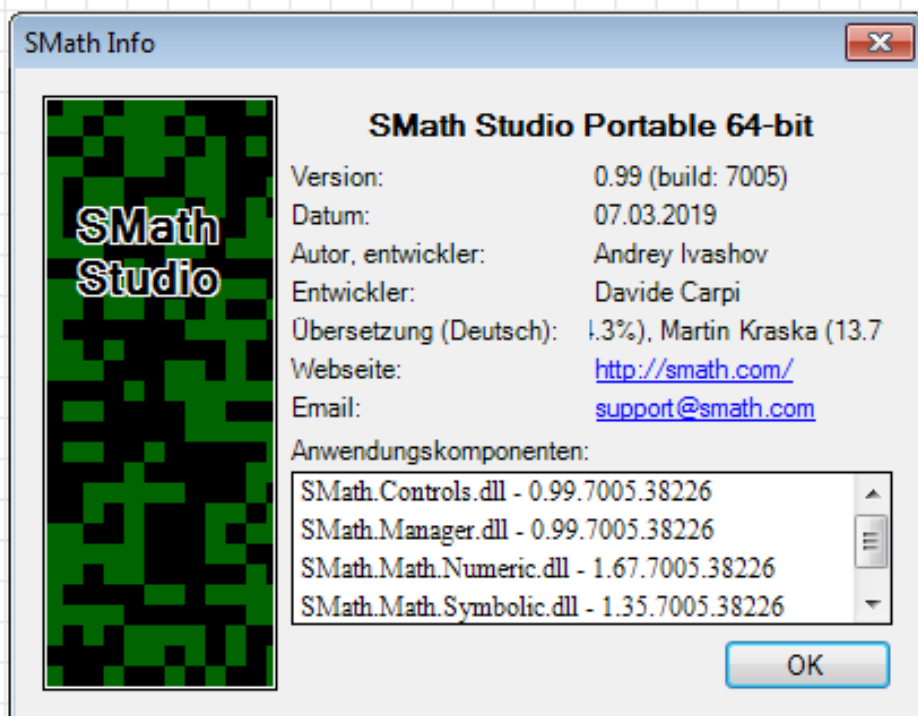
2. Legen Sie das Installationsverzeichnis an, z.B. *D:\SMath\SMath 0.99.7005* (eindeutige Versionsbezeichnung). Wenn Sie eine neue SMath-Version erproben möchten, legen Sie parallel dazu ein neues Verzeichnis an.
3. Entpacken von SMath Studio in das eben genannte Installationsverzeichnis.
4. Erzeugen Sie z.B. mit einem Texteditor die leere Datei *portable.version* im Installationsverzeichnis. Achten Sie darauf, dass Windows nicht noch eine unsichtbare Dateiendung anfügt.



5. Damit ist die portable Version eingerichtet. Sie können sie nun im Dateimanager starten (Datei *SMathStudio\_Desktop.exe*). Überzeugen Sie sich davon, dass Sie tatsächlich eine portable Version haben, indem Sie mit `SettingsDirectory()` das Einstellungsverzeichnis abfragen oder unter **Hilfe** > **SMath Info** das Impressum anzeigen lassen (das erfordert bereits das Plug-in *Custom Functions*).

```
appVersion(4) = "0.99.7005.38226"
```

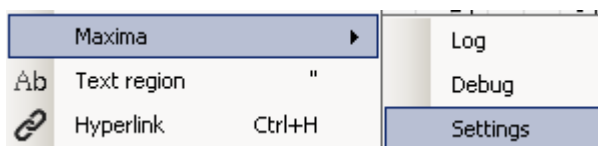
```
SettingsDirectory(1) = "C:\FHB\Software\SMath\SMath 99.7005\"
```



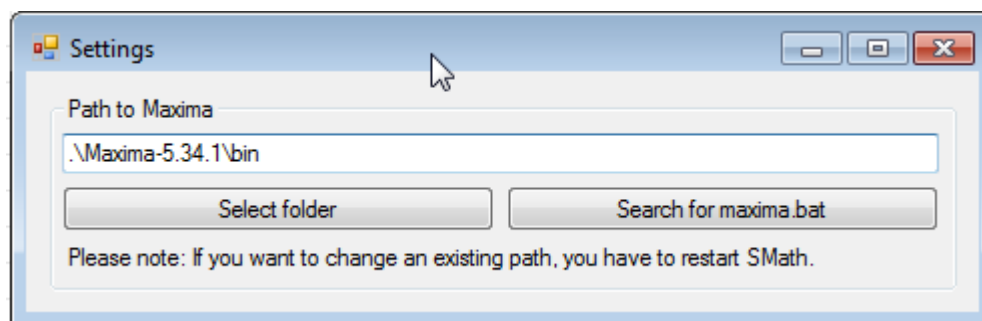
6. Laden Sie das Maxima-Plug-in und alle weiteren Plug-ins, die Sie benutzen wollen, aus der Online-Galerie im Erweiterungsmanager herunter und beenden Sie SMath Studio.
7. Laden Sie die Dokumentation, Code-Bausteine und Beispiele, die Teil der Installation sein soll, ebenfalls aus dem Erweiterungsmanager:
- Interaktives Handbuch (Setzen Sie den Haken bei „Standardmäßig nutzen“)
  - Dieses PDF-Handbuch
  - Beispiele: Balken-FEA, Hatch & Fill
  - Code-Bausteine: Draw, Hatch & Fill

## F.2. Konfiguration von SMath

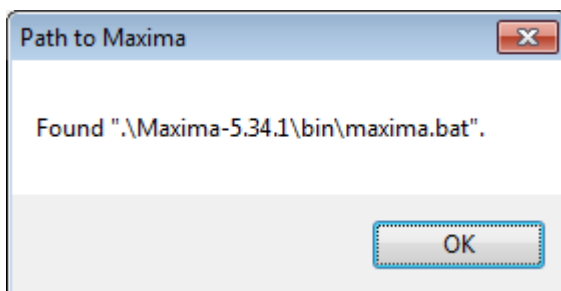
1. Starten Sie SMath neu. Öffnen Sie das Einstellungs-Fenster für die Schnittstelle:  
**Einfügen** > **Maxima** > **Settings**




2. Wählen Sie hier „Settings“ und geben Sie den Pfad Ihrer Maxima-Installation an



3. Wählen Sie nun „Search for maxima.bat“. Folgende Meldung sollte erscheinen:



4. Bestätigen Sie mit OK und schließen Sie das Settings-Fenster.
5. Testen Sie die Konfiguration mit der Testseite im Interaktiven Handbuch durchgeführt werden:

 > **Plugins** > **Maxima**

Bereits auf dieser Seite startet Maxima, weil einige Funktionen aufgerufen werden. Ganz unten findet man den Link auf die Testseite **Maxima Plugin Funktionstest**.

## F.3. Definition eigener Maßeinheiten

Maßeinheiten werden in der Datei `entries\Units.xml` (bezogen auf das Installationsverzeichnis) definiert. Man kann dort weitere Dateien dieses Formats ergänzen.

Einträge in `Units.xml`

<**prefixes**> Einheitenvorsätze definieren

<**add**> Vorsatz definieren

**factor** Faktor

**exp** Zehnerpotenz

**name** „x name“ Kennbuchstabe und ausgeschriebene Variante durch Leerzeichen getrennt.

**lang:xxx** Sprachvariante für den Namen

<**dimensions**> Dimensionen (Kategorien der Einheitenauswahl)

<**add**> Dimension hinzufügen

**baseunit** Standardeinheit für die Dimension. Wenn connection fehlt, handelt es sich um eine Basiseinheit.

**connection** Ausdruck der Standardeinheit durch Basiseinheiten

**id** Verweis auf den Namen (Zeilennummer in der Sprachdatei)

<**units**> Maßeinheiten definieren

<**property**> Physikalische Größe

**dimension** Zugeordnete Dimension

**id** Verweis auf den Namen (Zeilennummer in der Sprachdatei)

**title** englischer Name

**lang:xxx** Sprachversion des Namens

<**add**> Maßeinheit hinzufügen

**factor** Faktor gegenüber Standardeinheit

**exp** optional, Zehnerpotenz

**system** Einheitensystem: „Metric“, „Imperial“ oder „All“

**id** Verweis auf den Namen (Zeilennummer in Sprachdatei)

<**synonym**> Darstellungsart hinzufügen

**name** Darstellung der Einheit (gegebenenfalls Abkürzung)

**lang:xxx** Sprachversion der Darstellung

<**extension**> Vorsatz hinzufügen

**for** Darstellungsart

**lang:xxx** Sprachversion der Darstellungsart

<**prefix**>

**name** Kurzzeichen, wie im Block <extensions> definiert

**id** Verweis auf den Namen der Einheit mit Vorsatz (Zeilennummer in der Sprachdatei)

Einige nützliche Ids (entspricht der Id+9. Zeile):

| Id  | Inhalt                |
|-----|-----------------------|
| 252 | Winkel                |
| 265 | Frequenz              |
| 540 | Winkelgeschwindigkeit |
| 622 | Dimensionslos         |
| 634 | Prozent               |
| 635 | Promille              |

## F.4. Auslieferung

1. Aktualität der Plug-ins und des interaktiven Handbuchs sicherstellen (Erweiterungsverwaltung)
  - a) SVN Export in ein temporäres Verzeichnis
  - b) Zippen ohne Kompression als Activebook.zip
  - c) Hochladen auf <http://en.smath.info/upload/extensions.aspx>
2. PDF-Handbuch installieren
3. Ältere Plug-in- und Handbuchversionen entfernen
4. Funktionstest mit *Plugin Maxima Test.sm* (im Handbuchverzeichnis)
5. Dateihistorie zurücksetzen
6. Mathcad EFI konfigurieren
  - a) 32-bit-Version der exe erzeugen (`corflags /32bit+ SMathStudio_Desktop.exe`)
  - b) dlls installieren (aus *Installationsdaten/userefi* in das Unterverzeichnis *extensions\plugins\91a20d24-c30b-4774-9b1b-4d502a80420f\version\userefi* kopieren)
  - c) *welcome.sm* anpassen und als png abspeichern (für die Wiki-Seite)
7. (Eigene Maßeinheiten installieren.)
8. Einpacken in zip-File.
9. Zip-File in den **File-Manager** hochladen.
10. Eintrag auf der Wiki-Seite aktualisieren.

## F.5. Handlungsbedarf

- Zugeschnittene Maxima-Version verwenden
- Fehler beseitigen
- Maxima-Draw-Bereiche verbessern
- Seitenleisten-Palette ergänzen

### Bau einer Windows-Version von Maxima aus dem GIT-Repository

Aus der Maxima-Diskussions-Email-Liste von David Scherfgen <d.scherfgen@googlemail.com>, 23.2.15

This is a quick guide for Windows users who want to run the latest version of Maxima from the Git source repository together with wxMaxima. I needed to do this because some critical bugs in the latest "stable" build were making my life hard.

The steps don't involve any GNU make tools or C++ compilers and should be straightforward to follow.

1. Install the latest "stable" Maxima build for Windows from <https://sourceforge.net/projects/maxima/files/Maxima-Windows/>. We will use the wxMaxima version that comes with this build and just replace Maxima itself.
2. If you don't have one, install a Git client for Windows from <http://git-scm.com/download/win>.
3. Clone the Maxima Git repository into some directory. In this guide, I will assume that this is C:\Maxima-Git. The repository's URL is <git://git.code.sf.net/p/maxima/code>.
4. If you don't have it already, install Steel Bank Common Lisp (SBCL) for Windows from <http://www.sbcl.org/platform-table.html>. Since I'm on a 64-bit Windows, I chose the 64-bit version, and it worked fine.
5. Now we will build Maxima using SBCL only. For that, precisely follow the instructions found in the INSTALL.lisp file in C:\Maxima-Git:
  - a) Open a command prompt and CD to C:\Maxima-Git.
  - b) Start SBCL by entering: sbcl
  - c) Now we configure Maxima by entering (load "configure.lisp"), then (configure), and finally (quit)
  - d) Navigate to the C:\Maxima-Git\src directory: CD src
  - e) Start SBCL again: sbcl
  - f) Now we compile Maxima by entering (load "maxima-build.lisp"), then (maxima-compile), and finally (quit)
  - g) Start SBCL again: sbcl
  - h) Now we load the compiled Maxima program by entering (load "maxima-build.lisp") again, then (maxima-load)
  - i) Finally, we dump an image by entering (maxima-dump)
6. After these steps, you should have a compiled version of the latest Maxima. Verify that everything worked by running the maxima.bat file that you should find in C:\Maxima-Git\src. The Maxima command prompt should show up.
7. Now we will make wxMaxima use our self-compiled Maxima version:
  - a) Open an Explorer window and go to the bin subdirectory of your latest "stable" Maxima version. For me, this is C:\Program Files (x86)\Maxima-sbcl-5.35.1.2\bin.
  - b) Rename the maxima.bat file to maxima.bat.bak.
  - c) Rename the sbcl.exe file to sbcl.exe.bak. You need to do this because the supplied version of SBCL seems to be outdated, and that would result in troubles. By renaming the file, the SBCL version you earlier installed on your machine will be used instead.
  - d) Copy C:\Maxima-Git\src\maxima.bat to the current directory.
8. Verify that everything worked by starting wxMaxima and entering build\_info(); - pay attention to the timestamp field, which should reflect the date and time when you compiled Maxima earlier.

9. That's it! From time to time, you should pull new commits from the repository and rebuild Maxima by following step 5 again. Copying the files again (step 7) should not be necessary.
10. If you want to undo the changes, delete the maxima.bat file and rename maxima.bat.bak back to maxima.bat and sbcl.exe.bak back to sbcl.exe. I hope this guide proves useful. If you have any comments or additions, please don't hesitate. Maybe we can put the guide somewhere into the Git repository or the website.

Start Over with Maxima



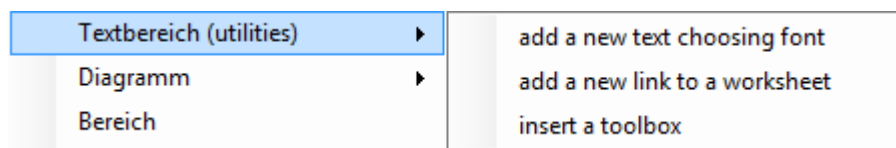
# G. Dachboden

Hier finden sich Abschnitte zu Funktionen, die nicht für die standardmäßige Benutzung empfohlen werden, da sie durch leistungsfähigere oder bedienungsfreundlichere ersetzt wurden. In alten Rechenblättern können diese allerdings noch vorhanden sein oder es mag im Einzelfall Gründe geben, sie auch in neuen Rechenblättern einzusetzen.

## G.1. Schriftart in Textbereichen ändern

Wer Texte in anderen Schriftarten schreiben will, sollte den erweiterten Textbereich (Abschnitt 5.6) benutzen. Daher wurde dieser Abschnitt in den Dachboden verschoben.

SMath Studio bringt von Haus aus gegenwärtig (Februar 2018) keine Möglichkeit mit, die Schriftart von Textbereichen zu ändern. Das Plug-in *Text Region Utilities* bietet aber Alternativen an. Es erzeugt im Menü „Einfügen“ neue Einträge:



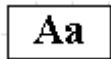
**Add a new text choosing font** Neuen Textbereich mit wählbarer Schriftart einfügen. Angeboten werden alle im System vorhandenen Schriftarten. Zum Gitterraster passen optimal die 11-Punkte Schriften:

|                            |                             |                            |
|----------------------------|-----------------------------|----------------------------|
| Arial Standard 11pt        | Times New Roman 11pt        | Standard SMath 11pt        |
| Arial Standard 11pt        | Times New Roman 11pt        | Standard SMath 11pt        |
| Arial Standard 11pt        | Times New Roman 11pt        | Standard SMath 11pt        |
| <b>Arial Standard 11pt</b> | <b>Times New Roman 11pt</b> | <b>Standard SMath 11pt</b> |
| <i>Arial Standard 11pt</i> | <i>Times New Roman 11pt</i> | <i>Standard SMath 11pt</i> |
| <u>Arial Standard 11pt</u> | <u>Times New Roman 11pt</u> | <u>Standard SMath 11pt</u> |

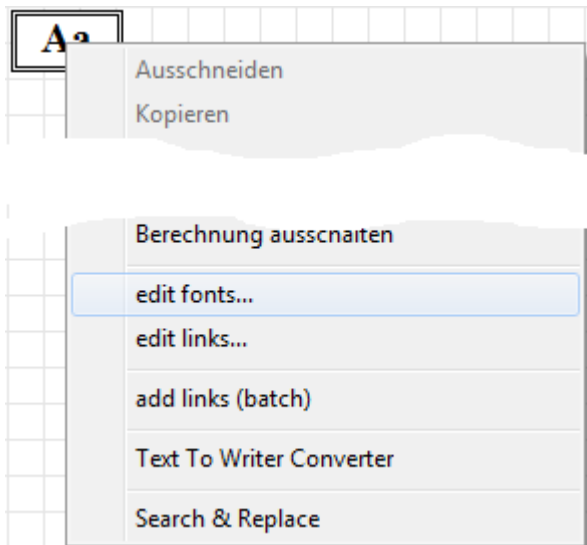
Die nachträgliche Änderung der Schriftart von Textbereichen ist etwas umständlich. Es wäre logisch und einfach, in der Werkzeugleiste oder im Kontextmenü von Textbereichen einen Eintrag zur Auswahl der Schriftart zu haben. Leider ist dies im Basisprogramm noch nicht realisiert und es gibt auch keine Schnittstelle, dies über ein Plug-in zu ergänzen. Das hat zunächst die Folge, das man die Schriftart bestehender Bereiche nicht ändern kann.

Im Plug-in *Text Region Utilities* wird daher ein neuer Bereichstyp (Toolbox) angeboten, der alle Texte kennt und deren Änderung über sein Kontextmenü anbietet. Er wird folgendermaßen angewendet:

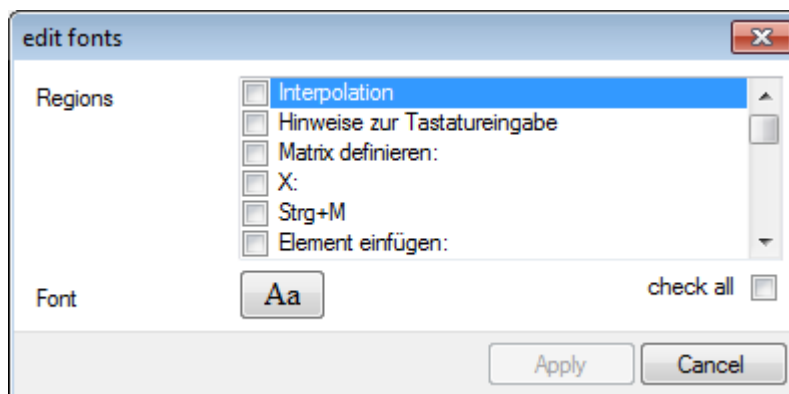
1. Einfügen des Toolbox-Bereichs mit **Einfügen** > **Textbereich (Utilities)** > **insert a toolbox**



2. Kontextmenü der Toolbox öffnen (rechte Maustaste) und den Eintrag „edit fonts“ auswählen:



3. Im Dialog „edit fonts“ sind alle vorhandenen Texte aufgelistet. Sie können einzeln oder insgesamt für die Änderung der Schriftart ausgewählt werden.



4. Mit dem Knopf Font [Aa] ruft man dann den Schriftauswahldialog auf. Dort wird mit OK bestätigt und dann im Dialog „edit fonts“ mit [Apply] angewendet (erst dann sieht man, ob die Einstellung sinnvoll war)

## G.2. Gleichungslösung mit solve() und roots()

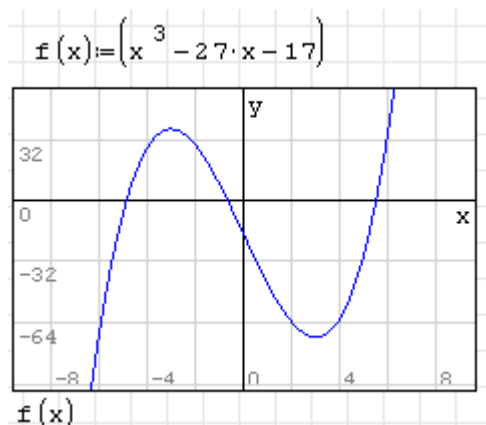
Diese Funktionen sollten Sie nur nutzen, wenn Sie genau wissen, warum Sie die in Abschnitt 6.11 beschriebenen Funktionen nicht benutzen wollen. Das kann z.B. der Fall sein, wenn Sie die Plug-ins nicht benutzen können oder eine ausführbare Anwendung erzeugen wollen (dies funktioniert aktuell mit Maxima nicht).

SMath bietet zwei eingebaute Funktionen, die auch nichtlineare Gleichungen lösen können: `solve()` und `roots()`. Bei beiden Funktionen ist das erste Argument die Gleichung, das zweite Argument die Unbekannte, nach der aufzulösen ist. Wird nur eine Seite der Gleichung angegeben, dann wird angenommen, dass die Nullstellen dieses Ausdrucks zu suchen sind. Als rechte Seite wird also eine Null ersetzt.

`solve()` sucht innerhalb eines Intervalls nach den Lösungen, `roots()` sucht von einem Startpunkt aus nach einer Lösung und kann Gleichungssysteme nach mehreren Variablen auflösen.

Die Gleichung kann einheitenbehaftet sein, die Unbekannte wird aber als dimensionslos betrachtet. Man kann also nicht direkt nach Längen, Kräften oder ähnlichem suchen. Vielmehr muss man die Variable dann als Vielfaches einer Bezugsgröße einbauen. Das ist insofern lästig, als dass in praktischen Anwendungen meist nach einheitenbehafteten Größen zu suchen ist.

Es ist stets zu empfehlen, die Funktion, deren Nullstellen man sucht, zunächst zu plotten, um eine Vorstellung davon zu bekommen, wie die Funktion aussieht und wo die Nullstellen ungefähr liegen könnten. Auch dabei ist zu beachten, dass die Variable  $x$  im Plot dimensionslos ist.



In diesem Beispiel benutzt `solve()` den voreingestellten Suchbereich (**Extras** > **Einstellungen** > **Berechnung** > **Wurzeln (Bereich)**). Er ist von -20 bis 20 eingestellt und enthält alle Nullstellen der Funktion  $f(x)$ .

$$\text{solve}(f(x); x) = \begin{pmatrix} -4,85 \\ 0,64 \\ 5,49 \end{pmatrix}$$

Wie man sieht, finden `solve()` alle Nullstellen. Im folgenden Fall werden die Nullstellen von  $f(x/10)$  gesucht. Sie entsprechen dem Zehnfachen der oben gefundenen Werte. Nun liegt nur noch eine Nullstelle im voreingestellten Intervall:

$$\text{solve}\left(f\left(\frac{x}{10}\right); x\right) = -6,39$$

Man kann aber `solve()` explizit ansagen, in welchem Intervall gesucht werden soll. Diese Angabe überstimmt die Voreinstellung, ändert sie jedoch nicht. Das geht nur über das Menü.

$$\text{solve}(f(x); x; -100; 100) = \begin{pmatrix} -4,85 \\ 0,64 \\ 5,49 \end{pmatrix}$$

Damit man nicht von der Voreinstellung abhängig ist, sollte man stets die Bereichsgrenzen angeben.

Die Funktion `roots()` findet von vornherein nur eine Lösung.

```
roots(f(x); x)=-0,64
```

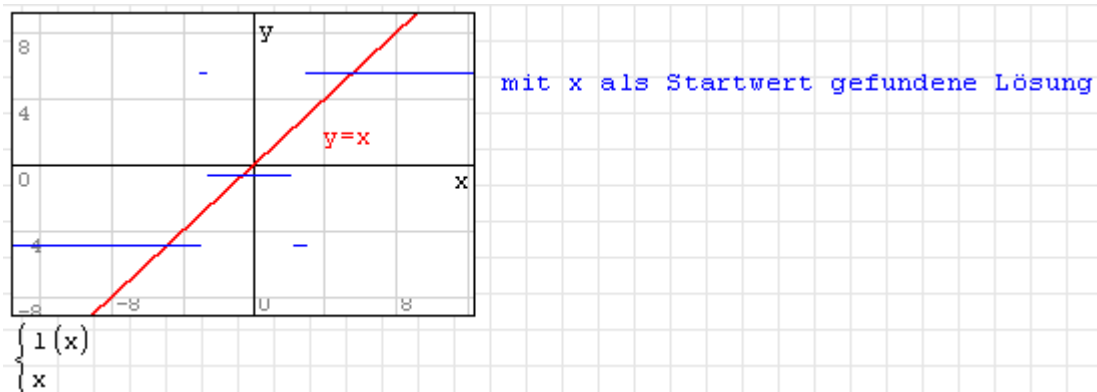
Diese ist vom Startpunkt der Suche abhängig, den der Anwender nicht kennt, wenn er ihn nicht vorgibt. Vermutlich ist Null als Startwert voreingestellt, aber das kann man weder direkt prüfen, noch ändern. Der Startpunkt kann aber als drittes Argument angegeben werden:

```
roots(f(x); x; 4)=5,49
```

Man darf sich nicht darauf verlassen, dass immer die zum Startwert nächstliegende Lösung gefunden wird. Das kann man mit verschiedenen Startwerten ausprobieren. Dafür definiert man sich eine Funktion, die den Startwert als Argument bekommt und die damit bestimmte Lösung als Ergebnis liefert. Damit sich SMATH nicht beschwert, dass  $x$  in der Funktion unbekannt ist, packen wir die Rechnung in einen Anweisungsblock (in welchem lokale Variablen möglich sind).

```
l(x_0):=roots(f(y); y; x_0)
```

Ein Plot dieser Funktion (blau) zeigt nun, mit welchem Startwert man welche Lösung bekommt. zum Vergleich ist in Rot die Linie  $y = x$  gezeichnet.



Der Plot wurde auf Punktdarstellung umgeschaltet, damit die blaue Kurve der gefundenen Lösungen keine vertikalen Linien enthält.

## Gleichungslösung mit Maßeinheiten

Wie bereits erwähnt, werden die Variablen, nach denen `roots()` und `solve()` auflösen, als dimensionslose Zahlen betrachtet. Wenn der Wert einer einheitenbehafteten physikalischen Größe ermittelt werden soll, so muss diese in der zu lösenden Gleichung als Produkt aus Zahlenwert (Variable, nach der gelöst wird) und einer Maßeinheit dargestellt werden. Das Ergebnis ist dann mit der gleichen Maßeinheit zu multiplizieren, um die gesuchte Größe zu bestimmen.

Hier ein Beispiel:

```
Assmann/Selke, Technische Mechanik 2, Aufgabe 7.7
```

```
Eine Druckspindel,  $l = 800$  mm lang, mit  $F = 20$  kN belastet, soll als  
Hohlwelle mit  $d_i = 25$  mm ausgeführt werden. Für  $S_K > 3,5$  ist der Außendurchmesser auf volle mm gerundet zu bestimmen.
```

```
7-7  $d_a = 33$  mm;  $F_d = 23$  kN
```

|                                    |  |
|------------------------------------|--|
| $l := 800 \text{ mm}$              | Spindellänge                           |
| $F_{\text{dzul}} := 20 \text{ kN}$ | Druckkraft                             |
| $d_i := 25 \text{ mm}$             | Innendurchmesser                       |
| $S_K := 3,5$                       | Knicksicherheit                        |
| $E = 210000 \text{ MPa}$           | E-Modul Stahl (Annahme)                |
| $l_K := 1$                         | beidseitig gelenkig gelagert (Annahme) |

Das Flächenträgheitsmoment  $I_{\min}$  wird durch den Zahlenwert  $\{I_{\min}\} = zI_{\min}$  und eine beliebige Längeneinheit zur vierten Potenz ausgedrückt. Das Ergebnis ist nicht abhängig von der gewählten Einheit. Die Funktion `roots()` sucht Werte für  $zI_{\min}$ , die die gegebene Gleichung erfüllen. An der Gleichung selbst muss nichts geändert werden. Die Löser berücksichtigen bei der Auswertung alle bereits definierten Zusammenhänge für die in der Gleichung vorkommenden Größen. Daher kann nach  $zI_{\min}$  aufgelöst werden, auch wenn es in der Gleichung nicht explizit vorkommt.

|   |                                |
|---|--------------------------------|
| Erforderliches Flächenträgheitsmoment   |                                |
| $I_{\min} := zI_{\min} \text{ cm}^4$  |                                |
| $zI_{\min} := \text{roots} \left( F_{\text{dzul}} = \frac{\pi^2 \cdot E \cdot I_{\min}}{S_K \cdot l_K^2}; zI_{\min}; 0 \right)$ | $I_{\min} = 2,16 \text{ cm}^4$ |

Die Bestimmungsgleichung zur Ermittlung des Durchmessers ist nichtlinear. Die Vorgehensweise ist aber identisch.

|  |                                  |
|--|----------------------------------|
| Erforderlicher Außendurchmesser  |                                  |
| $D := zD \text{ mm}$   | +                                |
| $zD := \text{roots} \left( I_{\min} = \frac{\pi}{64} \cdot (D^4 - d_i^4); zD; \frac{d_i}{\text{mm}} \right)$ | $D = 30,19 \text{ mm}$           |
| $D_{\text{gew}} := \text{Ceil} \left( \frac{D}{\text{mm}} \right) \text{ mm}$                                | $D_{\text{gew}} = 31 \text{ mm}$ |

## G.3. Ebene Polygone

Ebene Polygone sind Grafikobjekte, die in SMATH als speziell formatierte Matrizen dargestellt werden. Funktionen zum Umgang mit solchen Objekten werden vom Plug-in *GPCPlug-in* bereitgestellt.

Das Plug-in bindet eine Bibliothek **GPC** der Uni Manchester für 2D-Geometrieoperationen mit Polygonen (stückweise geradlinig berandete Flächen) ein. Die Bibliothek ist frei nutzbar für nichtkommerzielle Zwecke. Lizenzinformationen finden Sie [hier](#). Der Leistungsumfang beinhaltet:

- Geometrieoperationen: Differenz, Schnittmenge, Exklusives Oder und Vereinigung,
- Flächen können durch mehrere Randkonturen gebildet werden.
- Die Randpunkte können mit oder gegen den Uhrzeigersinn angegeben werden.
- Ränder können konvex, konkav oder selbstüberschneidend sein.
- Ränder können geschachtelt werden (d.h. Flächen können Löcher haben).
- Ergebnisformat als Randkurve (contour) oder Dreieckskette (triangle strip),
- Außen- und Innenränder im Ergebnis unterscheidbar,
- Robust gegen doppelte Punkte und entartete Gebiete.

Polygone werden als Matrizen dargestellt:

$$\begin{bmatrix} r & p_1 & \dots & p_r & x_{11} & \dots & x_{1p_1} & \dots & x_{r1} & \dots & x_{rp_r} \\ p & f_1 & \dots & f_r & y_{11} & \dots & y_{1p_1} & \dots & y_{r1} & \dots & y_{rp_r} \end{bmatrix}$$

$r$  Zahl der Randkurven

$p$  Gesamtzahl der Eckpunkte

$p_i$  Zahl der Eckpunkte in Randkurve  $i \in 1 \dots r$

$f_i$  Schalter für Randkurve  $i$ : 1 = Innenrand (schneidet ein Loch), 0 = Außenrand

$x_{ij}$   $x$ -Koordinate des  $j$ -ten Punktes von Randkurve  $i$  mit  $j \in 1 \dots p_i$

$y_{ij}$   $y$ -Koordinate des  $j$ -ten Punktes von Randkurve  $i$

Die Matrix hat demzufolge  $1 + r + \sum p_i$  Spalten.

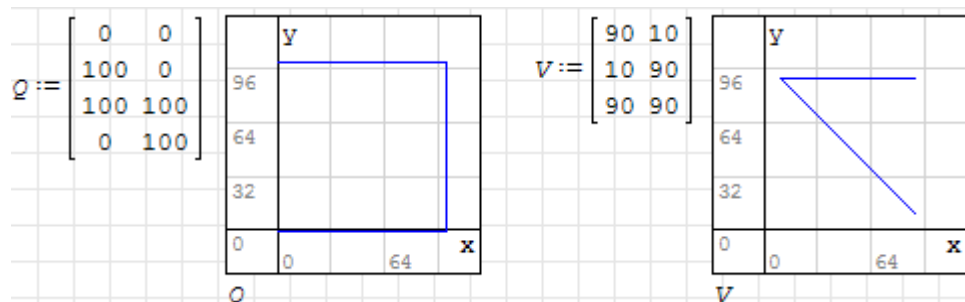
Die gleiche innere Darstellung wird für Dreiecksketten verwendet, dabei ist  $r$  die Zahl der Dreiecksketten und  $f_i$  ist stets gleich Null.

Beispiele (im russischen SMath-Forum)<sup>1</sup>

### G.3.1. Polygon-Objekte erzeugen

Polygon-Objekte können aus Koordinatenmatrizen erzeugt werden, die die Randpunktfolge darstellen (Zeilen: Punkte, Spalte 1:  $x$ , Spalte 2:  $y$ ). Die Randpunktfolge kann, muss aber nicht geschlossen sein.

Gegeben sind beispielsweise zwei Matrizen  $Q$  und  $V$ .  $Q$  enthält 4 Spalten und stellt ein Quadrat dar,  $V$  stellt die drei Ecken eines Dreiecks dar, das vollständig in das Quadrat hineinpasst:



<sup>1</sup>[http://ru.smath.info/forum/yaf\\_postst1274\\_Plaghin-dlia-opieratsii-s-mnogoughol-nikami.aspx](http://ru.smath.info/forum/yaf_postst1274_Plaghin-dlia-opieratsii-s-mnogoughol-nikami.aspx)

Derartige Koordinatenlisten kann man mit `gpc_polygon()` zu einem Polygon-Objekt machen. Spalte 1 der Ergebnismatrix besagt, dass es einen Rand und 4 Punkte im ganzen Polygon gibt. Spalte 2 gibt an, dass der erste Rand 4 Punkte hat und ein Außenrand ist (Kennzahl 0, Innenrand wäre Kennzahl 1). Die weiteren Spalten sind die  $x, y$ -Koordinaten der jeweiligen Punkte.

```
P := gpc_polygon(Q)  P = [ 1 4 0 100 100 0 ]
                       [ 4 0 0 0 100 100 ]
```

Fügen wir nun das Dreieck als Innenrand hinzu. Die Funktion `gpc_add_contour(Polygon; Punktematrix; Kennzahl)` fügt je nach Kennzahl einen Außen- oder einen Innenrand hinzu.

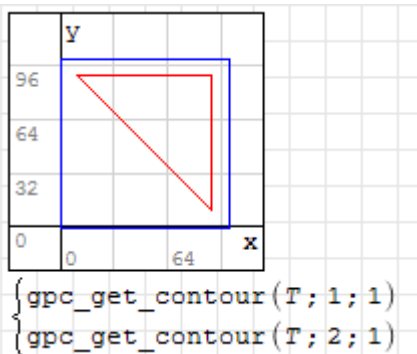
```
T := gpc_add_contour(P; V; 1)  T = [ 2 4 3 0 100 100 0 90 10 90 ]
                                   [ 7 0 1 0 0 100 100 10 90 90 ]
```

Das Polygon  $T$  besteht nun aus 2 Rändern mit insgesamt 7 Punkten. Der erste Rand hat 4 Punkte und ist ein Außenrand (das erkennt man an Spalte 2), der zweite Rand (Info aus Spalte 3) enthält 3 Punkte und ist ein Innenrand. Es folgen die Punkte in der Reihenfolge der Ränder, also erst die vier Punkte des Außenrandes und dann die drei Punkte des Innenrandes.

### G.3.2. Polygon-Objekte darstellen

Das Ergebnis von Polygon-Operationen will man natürlich auch darstellen können. Im Plug-in *GPCPlug-in* gibt es die Funktion `gpc_get_contour(Polygon; Rand; Schalter)`, die einen beliebigen Rand wieder in eine plotbare Koordinatenmatrix umwandelt. Ist der Schalter 0, wird ein offener Rand erzeugt, ist er 1, wird der Rand geschlossen, d.h. der erste ist gleich dem letzten Punkt.

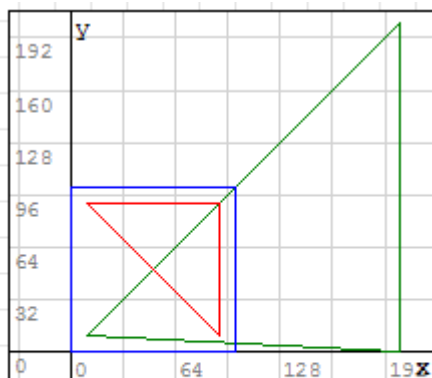
```
gpc_get_contour(T; 2; 0) = [ 90 10 ]
                           [ 10 90 ]
                           [ 90 90 ]
gpc_get_contour(T; 2; 1) = [ 90 10 ]
                           [ 10 90 ]
                           [ 90 90 ]
                           [ 90 10 ]
```



### G.3.3. Polygon-Operationen

Erzeugen wir zunächst ein weiteres Polygon  $N$ , welches beide Ränder von  $T$  schneidet und sich auch außerhalb von  $T$  erstreckt:

```
N := gpc_polygon(
  [
    [ 10 10 ]
    [200 200]
    [200  0]
  ]
)
N = [
  [ 1 3 10 200 200 ]
  [ 3 0 10 200  0 ]
]
```



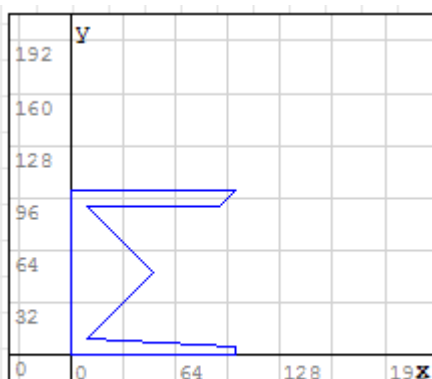
```
{ gpc_get_contour(T; 1; 1)
  gpc_get_contour(T; 2; 1)
  gpc_get_contour(N; 1; 1)
```

Polygon-Operationen werden mit der Funktion `gpc_clip(Kennzahl; Polygon1; Polygon2)` ausgeführt. Die Kennzahl steht für die auszuführende Operation:

| Kennzahl | Operation       |
|----------|-----------------|
| 0        | Differenz       |
| 1        | Schnittmenge    |
| 2        | Exklusives Oder |
| 3        | Vereinigung     |

**Differenz:** Polygon  $N$  wird von  $T$  abgezogen. Ergebnis sind die Flächen, die in  $T$ , aber nicht in  $N$  vorhanden sind.

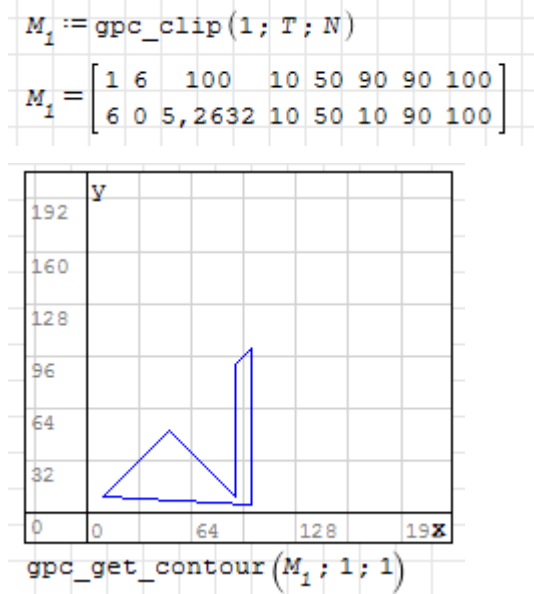
```
M_0 := gpc_clip(0; T; N)
M_0 = [
  [ 1 9 90 10 50 10 100 100 0 0 100 ]
  [ 9 0 90 90 50 10 5,2632 0 0 100 100 ]
]
```



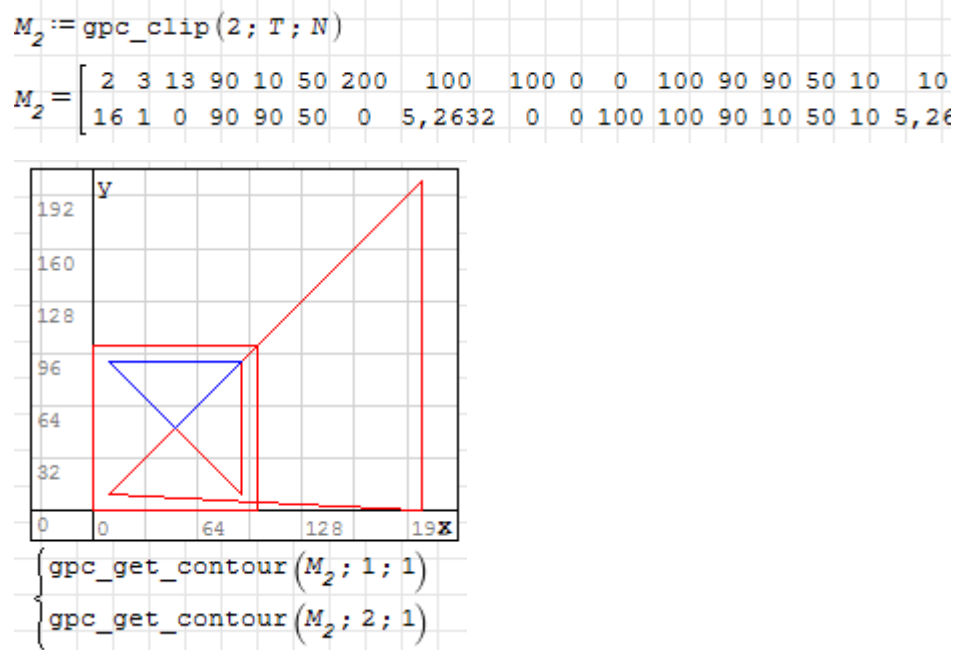
```
gpc_get_contour(M_0; 1; 1)
```

**Schnittmenge:** Ergebnis ist die gemeinsame Fläche beider Polygone.

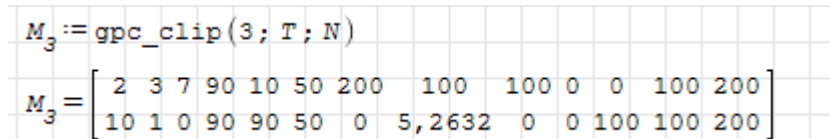


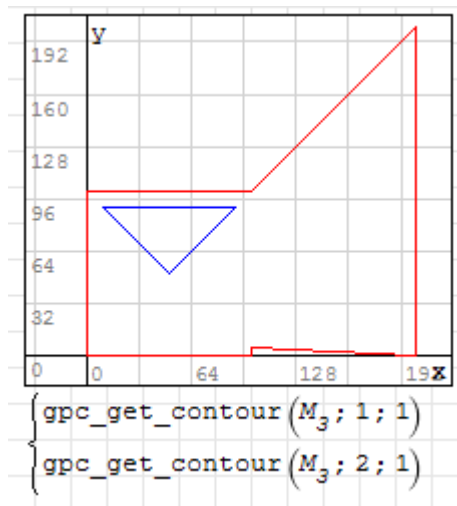


**Exklusives Oder:** Ergebnis sind die Flächen, die nur in einem der beiden Polygone vorkommen.



**Vereinigung:** Das sind die Flächen, die in mindestens einem der beiden Polygone vorhanden sind.





### G.3.4. Polygon-Objekte lesen und schreiben

Das Plug-in bietet auch zwei Funktionen zum Lesen und schreiben. Das letzte Argument ist jeweils der Schalter für die Randkennzahl. Steht er auf 1, wird die Randkennzahl (innerer oder äußerer Rand) mitgeschrieben und gelesen, ansonsten nicht. Verwenden Sie nur einen anderen Wert als 1, wenn Sie genau wissen, warum.

```
dir := CurrentDirectory(DocumentDirectory(""))
dir = "C:\FHB\Software\SMath\SMath Skript\Bilder\"
T = [ 2 4 3 0 100 100 0 90 10 90
      7 0 1 0 0 100 100 10 90 90 ]
gpc_write_polygon(T; "GpcTest.txt"; 1) = 1

R := gpc_read_polygon("GpcTest.txt"; 1)
R = [ 2 4 3 0 100 100 0 90 10 90
      7 0 1 0 0 100 100 10 90 90 ]
```

Der Dateiname kann absolut oder relativ (bezogen auf das aktuelle Verzeichnis) sein.

## G.4. 3D-Diagramme (Standard)

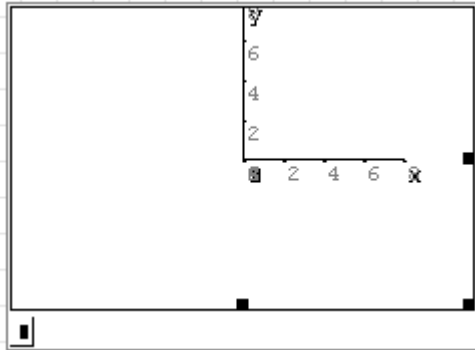
SMath kann 3D-Diagramme darstellen. Die Funktion ist allerdings noch sehr unausgereift und wurde auch in den letzten Jahren nicht weiterentwickelt:

- Die Ansicht (Zoom, Richtung und Verschiebung) werden nicht korrekt in der Rechenblattdatei gespeichert.
- Die angezeigten Achsen verlieren die Skalierung und Richtung bei Neuberechnung des Blattes.
- Das Koordinatensystem ist linkshändig.
- Bereichsgrenzen und Netzdichte für Funktionen von  $x$  und  $y$  können nicht beeinflusst werden.

Als Alternative werden die Maxima-Draw-Funktionen empfohlen (Abschnitt 8.1).

### G.4.0.1. 3D-Diagrammbereich erzeugen

Einfügen > Grafik > Dreidimensional



Der Diagrammbereich erscheint dann mit der  $x$ -Achse nach rechts, der  $y$ -Achse nach oben und der  $z$ -Achse nach hinten (Linkssystem!). Diese Ansicht wird stets wiederhergestellt, wenn die Datei geladen wird oder wenn der Knopf „Aktualisieren“ im Grafikpanel der Seitenleiste gedrückt wird.



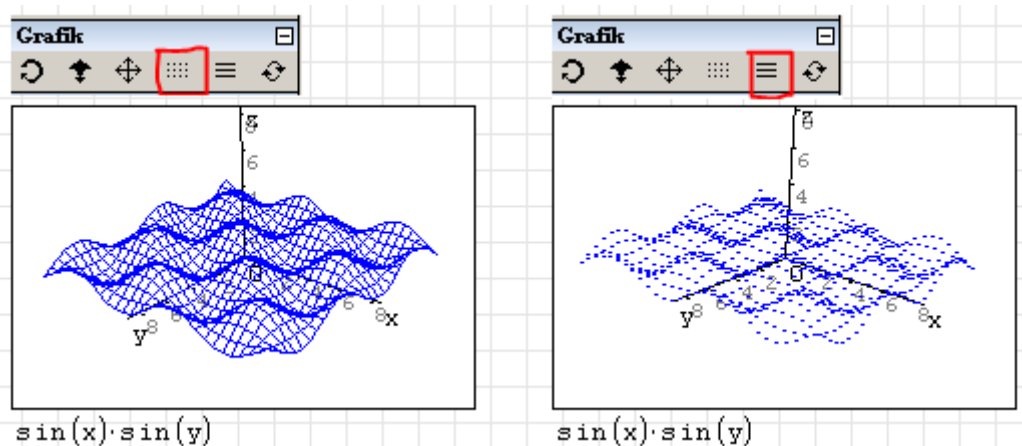
Der Diagrammbereich kann mit der Maus am rechten und unteren Rand in die gewünschte Größe gezogen werden.

Die Einfügemarke links unten nimmt darstellbare Objekte oder Listen darstellbarer Objekte auf.

### G.4.0.2. Darstellbare Objekte

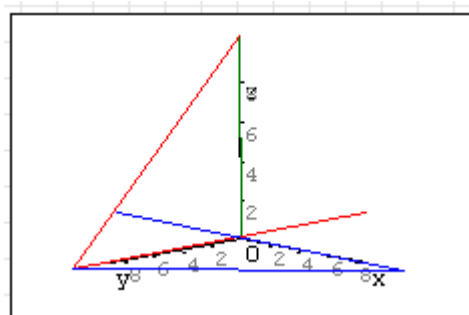
**Funktionen von  $x$  und  $y$ .** Dabei kann weder auf die Zahl der Gitterlinien, noch auf den Wertebereich von  $x$  und  $y$  Einfluss genommen werden.

Im Grafikpanel der Seitenleiste kann man einstellen, ob man Linien oder Punkte sehen will.



**Dreispartige Matrizen**, deren Zeilen jeweils die Koordinaten  $x$ ,  $y$  und  $z$  eines Punktes enthalten. Die Punkte einer Matrix werden mit einem Linienzug verbunden. Das

Koordinatensystem scheint linkshändig zu sein. Die  $x$ -Achse weist nach vorn rechts, die  $y$ -Achse weist nach vorn links und die  $z$ -Achse weist nach oben. Dies passt nicht zu einem rechtshändigen Koordinatensystem.



$$\begin{pmatrix} -10 & 0 & 0 \\ 10 & 0 & 0 \\ 0 & 10 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & -10 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 10 \end{pmatrix}$$

**Parametrische Flächen** Mit der Funktion `CreateMesh()` aus dem Plug-in `3DPlotRegion` können Gitternetze aus parametrischen Funktionen

$$\begin{bmatrix} x(u, v) \\ y(u, v) \\ z(x, y) \end{bmatrix}$$

erzeugt werden. Mit der Einstellung

$$\begin{bmatrix} u \\ v \\ f(u, v) \end{bmatrix}$$

erzeugt diese Funktion Darstellungen analog zur eingebauten Funktionsdarstellung, allerdings sind dann Wertebereich und Netzdichte einstellbar. Siehe dazu auch das Beispiel in Abschnitt ??.



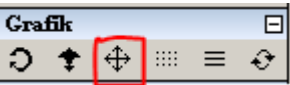
### G.4.0.3. Navigation im 3D-Diagrammfenster

Die Ansicht kann im 3D-Fenster mit der Maus geändert werden. Diese Einstellungen bleiben aber beim Schließen der Datei nicht erhalten.

Beachten Sie: Koordinatenachsen und angezeigte Grafikobjekte passen nur zusammen, wenn vorher der Knopf „Aktualisieren“ im Grafikpanel der Seitenleiste gedrückt wird.



Die Navigationsfunktionen werden in der Seitenleiste im Panel „Grafik“ angewählt.

|         |   |   |   |
|---------|---|---|---|
|         |  |  |  |
| Aktion  | Ziehen  | Drehen  | Zoom  |
|         |   |   | Verschieben   |
| Mausrad | Zoom  | Zoom  | Zoom  |

## G.5. Diagramme mit plot2D/3D

In Maxima gibt es zwei Wege zu Diagrammen zu kommen:

1. Das Paket *draw* mit den Funktionen `draw2d()` und `draw3d()`. Diese werden für den Standardanwender zur Benutzung empfohlen.
2. Die Funktionen `plot2d()` und `plot3d()`. Diese sind weniger flexibel, wurden daher nicht so stark im Maxima-Plug-in unterstützt, wie das Paket *draw*.

Dieser Abschnitt zur Dokumentation der Funktionen `plot2d()` und `plot3d()` fällt daher auch sehr knapp aus.

Es stehen folgende Plot-Befehle in Maxima zur Verfügung

`contour_plot(expr; xrange; yrange...)` Höhenliniendiagramm für eine einzelne Funktion

`implicit_plot(expr_list; xrange; yrange...)` Darstellung einer oder mehrerer implizit gegebener Funktionen. Muss mit `load(implicit_plot)` geladen werden.

`plot2d(expr_list; xrange,...)` Darstellung von expliziten Funktionen

`plot2d(expr_list;...)` parametrische Funktionen oder Punktmengen, hier ist die Option `xrange` nicht Pflicht.

- Diskrete Punktmengen: Stellt Listen von Koordinatenpaaren dar, z.B. mittels `args(Matrix)` erzeugt)

$$M := \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{pmatrix} \quad \left\{ \begin{array}{l} \text{discrete} \\ \text{args}(M) \end{array} \right.$$

- Parametrische Funktion:

$$\left\{ \begin{array}{l} \text{parametric} \\ fx \\ fy \\ \left\{ \begin{array}{l} t \\ tmin \\ tmax \end{array} \right. \end{array} \right.$$

`plot3d(expr; xrange; yrange...)` Parametrische oder explizit gegebene Funktion

### G.5.1. Optionen

(immer als Listen mit dem Schlüsselwort zuerst angeben)

| Option                        | (Voreinstellung): Bedeutung  |
|-------------------------------|--|
| <code>adapt_depth</code>      | (5): Maximale Teilungstiefe der adaptiven Kurvenabtastung  |
| <code>axes</code>             | (true): beide Achsen zeichnen<br>false: keine Achse zeichnen<br>x oder y: eine Achse zeichnen  |
| <code>azimuth</code>          | (30): Drehwinkel um $z$ in $^\circ$  |
| <code>box</code>              | (true): Kastenrand zeichnen<br>false: keinen Kastenrand zeichnen   |
| <code>color</code>            | (blue, red, green, magenta, black, cyan): Farben für die Kurven oder die Ober- und Unterseite von 3D-Flächen (wenn keine Palette verwendet wird)   |
| <code>color_bar</code>        | (false): keine Farblegende zeichnen<br>true: Farblegende zeichnen  |
| <code>color_bar_ticks</code>  | ( ): Achsmarken an der Farblegende, Startwert oder Startwert und Inkrement oder Startwert, Inkrement und Endwert.  |
| <code>elevation</code>        | (60): Drehung um $x$ in $^\circ$   |
| <code>gnuplot_term</code>     | ( ): Ausgabe-Format-Spezifikation (Text nach „set terminal“)   |
| <code>gnuplot_out_file</code> | (maxplot.xxx) Ausgabedateiname   |
| <code>grid</code>             | (30, 30): Abtastpunkte in $x$ - und $y$ -Richtung bei 3D-Diagrammen  |
| <code>grid2d</code>           | (false): Gitternetz in der $xy$ -Ebene   |
| <code>iterations</code>       | (9): Iterationen bei <code>mandelbrot</code> und <code>julia</code>  |
| <code>label</code>            | Liste mit text, x, y. Text wird an den gegebenen Koordinaten angezeigt.  |
| <code>legend</code>           | (Funktionsnamen): Liste der Legendeneinträge für die Kurven<br>false: keine Legende zeichnen   |
| <code>logx</code>             | ( ): wenn vorhanden, wird $x$ logarithmisch skaliert   |
| <code>logy</code>             | ( ): wenn vorhanden, wird $y$ logarithmisch skaliert   |
| <code>mesh_lines_color</code> | (black): Farbe der Gitterlinien, wenn eine Farbpalette benutzt wird.<br>false: keine Gitterlinien anzeigen   |
| <code>nticks</code>           | (29): Punktezahl für die Kurvenabtastung   |
| <code>palette</code>          | ({hue; 0,25, 0,7; 0,8; 0,5}) Farbpalette für die Flächenfärbung<br>false: Flächen nicht färben   |
| <code>png_file</code>         | ( ): Ausgabe in PNG-Datei  |
| <code>point_type</code>       | (Liste): Symbole für Kurvenart <code>points</code> oder <code>linespoints</code>   |
| <code>same_xy</code>          | ( ): gleiche Skalierung für die $x$ - und $y$ -Achse   |
| <code>same_xyz</code>         | ( ): gleiche Skalierung für die $x$ -, $y$ - und $z$ -Achse  |
| <code>style</code>            | (lines):, Parameter: Linienstärke, Farbe (1: blau, 2: rot, 3: magenta, 4: orange, 5: braun, 6: lime, 7: aqua<br>points: Parameter: Radius, Farbe (wie bei lines), Symbol<br>Linespoints: Linienstärke, Punktradius, Farbe, Symbol<br>impulses: |
| <code>svg_file</code>         | ( ): Ausgabe als SVG-Datei   |
| <code>t</code>                | (-3, 3): Parameterbereich für parametrische Kurven   |
| <code>title</code>            | ( ): Diagrammtitel   |

|                           |  |
|---------------------------|--|
| <code>transform_xy</code> | (false): kartesische Koordinaten<br>polar_to_xy: Polarkoordinaten<br>spherical_to_xy: Kugelkoordinaten |
| <code>yx_ratio</code>     | Verhältnis zwischen vertikaler und horizontaler Bildabmessung  |
| <code>x, y, z</code>      | min, max: Kurvenbereich oder Anzeigebereich  |
| <code>x/y/zlabel</code>   | ( ): Achsenbeschriftung  |
| <code>x/y/ztics</code>    | ( ): Achsmarken. Startwert oder Startwert und Inkrement oder Startwert, Inkrement und Endwert.         |
| <code>xy_scale</code>     | sx, sy. Skalierfaktoren des Diagrammbereichs   |
| <code>zmin</code>         | minimale z-Koordinate in 3D-Plots  |

| point_type |    |
|------------|----|
| bullet     | 1  |
| circle     | 2  |
| plus       | 3  |
| times      | 4  |
| asterisk   | 5  |
| box        | 6  |
| square     | 7  |
| triangle   | 8  |
| delta      | 9  |
| wedge      | 10 |
| nabla      | 11 |
| diamond    | 12 |
| lozenge    | 12 |

Tabelle G.2.: Werte für `point_type` in `plot2D/plot3D()`

## G.6. Wie dieses Handbuch entsteht

Das Handbuch wurde mit  $\text{\LaTeX}$  erstellt. Das ist ein leistungsfähiges Satzsystem mit besonderen Fähigkeiten bei mathematischen Formeln.

Um nicht den zuweilen unleserlichen  $\text{\LaTeX}$ -Quelltext editieren zu müssen, kommt das Programm  $\text{\LyX}$  zum Einsatz.

Grundsätzlich könnte man auch verschiedene Sprachversionen des Handbuchs erzeugen, der Autor hat jedoch noch nicht herausgefunden, wie das mit  $\text{\LyX}$  effektiv gemacht werden kann.

Die Erzeugung der Screenshots erfolgt teilautomatisch unter Verwendung von Snapshot-Bereichen.

Screenshots mit Callouts gehen mit Greenshot, am einfachsten, indem man eine Auslöseverzögerung einstellt. Also Screenshot anfordern, gewünschte Situation herstellen, Auslösung abwarten.

# Literaturverzeichnis

- (1985):** IEEE Standard for Binary Floating-Point Arithmetic for microprocessor systems. 97, 98
- Assmann, B./Selke, P. (2009):** Technische Mechanik 2. Festigkeitslehre. Oldenbourg 157, 159
- Buchmayr, B. (2002):** Werkstoff- und Produktionstechnik mit Mathcad. Modellierung und Simulation in Anwendungsbeispielen. Springer 19, 181
- Grobstich, P./Strey, G. (2004):** Mathematik für Bauingenieure. Grundlagen, Verfahren und Anwendungen mit Mathcad. Teubner 19
- Henning, G./Jahr, A./Mrowka, U. (2004):** Technische Mechanik mit Mathcad, Matlab und Maple. Vieweg 19
- Krawietz, A. (1997):** Maple V für das Ingenieurstudium. Springer 19
- Liengme, B. V. (2015):** SMATH for Physics. Morgan & Claypool 20
- Werkle, H. (2012):** Mathcad in der Tragwerksplanung. Elektronische Arbeitsblätter für Statik, Stahlbetonbau, Stahlbau und Holzbau. Vieweg+Teubner 19



# Index

## A

Abstand  
    zwischen Bereichen, 63  
acre, 305  
Aktuelles Verzeichnis, 51  
aktuelles Verzeichnis, 51  
Ampere, 309  
Angström, 305  
Ar, 305  
Argumenttrennzeichen, 60  
Arithmetik, 96  
ASCII-Code, 274  
Assign(), 156  
Astronomische Einheit, 305  
Atmosphäre, 308  
Atomare Masseneinheit, 255, 313  
Auflösung, 312  
Avogadro-Zahl, 256, 314

## B

Barn, 305  
bars(), 201  
Beleuchtungsstärke, 311  
Bereich  
    CheckBoxList, 80  
    Combobox, 80  
    ComboBoxList, 80  
    Numeric-up-down, 80  
    RadioButtonList, 81  
    Slider, 81  
Betrag eines Vektors, 139  
Bit, 312  
Blattbereich, 84  
Bohrscher Radius, 305  
Boltzmann-Konstante, 256, 314  
BTU, 309  
Byte, 312

## C

Candela, 311  
CheckBoxList, 80  
col(), 130

cols(), 129  
Combobox, 80  
ComboBoxList, 80  
Coulomb, 310  
CurrentDirectory(), 51

## D

Debye, 311  
Development Tools  
    Math's visual debug, 93  
Dezimaltrennzeichen, 60  
Dezimeter, 305  
Diagrammbereich, 69  
diff(), 146  
DocumentDirectory(), 51  
Dokumentverzeichnis, 51  
dpi, 312  
Druck, 308  
Dyn, 308  
Dynamische Hilfe, 26, 45, 246, 317

## E

Einstellungsverzeichnis, 50  
Elektrische Leitfähigkeit, 311  
Elektrisches Dipolmoment, 311  
Elektronenmasse, 255, 313  
Elementarladung, 255, 313  
Elementindex, 35  
elevation\_grid(), 203  
ellipse(), 203  
Energie, 308  
Erdbeschleunigung, 255, 313  
Erg, 309  
Erweiterungsmanager, 287

## F

Farad, 310  
Farben  
    Interne Grafik, 226  
    Maxima, 221  
FindRoot(), 161  
Flächeneinheiten, 305

- Formelbereich, 25  
for-Schleife, 39  
Frequenz, 306  
Furlong, 305  
Fuß, 305  
**Fußzeile**, 57, 63
- G**  
Gallone, 305  
Gaskonstante, 256, 314  
Gauss (Maßeinheit), 310  
Geschwindigkeit, 307  
GetFolderPath(), 51  
Gigahertz, 306  
Gleichungssysteme  
    mit Maxima lösen, 156  
    Numerische Lösung mit FindRoot(),  
        161  
Gon, 304  
GPC, 333  
Grad, 304  
Grad Celsius, 313  
Grad Fahrenheit, 313  
Grad Rankine, 313  
Grad Reamur, 313  
Gramm, 307  
Gravitationskonstante, 255, 313  
Gray, 311  
Griechische Buchstaben, 28, 91
- H**  
Hektar, 305  
Henry, 310  
Hertz, 306  
Hessian(), 147  
histogram\_description(), 206  
Histogramm, 206  
Hyperlink-Bereich, 77
- I**  
ID, 294  
Identifikationsnummer, 294  
Implizite Schleife, 131  
Indexvektoren, 130  
Induktivität, 310  
Information, 312  
Installation, 21  
Installationsverzeichnis, 50  
Internes Format, 93  
Interpolation, 89
- J**  
Jacobian(), 146  
Jahr, 306  
Joule, 308
- K**  
Kalorie, 309  
Kapazität, 310  
Katal, 312  
Katalytische Aktivität, 312  
Kelvin, 313  
Kilogramm, 307  
Kilohertz, 306  
Kilometer, 305  
Kilopond, 308  
Kilosekunde, 306  
Klammern  
    löschen, 28  
Knoten, 307  
Kopfzeile, 57, 63  
Kraft, 307  
Kreisfrequenz, 306  
Kreuzprodukt, 139
- L**  
label(), 208  
Ladung, 310  
Längeneinheiten, 305  
Leistung, 309  
length(), 129  
Lichtgeschwindigkeit, 255, 313  
Lichtjahr, 305  
Lichtstärke, 311  
Lichtstrom, 311  
Linux, 16  
Liter, 305  
Lorentz-Attraktor, 169  
Lumen, 311  
Lux, 311
- M**  
Magnetische Feldstärke, 310  
Magnetische Flussdichte, 310  
Magnetischer Fluss, 310  
Masse, 307  
Maßeinheiten, 26, 245  
    Auflösung, 312  
    beim Gleichungslösen, 332  
    beim Plotten, 253  
    Beleuchtungsstärke, 311

- Druck, 308
  - Elektrische Leitfähigkeit, 311
  - Elektrisches Dipolmoment, 311
  - Energie, 308
  - Fläche, 305
  - Frequenz, 306
  - Geschwindigkeit, 307
  - in Integralen, 153
  - Induktivität, 310
  - Information, 312
  - Kapazität, 310
  - Katalytische Aktivität, 312
  - Ladung, 310
  - Länge, 305
  - Leistung, 309
  - Lichtstärke, 311
  - Lichtstrom, 311
  - Magnetische Feldstärke, 310
  - Magnetische Flussdichte, 310
  - Magnetischer Fluss, 310
  - Masse, 307
  - Spannung, 309
  - Stoffmenge, 311
  - Strahlendosis, 311
  - Stromstärke, 309
  - Temperatur, 313
  - Viskosität, 312
  - Volumen, 305
  - Widerstand, 311
  - Winkel, 304
  - Winkelgeschwindigkeit, 306
  - Zahlenwerte, 304
  - Zeit, 306
- Matrix
- Lineare Adressierung, 132
- Maxima
- append(), 145
  - apply(), 145
  - args(), 144
  - assume(), 149
  - bars(), 201
  - cylindrical(), 202
  - ellipse(), 203
  - explicit(), 205
  - image(), 207
  - implicit(), 207, 208
  - label(), 208
  - parametric(), 209, 210
  - parametric\_surface(), 210
  - region(), 214
  - spherical(), 196
- Maxima(), 190
- Megagramm, 307
- Megahertz, 306
- Megapond, 308
- Meile, 305
- Meter, 305
- Mikrogramm, 307
- Mikrometer, 305
- Mikrosekunde, 306
- Milligramm, 307
- Milliliter, 305
- Millimeter, 305
- Millisekunde, 306
- Minute, 306
- Modeller-Bereich, 238
- Mol, 311
- Mono, 16
- N**
- Nanometer, 305
- Nanosekunde, 306
- Neugrad, 304
- Neutronenmasse, 255, 313
- Newton, 307
- Numeric-up-down, 80
- O**
- ODE.2(), 170
- Oersted, 310
- Ohm, 311
- P**
- Parsec, 305
- Pascal, 308
- Permeabilität des Vakuums, 256, 314
- Pfund, 307
- Pfund (Kraft), 308
- Picometer, 305
- Picosekunde, 306
- piechart\_description(), 212
- Plancksches Wirkungsquantum, 256, 314
- Plug-in, 291
- Custom Functions, 143, 144, 273
  - Data Exchange, 172
  - Excel IO, 172
  - GPCPlug-in, 333
  - Mathcad File Access, 172
  - Non Linear Solvers, 161

- ODE Solvers, 167, 169
- SQLite Interface, 172
- XLSX Import/Export, 172
- X-Y Plot Region, 169, 229
- Poise, 312
- `polygon()`, 214
- Polygone, 333
- Portables SMath Studio, 322
- Produkt, 136
- Protonenmasse, 255, 313
- Prozentrechnung, 102
- PS, 309
- R**
- rad, 304
- Radiant, 304
- RadioButtonList, 81
- `range()`, 130
- Regression
  - multilineare, 177
- `row()`, 130
- `rows()`, 129
- Rückgängig machen, 25
- S**
- Schieberegler, 81
- Schleife
  - implizit, 131
- Schriftart, 329
- Sektordiagramm, 212
- Sekunde, 306
- Siemens, 311
- Sievert, 311
- Skalarprodukt, 138
- Slider, 81
- Slug, 307
- SMath Studio Portable, 322
- SMath Viewer, 280
- SMathMatrixEditor, 122
- Smoot, 305
- Spannung, 309
- Spatprodukt, 140
- Stoffmenge, 311
- Stokes, 312
- Strahlendosis, 311
- Stromstärke, 309
- stückweise stetige Funktionen, 108
- Stunde, 306
- Suchen, 316
- Summe, 136
- Systemverzeichnisse, 51
- T**
- Tabellenbereich, 73
- Tag, 306
- Temperatur, 249, 313
- Tesla, 310
- Textbereich, 65
- Textindex, 28
- Tonne, 307
- Tonne (Kraft), 308
- Trennlinie, 84
- U**
- Umdrehung, 304
- Unze, 307
- V**
- `vector()`, 216
- `vectorize()`, 134
- Version, 294
- Verzeichnis
  - aktuelles, 51
  - Dokument-, 51
  - Einstellungs-, 50
  - Installations-, 50
- Viskosität, 312
- Volt, 309
- Volumeneinheiten, 305
- W**
- Watt, 309
- Weber, 310
- Widerstand, 311
- Wiederherstellen, 25
- Winkelinheiten, 304
- Winkelgeschwindigkeit, 306
- Y**
- Yard, 305
- Z**
- Zeilenumbruch, 66
- Zeiteinheiten, 306
- Zentimeter, 305
- Zoll, 305
- Zwischenablage, 48