Einführung in *Mathematica* - Teil 1 -

zur Vorlesung *Mathematische Methoden der Physik* im WiSe 2011/12

Norbert Dragon und Michael Flohr (mit Unterstützung von *Martin Paech*)
16. & 18. 11. 2011

Vorbemerkung: *Mathematica* unterscheidet strikt zwischen Groß- und Kleinschreibung. Alle internen Befehle, Symbole, Funktionen, Konstanten usw. beginnen mit einem Großbuchstaben, bspw. **`Plot`**. Zusammensetzungen enthalten auch große Buchstaben innerhalb der Bezeichnung, z. B. **`ParametricPlot`**.

Arithmetik

Elementarer geht es kaum: Selbstverständlich kann *Mathematica* auch, was man von einem Taschenrechner erwartet! Die Eingabe wird jeweils nach Drücken von SHIFT + ENTER am Zeilenende ausgewertet.

```
In[1]:= 5 - 4
Out[1]= 1
```

(Nebenbemerkung: Programmintern wird zur Befehlsverarbeitung eine sog. Infix-Notation verwendet, die in Baumstrukturen beliebig verschachtelt werden kann.)

```
In[2]:= Plus[5, Times[-1, 4]]
Out[2]= 1
```

Als Multiplikationszeichen kann Stern eingegeben werden.

```
In[3]:= 7 * 9
Out[3]= 63
```

Gebräuchlicher ist jedoch schlicht ein Leerzeichen, welches *Mathematica* bei Zahlen automatisch durch ein Kreuz ersetzt.

```
In[4]:= 7 × 9
Out[4]= 63
```

Potenzen werden durch das Symbol ^ erzeugt.

```
In[5]:= 2^64
Out[5]= 18 446 744 073 709 551 616
```

Mathematica bleibt exakt, wo immer möglich.

In[6]:=
```
2 / 12
```
Out[6]=
$$\frac{1}{6}$$

Gleitkommadarstellung des vorausgegangenen Ergebnisses mit dem Befehl **N**: Befehlsargumente werden in eckigen Klammer eingegeben; die hier verwendete %-Variable ist stets mit der letzten (%% mit der vorletzten) Ausgabe "befüllt".

In[7]:=
```
N[%]
```
Out[7]=
```
0.166667
```

Der Dezimalpunkt kennzeichnet reelle Zahlen, die Dezimaldarstellung wird dadurch erzwungen.

In[8]:=
```
2. / 12
```
Out[8]=
```
0.166667
```

Konstanten sind, wie man auf diesen etwas umständlichen Wegen sieht, eingebaut.

In[9]:=
```
Exp[1]
```
Out[9]=
e

In[10]:=
```
2 ArcCos[0]
```
Out[10]=
π

Numerische Berechnungen bzw. Ausgaben können mit beliebiger Genauigkeit erfolgen: Zweites (optionales) Argument von **N** ist die Anzahl der Nachkommastellen. (Man beachte, daß *Mathematica* bei der obigen Ausgabe der Konstanten eine typographisch ansprechende Form - π und e - verwendet; die Eingabe kann hingegen (auch) über den lateinischen Zeichensatz der Tastatur im internen Format - **Pi** und **E** - erfolgen.)

In[11]:=
```
N[Pi, 100]
```
Out[11]=
```
3.14159265358979323846264338327950288419716939937510582097494459230781640628620899·
 8628034825342117068
```

In[12]:=
```
N[E, 1000]
```

Out[12]=
```
2.71828182845904523536028747135266249775724709369995957496696762772407663035354759\
 45713821785251664274274663919320030599218174135966290435729003342952605956307381\
 32328627943490763233829880753195251019011573834187930702154089149934884167509244\
 76146066808226480016847741185374234544243710753907774499206955170276183860626133\
 13845830007520449338265602976067371132007093287091274437470472306969772093101416\
 92836819025515108657463772111252389784425056953696770785449969967946864454905987\
 93163688923009879312773617821542499922957635148220826989519366803318252886939849\
 64651058209392398294887933203625094431173012381970684161403970198376793206832823\
 76464804295311802328782509819455815301756717361332069811250996181881593041690351\
 59888851934580727386673858942287922849989208680582574927961048419844436346324496\
 84875602336248270419786232090021609902353043699418491463140934317381436405462531\
 52096183690888707016768396424378140592714563549061303107208510383750510115747704\
 17189861068739696552126715468895703503540212340784981933432106817012100562788023\
 51930332247450158539047304199577770935036604169973297250886876966403555707162268\
 44716256079882651787134195124665201030592123667719432527867539855894489697096409\
 75459185695638023637016211204774272283648961342251644507818244235294863637214174\
 02388934412479635743702637552944483379980161254922785092577825620926226483262779\
 33386566481627725164019105900491644998289315056604725802778631864155195653244258\
 69829469593080191529872117255634754639644791014590409058629849679128740687050489\
 58586717479854667757573205681288459205413340539220001137863009455606881667400169\
 84205580403363795376452030402432256613527836951177883863874439662532249850654995\
 88623428189970773327617178392803494650143455889707194258639877275471096295374152\
 11151368350627526023264847287039207643100595841166120545297030236472549296669381\
 15137322753645098889031360205724817658511806303644281231496550704751025446501172\
 72115551948668508003685322818315219600373562527944951582841882947876108526398139
```

Mehrere Anweisungen lassen sich in einer Zelle zusammenfassen. Aus Gründen der Übersichtlichkeit sollte mit dieser kompakten Eingabeform jedoch vorsichtig umgegangen werden.

In[13]:=
```
137!
N[%]
```

Out[13]=
```
5 012 888 748 274 991 661 034 926 292 112 253 883 237 205 694 398 754 483 388 962 668 892 510 972 \
 746 226 260 034 675 717 797 072 343 372 830 591 567 227 826 571 884 373 881 355 612 819 314 826 \
 377 917 827 129 740 056 802 397 016 509 378 163 883 274 055 583 382 110 208 000 000 000 000 000 \
 000 000 000 000 000 000
```

Out[14]= 5.01289×10^{234}

Algebra

Definition von Funktionen

Neue Funktionen (Kleinschreibung empfohlen) werden wie folgt definiert, wobei der Doppelpunkt die Auswertung der rechten Seite verhindert, ...

In[15]:=
```
f[x_] := x^6 - 2 x^5 - 30 x^4 + 36 x^3 + 190 x^2 - 36 x - 150
```

... und verhalten sich wie interne Befehle (die stets einen großen Anfangsbuchstaben haben). Bei Funktionsdefinitionen mit `:=` erfolgt keine bestätigende Ausgabe. Man beachte ferner, daß bei einem Produkt aus einer Zahl und einem Symbol ein Leerzeichen zwischen beiden Faktoren automatisch eingefügt wird. Da Variablennamen aus mehreren Buchstaben bestehen dürfen, muß/müssen beim Produkt von Symbolen das/die Leerzeichen manuell eingegeben werden.

Wichtig: An zu "befüllende" Variablen eines Befehles (jede Funktion stellt im Grunde einen Befehl dar), hier **x**, muß das Zeichen _ angehängt werden. Alle nicht deklarierten Symbole werde als Konstanten angesehen, siehe unten.

In[16]:=
```
f[2903]
```

Out[16]=
```
598 110 410 321 877 793 837
```

Zuweisung eines Wertes zu einer Variablen erfolgt mit dem üblichen Gleichheitszeichen. Hier unterdrückt das abschließende Semikolon die Bestätigungsantwort.

In[17]:=
```
r = 17;
```

In[18]:=
```
f[r]
```
Out[18]=
```
19 023 241
```

"Typographische" Eingaben, auch solche von griechischen Buchstaben, Relationssymbolen, Operatoren etc. können aus Paletten (siehe Writing Assistent im Menü Palettes) an der aktuellen Schreibposition eingegeben werden.

In folgendem Beispiel ist μ ein Argument der Funktion λ, ν hingegen nicht.

In[19]:=
```
λ[μ_] := μ^ν
```

In[20]:=
```
λ[2]
```
Out[20]=
2^{ν}

In[21]:=
```
ν = 4
```
Out[21]=
```
4
```

In[22]:=
```
λ[2]
```
Out[22]=
```
16
```

Selbstdefinierte Befehle und Variablen werden wie folgt gelöscht:

In[23]:=
```
Clear[λ, ν]
```

Hier sind nun sowohl μ wie auch ν Funktionsargumente.

In[24]:=
```
λ[μ_, ν_] := μ^ν
```

In[25]:=
```
λ[2, 4]
```
Out[25]=
```
16
```

Symbolische Argumente - also solche, die Variablen ohne zugewiesenen numerischen Wert enthalten - werden eingesetzt und der Gesamtausdruck ggf. vereinfacht und sortiert.

In[26]:=
```
λ[(ξ - 2), 3]
```
Out[26]=
$(-2+\xi)^3$

In[27]:=
```
f[1 / (2 t)]
```
Out[27]=
$$-150+\frac{1}{64\,t^6}-\frac{1}{16\,t^5}-\frac{15}{8\,t^4}+\frac{9}{2\,t^3}+\frac{95}{2\,t^2}-\frac{18}{t}$$

Auflösen von Gleichungen

Nullstellenfindung, hier analytisch möglich: Das Symbol **==** bezeichnet die Gleichheit zweier Ausdrücke.

In[28]:=
```
Solve[x^2 + p x + q == 0, x]
```

Out[28]=
$$\left\{\left\{x \to \frac{1}{2}\left(-p - \sqrt{p^2 - 4\,q}\right)\right\}, \left\{x \to \frac{1}{2}\left(-p + \sqrt{p^2 - 4\,q}\right)\right\}\right\}$$

Mathematica liefert mehrere Ergebnisse stets als Liste, eingefaßt in geschweifte Klammern, zurück; genauergesagt handelt es sich bei voriger Ausgabe um eine (zweielementige) Liste aus (einelementigen) Listen.

Der Zugriff auf ein Listenelement erfolgt über dessen Nummer (beginnend mit 1) behelfs des Zugriffsbefehls aus **[[** und **]]**, ...

In[29]:=
```
%[[2]]
```

Out[29]=
$$\left\{x \to \frac{1}{2}\left(-p + \sqrt{p^2 - 4\,q}\right)\right\}$$

... der sich mehrfach aneinandergereiht werwenden läßt.

In[30]:=
```
%%[[2]][[1]]
```

Out[30]=
$$x \to \frac{1}{2}\left(-p + \sqrt{p^2 - 4\,q}\right)$$

Der Pfeil → symbolisiert eine Transformationsregel; dieses in *Mathematica* wichtige Zeichen wird bei der Eingabe automatisch aus **->** erzeugt.

Der Ersetzungsoperator

Wird nur die rechte Seite obiger Lösung benötigt, muß mit dem Ersetzungsoperator **/.** "extrahiert" werden: **x** wird hier durch das ersetzt, wozu **x** im vorherigen Ergebnis "transformiert" wird - oder schlichter gesagt: Die Transformationsregel wird für **x** explizit ausgeführt. Allgemein erwartet der Ersetzungsoperator links stets einen Ausdruck und rechts eine Transformationsregel.

In[31]:=
```
x /. %
```

Out[31]=
$$\frac{1}{2}\left(-p + \sqrt{p^2 - 4\,q}\right)$$

Nullstellenfindung, analytisch nicht möglich. *Mathematica* gibt keinen Fehler, sondern die sechs Wurzeln als "allgemeine Lösung" in Form der Vorstufe zur numerischen Lösung (gekennzeichnet mit **Root**) aus.

In[32]:=
```
Solve[f[x] == 0, x]
```

Out[32]=
```
{{x → Root[-150 - 36 #1 + 190 #1^2 + 36 #1^3 - 30 #1^4 - 2 #1^5 + #1^6 &, 1]},
 {x → Root[-150 - 36 #1 + 190 #1^2 + 36 #1^3 - 30 #1^4 - 2 #1^5 + #1^6 &, 2]},
 {x → Root[-150 - 36 #1 + 190 #1^2 + 36 #1^3 - 30 #1^4 - 2 #1^5 + #1^6 &, 3]},
 {x → Root[-150 - 36 #1 + 190 #1^2 + 36 #1^3 - 30 #1^4 - 2 #1^5 + #1^6 &, 4]},
 {x → Root[-150 - 36 #1 + 190 #1^2 + 36 #1^3 - 30 #1^4 - 2 #1^5 + #1^6 &, 5]},
 {x → Root[-150 - 36 #1 + 190 #1^2 + 36 #1^3 - 30 #1^4 - 2 #1^5 + #1^6 &, 6]}}
```

In[33]:=
```
N[%]
```

Out[33]=
```
{{x → -4.42228}, {x → -2.14285}, {x → -0.937347},
 {x → 0.972291}, {x → 3.35802}, {x → 5.17217}}
```

Numerische Nullstellensuche auf direktem Wege mit dem Ergebnis auf zehn Nachkommastellen, "natürlich" in Listenform:

In[34]:=
```
NSolve[f[x] == 0, x, 10]
```

Out[34]=
```
{{x → -4.422280787}, {x → -2.142852331}, {x → -0.9373473230},
 {x → 0.9722910610}, {x → 3.358016638}, {x → 5.172172742}}
```

Einige weitere algebraische Manipulationen

In[35]:=
```
x y^6
```

Out[35]=

$x\, y^6$

Anwendung (`/.`) einer neuen Transformationsregel (**y→v+w**) auf die vorherige Eingabe:

In[36]:=
```
% /. y → v + w
```

Out[36]=

$(v + w)^6\, x$

Ausmultiplizieren ...

In[37]:=
```
Expand[%]
```

Out[37]=

$v^6 x + 6 v^5 w x + 15 v^4 w^2 x + 20 v^3 w^3 x + 15 v^2 w^4 x + 6 v w^5 x + w^6 x$

... und wieder zusammenfassen:

In[38]:=
```
Simplify[%]
```

Out[38]=

$(v + w)^6\, x$

Eine weitere oft nützliche Vereinfachung kann das Faktorisieren sein.

In[39]:=
```
Factor[3 628 800 - 10 628 640 x + 12 753 576 x^2 - 8 409 500 x^3 +
  3 416 930 x^4 - 902 055 x^5 + 157 773 x^6 - 18 150 x^7 + 1320 x^8 - 55 x^9 + x^10]
```

Out[39]= $(-10+x)(-9+x)(-8+x)(-7+x)(-6+x)(-5+x)(-4+x)(-3+x)(-2+x)(-1+x)$

Reihenentwicklungen

Taylor- bzw. Potenzreihenentwicklung der Exponentialfunktion um 0 bis einschließlich 7. Ordung

In[40]:=
```
Series[Exp[x], {x, 0, 7}]
```

Out[40]= $$1+x+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+\frac{x^5}{120}+\frac{x^6}{720}+\frac{x^7}{5040}+O[x]^8$$

"Abschneiden" des symbolischen Restterms

In[41]:=
```
Normal[%]
```

Out[41]= $$1+x+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+\frac{x^5}{120}+\frac{x^6}{720}+\frac{x^7}{5040}$$

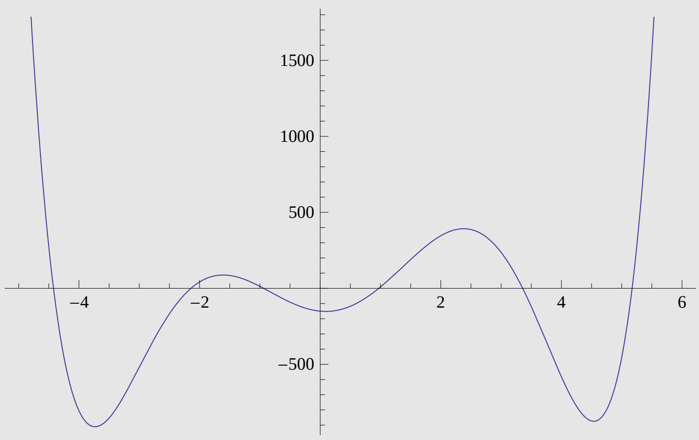
Analysis

Graphische Darstellung von Funktionen

Mathematica wählt viele Darstellungsparameter recht vernünftig.

In[42]:=
```
Plot[f[x], {x, -5, 6}]
```

Out[42]=

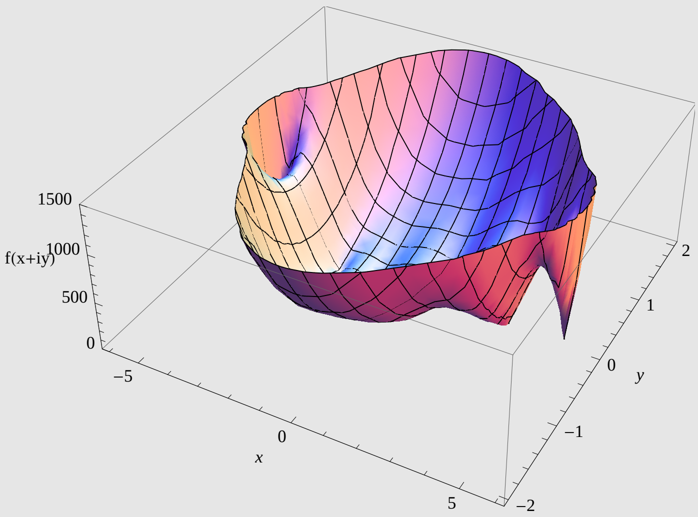


Man kann auch sehr schön drei-dimensionale Plots erstellen.

In[43]:=

```
Plot3D[Abs[f[x + I y]], {x, -6, 6}, {y, -2, 2}, PlotRange → {0, 1500},
 AxesLabel → {x, y, "f(x+iy)"}, ClippingStyle -> None]
```

Out[43]=



Ach ja, bei komplexen Zahlen bezeichnet `I` die imaginäre Einheit:

In[44]:=

```
I^2
```

Out[44]= -1

Differenzieren

Erste Ableitung: Die Angabe einer 1 im Befehl ist optional. (Bei einer solchen Funktionsdefinition darf kein Doppelpunkt vor dem Gleichheitszeichen stehen, da ja der Differentiationsbefehl ausgeführt werden soll.)

In[45]:=

```
g[x_] = D[f[x], {x, 1}]
```

Out[45]= $-36 + 380\,x + 108\,x^2 - 120\,x^3 - 10\,x^4 + 6\,x^5$

(Ein anspruchsvolleres Beispiel)

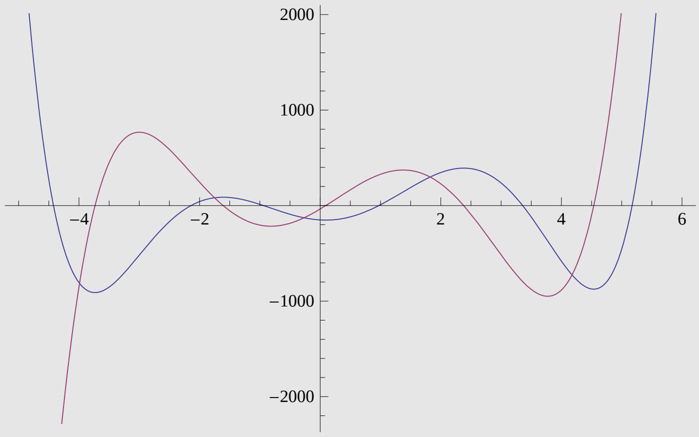
In[46]:=

```
D[Sqrt[x^3 Exp[4 x] Sin[x]], x]
```

Out[46]= $\left(e^{4x}\,x^3\,\mathrm{Cos}[x] + 3\,e^{4x}\,x^2\,\mathrm{Sin}[x] + 4\,e^{4x}\,x^3\,\mathrm{Sin}[x]\right) \Big/ \left(2\sqrt{e^{4x}\,x^3\,\mathrm{Sin}[x]}\right)$

Darstellung beider Funktionen

In[47]:=
```
Plot[{f[x], g[x]}, {x, -5, 6}]
```

Out[47]=


Integration

Stammfunktion, analytisch

In[48]:=
```
e[x_] = Integrate[f[x], x]
```

Out[48]=

$$-150\,x - 18\,x^2 + \frac{190\,x^3}{3} + 9\,x^4 - 6\,x^5 - \frac{x^6}{3} + \frac{x^7}{7}$$

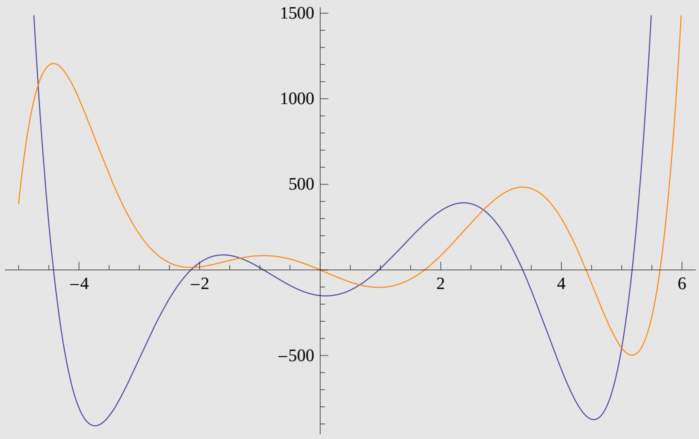
(Ein anspruchsvolleres Beispiel)

In[49]:=
```
Integrate[(2 x + 3) / (x^3 + x^2 - 2 x), x]
```

Out[49]=

$$\frac{5}{3}\,\mathrm{Log}[1 - x] - \frac{3\,\mathrm{Log}[x]}{2} - \frac{1}{6}\,\mathrm{Log}[2 + x]$$

In[50]:=
```
Plot[{f[x], e[x]}, {x, -5, 6}, PlotStyle → {Automatic, Orange}]
```

Out[50]=


Nochmals die Nullstellen, die in einer Liste gespeichert werden

In[51]:=
```
n = NSolve[f[x] == 0, x]
```
Out[51]=
```
{{x → -4.42228}, {x → -2.14285}, {x → -0.937347},
 {x → 0.972291}, {x → 3.35802}, {x → 5.17217}}
```

Zugriff auf das erste Listenelement

In[52]:=
```
n[[1]]
```
Out[52]=
```
{x → -4.42228}
```

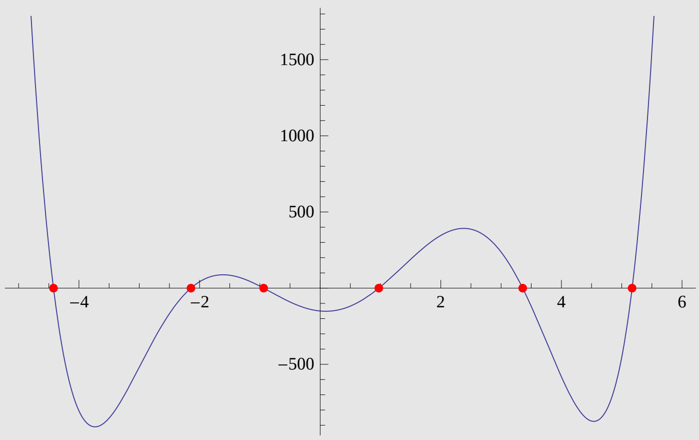
Extraktion und Zuweisung der ersten und zweiten Nullstelle mithilfe des bekannten Ersetzungsoperators. (Die letzte Listenebene, welche als Element bloß die Transformationsregel enthält, wird automatisch entfernt.)

In[53]:=
```
a = x /. n[[1]]
```
Out[53]=
```
-4.42228
```

In[54]:=
```
b = x /. n[[2]];
```

Eine weitere Verwendung der Nullstellenliste: Der Ersetzungsoperator läßt sich "überladen", d.h. praktisch: Er wirkt nicht nur, wie oben genutzt, auf ein ausgewähltes, sondern auf alle Elemente einer Liste und gibt eine entsprechend lange Liste zurück, hier von (zweidimensionalen kartesischen) Koordinaten, die ihrerseits in Listenform **{x,y}** mit **y=0** notiert werden.

In[55]:=
```
Plot[f[x], {x, -5, 6}, Epilog → {Red, PointSize[Medium], Point[{x, 0} /. n]}]
```
Out[55]=



Epilog "ist" eine Liste von Zeichenbefehlen, die nach dem Plotten der Funktion(en) ausgeführt werden - und damit der Schlüssel zu vielen graphischen "Spielereien". So zeichnet **Point** z. B. einen Punkt, (optional) mit **PointSize** definierter Größe und angegebener Farbe.

Numerische Auswerung eines bestimmten Integrals

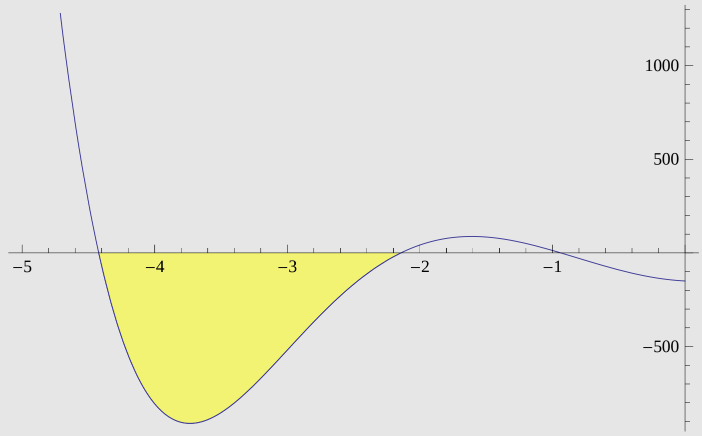
In[56]:=
```
NIntegrate[f[x], {x, a, b}]
```
Out[56]=
```
-1191.15
```

Alternative Möglichkeit zur Überlagerung von zwei Graphen - notwendig, um die Fläche unter der

Abszisse nur im Integrationsintervall einzufärben. (Mit `Directive` werden Stilparameter zusammengefaßt; `Opacity` bestimmt die Transparenz der Farbe.)

In[57]:=
```
Show[Plot[f[x], {x, -5, 0}], Plot[f[ξ], {ξ, a, b},
  Filling → Axis, FillingStyle → Directive[Yellow, Opacity[0.5]]]]
```

Out[57]=



Lineare Algebra

Vektoren und Matrizen

Die Verarbeitung von Vektoren und Matrizen erfolgt vollständig über Listen. Im Falle eines Vektors ist jede Komponente einfach ein Listenelement, ...

In[58]:=
```
b = {9, -2, 7}
```

Out[58]=
```
{9, -2, 7}
```

... im Falle einer Matrix ist jede Zeile, als Vektor notiert, ein Listenelement.

In[59]:=
```
p = {{2, 1, 3}, {1, -2, 1}, {3, 2, 2}}
```

Out[59]=
```
{{2, 1, 3}, {1, -2, 1}, {3, 2, 2}}
```

(`MatrixForm` führt zu einer ansprechenderen Ausgabe.)

In[60]:=
```
MatrixForm[%]
```

Out[60]//MatrixForm=

$$\begin{pmatrix} 2 & 1 & 3 \\ 1 & -2 & 1 \\ 3 & 2 & 2 \end{pmatrix}$$

In[61]:=
```
Clear[a, x, y, z]
```

In[62]:=
```
c = {x, y, z};
```

Folgende Operationen kann man (beinahe) erraten ...

Konstante mal Vektor

In[63]:=
```
a b
```
Out[63]=
```
{9 a, -2 a, 7 a}
```

Für das Skalarprodukt gibt es zwei Eingabevarianten, ...

In[64]:=
```
b.c
```
Out[64]=
```
9 x - 2 y + 7 z
```

In[65]:=
```
Dot[b, c]
```
Out[65]=
```
9 x - 2 y + 7 z
```

... ebenso eine typographisch schönere für das Vektorprodukt aus dem Abschnitt Typesetting des Writing Assistents.

In[66]:=
```
Cross[b, c]
```
Out[66]=
```
{-7 y - 2 z, 7 x - 9 z, 2 x + 9 y}
```

In[67]:=
```
b × c
```
Out[67]=
```
{-7 y - 2 z, 7 x - 9 z, 2 x + 9 y}
```

(`Cross` und `Dot` sind ebenso, wie eingangs bemerkten `Plus` und `Times`, Infix-Befehle - hier allerdings in sinnvollerer Anwendung.)

In[68]:=
```
Det[p]
```
Out[68]=
```
13
```

In[69]:=
```
MatrixForm[Transpose[p]]
```
Out[69]//MatrixForm=

$$\begin{pmatrix} 2 & 1 & 3 \\ 1 & -2 & 2 \\ 3 & 1 & 2 \end{pmatrix}$$

(Die Funktion `MatrixForm` kann auch als nachgestellte Formatierungsanweisung verwendet werden.)

In[70]:=
```
Inverse[p] // MatrixForm
```
Out[70]//MatrixForm=

$$\begin{pmatrix} -\frac{6}{13} & \frac{4}{13} & \frac{7}{13} \\ \frac{1}{13} & -\frac{5}{13} & \frac{1}{13} \\ \frac{8}{13} & -\frac{1}{13} & -\frac{5}{13} \end{pmatrix}$$

Das Produkt zweier Matrizen wird mit dem (überladenen) Punktoperator ausgeführt, der auch beim Skalarprodukt Verwendung findet.

In[71]:=
```
MatrixForm[Inverse[p].p]
```
Out[71]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Zugriffe auf einzelne Elemente

Ein allgemeiner Vektor mit einer beliebigen Dimensionalität läßt sich wie folgt erzeugen:

In[72]:=
```
t = Array[x, 3]
```
Out[72]=
```
{x[1], x[2], x[3]}
```

Der Zugriff auf eine Komponente erfolgt wie gehabt.

In[73]:=
```
t[[2]]
```
Out[73]=
```
x[2]
```

In[74]:=
```
t[[2]] = 8;
```

In[75]:=
```
t
```
Out[75]=
```
{x[1], 8, x[3]}
```

An folgendem Beispiel wird der Unterschied zwischen dem Zuweisungsoperator und dem Ersetzungsoperator besonders deutlich:

In[76]:=
```
t /. x[3] → 9
```
Out[76]=
```
{x[1], 8, 9}
```

In[77]:=
```
t
```
Out[77]=
```
{x[1], 8, x[3]}
```

Oben für Vektoren gezeigtes gilt auch für Matrizen (usw.):

In[78]:=
```
u = Array[d, {4, 4}]
```
Out[78]=
```
{{d[1, 1], d[1, 2], d[1, 3], d[1, 4]}, {d[2, 1], d[2, 2], d[2, 3], d[2, 4]},
 {d[3, 1], d[3, 2], d[3, 3], d[3, 4]}, {d[4, 1], d[4, 2], d[4, 3], d[4, 4]}}
```

In[79]:=
```
MatrixForm[u]
```
Out[79]//MatrixForm=

$$\begin{pmatrix} d[1,1] & d[1,2] & d[1,3] & d[1,4] \\ d[2,1] & d[2,2] & d[2,3] & d[2,4] \\ d[3,1] & d[3,2] & d[3,3] & d[3,4] \\ d[4,1] & d[4,2] & d[4,3] & d[4,4] \end{pmatrix}$$

Alternativ:

In[80]:=
```
v = Table[d_{i,j}, {i, 4}, {j, 4}]
```
Out[80]=

$\{\{d_{1,1}, d_{1,2}, d_{1,3}, d_{1,4}\}, \{d_{2,1}, d_{2,2}, d_{2,3}, d_{2,4}\}, \{d_{3,1}, d_{3,2}, d_{3,3}, d_{3,4}\}, \{d_{4,1}, d_{4,2}, d_{4,3}, d_{4,4}\}\}$

In[81]:=
```
MatrixForm[v]
```
Out[81]//MatrixForm=

$$\begin{pmatrix} d_{1,1} & d_{1,2} & d_{1,3} & d_{1,4} \\ d_{2,1} & d_{2,2} & d_{2,3} & d_{2,4} \\ d_{3,1} & d_{3,2} & d_{3,3} & d_{3,4} \\ d_{4,1} & d_{4,2} & d_{4,3} & d_{4,4} \end{pmatrix}$$

In[82]:=
```
MatrixForm[Array[e, {2, 2, 2}]]
```
Out[82]//MatrixForm=

$$\begin{pmatrix} \begin{pmatrix} e[1,1,1] \\ e[1,1,2] \end{pmatrix} & \begin{pmatrix} e[1,2,1] \\ e[1,2,2] \end{pmatrix} \\ \begin{pmatrix} e[2,1,1] \\ e[2,1,2] \end{pmatrix} & \begin{pmatrix} e[2,2,1] \\ e[2,2,2] \end{pmatrix} \end{pmatrix}$$

Man kann die Matrizen automatisch mit den Werten von Funktionen befüllen lassen:

In[83]:=
```
Table[h[i_, j_] = i + j, {i, 4}, {j, 4}] // MatrixForm
```
Out[83]//MatrixForm=

$$\begin{pmatrix} 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 7 \\ 5 & 6 & 7 & 8 \end{pmatrix}$$

Entnahme einer Zeile, also eines Elements der äußeren Liste

In[84]:=
```
u[[3]]
```
Out[84]=
```
{d[3, 1], d[3, 2], d[3, 3], d[3, 4]}
```

Entnahme einer Komponente des Zeilenvektors

In[85]:=
```
%[[2]]
```
Out[85]=
```
d[3, 2]
```

Praktisch: Der nun hinlänglich bekannte Listenzugriff kann auch direkt mit Zeilen - und Spaltenindex durchgeführt werden.

In[86]:=
```
u[[3, 2]]
```
Out[86]=
```
d[3, 2]
```

(äquivalente Langform)

In[87]:=
```
u[[3]][[2]]
```
Out[87]=
```
d[3, 2]
```

Und wie kommt man an eine Spalte? Indem man für jede Zeile ein Element rauspickt. Das sagt man *Mathematica* damit, dass man für den Zeilenbereich einfach **All** schreibt:

In[88]:=
```
u[[All, 2]]
```
Out[88]=
```
{d[1, 2], d[2, 2], d[3, 2], d[4, 2]}
```

Lösung von Gleichungssystemen

Automatisch ...

In[89]:=
```
LinearSolve[p, b]
```
Out[89]=
```
{-1, 2, 3}
```

... oder manuell, in diesem Sonderfall "direkt"

In[90]:=
```
Inverse[p].b
```
Out[90]=
```
{-1, 2, 3}
```