

## Konstanten, Operatoren und Funktionen von DERIVE (Auswahl für den Unterricht)

### Konstanten:

#e - Basis des natürlichen Logarithmus (=2.71828...) (Strg-E)  
 #i - Quadratwurzel von -1 (Strg-I)  
 pi - Flächeninhalt des Einheitskreises (=3.14159...) (Strg-P)  
 inf - plus Unendlich (Strg-0)  
 deg - Radiant pro Grad (=pi/180)  
 unit\_circle - beliebiger Punkt am Einheitskreis  
 euler\_gamma := Eulersche Konstante (=0.577215...)

### Operatoren

$u + v$  : u plus v ;  $-u$  : minus u ;  $u - v$  : u minus v ;  $u * v$  : u mal v ;  $u / v$  : u dividiert durch v  
 $u \wedge v$  u hoch v  
 $u \%$  u Prozent  
 $u!$  u Fakultät

### Relationen

$u = v$  u gleich v  
 $u \neq v$  u ungleich v  
 $u < v$  u kleiner als v  
 $u \leq v$  u kleiner oder gleich v  
 $u > v$  u größer als v  
 $u \geq v$  u größer oder gleich v

### Boolesche Operatoren und Funktionen

NOT p ist genau dann true, falls p falsch ist  
 p AND q ist genau dann true, falls sowohl p als auch q wahr ist  
 p OR q ist genau dann true, falls entweder p oder q wahr ist  
 p XOR q ist genau dann true, falls p oder q wahr ist, aber nicht beide gleichzeitig  
 p IMP q ist genau dann true, falls p falsch oder q wahr ist  
 TRUTH\_TABLE (p,q,...,u,v,...) p,q,... sind Variable und u, v,... Boolesche Ausdrücke

### Definieren und Löschen (Freigeben)

<b>x := u</b>	weist der Variable x den Wert u zu
<b>x :=</b>	<b>macht aus x eine beliebige Variable (Freigabe)</b>
<b>f(x,y,...) := u</b>	definiert eine Funktion f in Abhängigkeit von den Variablen x,y,... mit dem Funktionsterm u
<b>f(x,y,...) :=</b>	<b>macht aus f eine beliebige Funktion (Freigabe)</b>

### Funktionen für das Ausrechnen und Umwandeln

APPROX (u)	liefert einen Näherungswert von u bzgl. der aktuellen Genauigkeit
APPROX (u,n)	liefert einen Näherungswert von u mit n-stelliger Genauigkeit
EXPAND (u,amount,x,y,..)	multipliziert u(x,y) mit dem Ausmaß amount bzgl. der Variablen x,y,... aus
FACTOR (u,amount,x,y,..)	zerlegt u(x,y) mit dem Ausmaß amount bzgl. der Variablen x,y,... in Faktoren
ITERATES (u,x,x0)	liefert einen Vektor [x0,u(x0),u(u(x0)),...] bis sich ein Element wiederholt
ITERATES (u,x,x0,n)	returns the first n+1 elements of the vector [x0,u(x0),u(u(x0)),...]
ITERATE (u,x,x0)	liefert das erste Element der Folge x0, u(x0), u(u(x0)), ..., das sich wiederholt
ITERATE (u,x,x0,n)	liefert das n+1-te Element der Folge x0, u(x0), u(u(x0)),
ITERATE (u,[x1,x2,..],[y1,y2,..])	liefert 1. Elem. der Folge [x01,x02,..],u(y1,y2,..), das sich wiederholt
IF (r)	liefert 1, falls die Relation r wahr ist, und 0, falls r falsch ist
IF (r,t,f)	liefert den Ausdruck t, falls die Relation r wahr ist, und f, falls r falsch ist
IF (r,t,f,u)	liefert den Ausdruck t, falls die Relation r wahr ist, den Ausdruck f, falls die Relation r falsch ist, und u, falls der Wahrheitswert von r unbekannt ist.

**Funktionen zum Aufspalten von Ausdrücken**

NUMERATOR (u)	syntaktischer Zähler von u
DENOMINATOR (u)	syntaktischer Nenner von u
QUOTIENT (u,v)	Quotient bei der Polynomdivision u durch v
REMAINDER (u,v)	Rest bei der Polynomdivision u durch v
POLY_GCD (u,v)	GGT bei Polynomen u und v
TERMS (u)	Vektor von syntaktischen Summanden von u
FACTORS (u)	Vektor von syntaktischen Faktoren von u
VARIABLES (u)	Vektor der freien Variablen in u
LHS (r)	die linke Seite einer Relation r
RHS (r)	die rechte Seite einer Relation r

**Wurzeln**

SQRT(x) - Quadratwurzel von x;  $x^{(1/n)}$  - n-te Wurzel aus x

**Exponentialfunktionen**

EXP(x) - e hoch x (e mit Strg-E);  $a^x$  - a hoch x

**Logarithmische Funktionen**

LN(x)	Logarithmus von x zur Basis e
LOG(x)	Logarithmus von x zur Basis 10
LOG(x,w)	Logarithmus von x zur Basis w

**Trigonometrische Funktionen**

SIN(x*deg)	Sinus von x in Grad
SIN(x)	Sinus von x in rad
COS(x)	Cosinus von x in rad
TAN(x)	Tangens von x in rad
COT(x)	Cotangens von x in rad

**Trigonometrische Umkehrfunktionen**

ATAN(x)	Arcustangens von x in rad
ACOT(x)	Arcuscotangens von x in rad
ASIN(x)	Arcussinus von x in rad
ACOS(x)	Arcuscosinus von x in rad

**Funktionen aus der Analysis**

LIM (u,x,a)	Grenzwert von u(x) für x gegen a
LIM (u,x,a,1)	Grenzwert von u(x) für x gegen a von oben
LIM (u,x,a,-1)	Grenzwert von u(x) für x gegen a von unten
DIF (u,x)	Ableitung von u(x) bzgl. x
DIF (u,x,n)	Ableitung n-ter Ordnung von u(x) bzgl. x
TAYLOR (u,x,a,n)	Taylorreihenentw. n-ter Ordng von u(x) bei x=a
INT (u,x)	Stammfunktion von u(x) bzgl. x
DIF (u,x,-n)	Stammfunktion n-ter Ordnung von u(x) bzgl. x
INT (u,x,a,b)	bestimmtes Integral von u(x) von x=a bis b
SUM (u,k)	Summe über u(k) bzgl. k
SUM (u,k,m,n)	bestimmte Summe von u(k) für k=m bis n
SUM (u,k,v)	Summe über u(k) für die Elemente k eines Vektors v
SUM (v)	Summe über die Elemente eines Vektors v
PRODUCT (u,k)	Produkt von u(k) bzgl. k
PRODUCT (u,k,m,n)	bestimmtes Produkt von u(k) für k=m bis n
PRODUCT (u,k,v)	Produkt von u(k) für die Elemente k eines Vektors v
PRODUCT (v)	Produkt über die Elemente eines Vektors v

**Stückweise stetige Funktionen**

ABS(x)	Absolutbetrag von x
SIGN(x)	Signum von x
MAX(x, y, ...)	Maximum der Argumente
MIN(x, y, ...)	Minimum der Argumente
STEP(x)	liefert 1, falls $x > 0$ , aber 0 für $x < 0$
CHI(a, x, b)	liefert 1, falls $a < x < b$ , aber 0, falls $x < a$ oder $x > b$
FLOOR (m, n)	größte ganze Zahl $\leq m/n$
FLOOR (m)	größte ganze Zahl kleiner oder gleich m
MOD (m, n)	m modulo n (nicht negativer Rest von m/n)
MOD (m)	Bruchteil von m
MODS (m, n)	symmetrisch m modulo n

**Erzeugen von Vektoren und Matrixen**

[x1, x2, ..., xn]	n_elementiger Vektor
[[a, b], [c, d]]	2x2 Matrix
VECTOR (u,k,n)	Vektor über u(k) für k von 1 bis n mit der Schrittweite 1
VECTOR (u,k,m,n)	Vektor über u(k) für k von m bis n mit der Schrittweite 1
VECTOR (u,k,m,n,s)	Vektor über u(k) für k von m bis n mit der Schrittweite s
VECTOR (u,k,v)	Vektor über u(k) für die Elemente k aus dem Vektor v
IDENTITY_MATRIX (n)	nxn Einheitsmatrix
DIMENSION (v)	Anzahl der Elemente eines Vektors v
ABS (v)	Länge des Vektors v

**Vektoroperatoren**

$x \cdot v$	Zahl x mal Vektor (skalare Multiplikation)
$v + w$	Vektoraddition bei gleicher Dimension
$v - w$	Vektorsubtraktion bei gleicher Dimension
$v \cdot w$	Skalarprodukt der Vektoren v und w
CROSS (v,w)	Kreuzprodukt der Vektoren v und w

**Funktionen zum Bearbeiten von Vektoren**

v SUB n	n_tes Element des Vektors v
ELEMENT(v,n)	n-tes Element des Vektors v
A SUB j SUB k	Element in der j-ten Zeile und k-ten Spalte der Matrix A
ELEMENT(A,j,k)	Element in der j-ten Zeile und k-ten Spalte der Matrix A
APPEND (v, w)	Hängt die Elemente des Vektors w an den Vektor v an
DELETE_ELEMENT (v,n)	löscht das n-te Element des Vektors v
INSERT_ELEMENT (u,v,n)	fügt u vor dem n-ten Element des Vektors v ein
REPLACE_ELEMENT (u,v,n)	ersetzt das n-te Element des Vektors v durch u
REVERSE_VECTOR (v)	ordnet die Elemente des Vektors v in umkehrter Reihenfolge
SELECT (u,k,m,n,s)	Vektor aus jenen k von m bis n mit der Schrittweite s, für die u(k) wahr ist
SELECT (u,k,v)	Vektor aus jenen Elementen k aus v, für die u(k) wahr ist

**Matrixoperationen**

A . B	Produkt von Matrixen
A <sup>^</sup>	transponierte Matrix
DET (A)	Determinante von A
TRACE (A)	Spur von A (Summe der Diagonalelemente)
A <sup>-1</sup>	Inverse Matrix von A
ROW_REDUCE (A)	Einheitsform von A
ROW_REDUCE (A, B)	Einheitsform von A erweitert mit B
CHARPOLY (A, mu)	characteristisches Polynom von A mit der Variable mu
EIGENVALUES (A, mu)	Eigenwerte von A

### Funktionen aus der Zahlentheorie

GCD (m, n, ...)	größter gemeinsamer Teiler von m, n,...
LCM (m, n, ...)	kleinstes gemeinsames Vielfaches von m, n,...
PRIME (n)	true, falls n wahrscheinlich eine Primzahl ist, sonst false
PRIME (n,k)	führt k Iterationen eines probabilistischen Primzahlentests durch (Voreinstellung k=5)
NEXT_PRIME (n)	nächste Primzahl größer als n
NEXT_PRIME (n,k)	führt k Iterationen eines probabilistischen Primzahlentest durch, bevor die nächste Primzahl geliefert wird (Voreinstellung k=5)

### Funktionen aus der Wahrscheinlichkeitstheorie

x!	x Fakultät
GAMMA(x)	Gammafunktion von x
PERM (x,w)	Variationen von x Elementen zur Klasse w ohne Wiederholung
COMB (x,w)	Kombinationen von x Elementen zur Klasse w ohne Wiederholung
RANDOM (n)	falls n>1, eine ganzzahlige Zufallszahl aus dem Intervall [0,n)
RANDOM (n)	falls n=1, eine Zufallszahl aus dem Intervall [0,1)
RANDOM (n)	falls n<1, Initialisieren auf n
RANDOM (n)	falls n=0, Initialisieren auf die aktuelle Systemzeit

### Funktionen aus der Statistik

AVERAGE (x1, ..., xn)	arithmetisches Mittel
RMS (x1, ..., xn)	Wurzel des Mittelwerts der Quadrate
VAR (x1, ..., xn)	Varianz
STDEV (x1, ..., xn)	Standardabweichung
FIT (v, A)	Ausgleichsfunktion von $v=[x,f(x)]$ bzgl. der Datenmatrix A, wobei f eine Funktion ist, die linear von Parametern abhängt.

### Funktionen in Zusatzdateien

Es gibt Tausende, ich habe nur einige ausgewählt, die evtl. im Schulunterricht gebraucht werden könnten.

<b>VECTOR.MTH</b>	Zusätzliche Vektor und Matrixfunktionen. v und w seien Vektoren, A und B Matrizen	
i_	= [1, 0, 0]	Einheitsvektor für die x-Achse
j_	= [0, 1, 0]	Einheitsvektor für die y-Achse
k_	= [0, 0, 1]	Einheitsvektor für die z-Achse
OUTER (v,w)	Äußeres Produkt von v und w	
KRONECKER (i,j)	Kroneckersymbol	
ADJOIN_ELEMENT (e,v)	fügt e vorne an v an	
APPEND_COLUMNS (A,B)	fügt die Spalten von A und B aneinander	
MINOR (A,i,j)	löscht die i-te Zeile und die j-te Spalte aus der Matrix A	
SWAP_ELEMENTS (v,i,j)	vertauscht die Elemente i und j des Vektors v	
SCALE_ELEMENT (v,i,s)	multipliziert das i-te Element von v mit s	
SUBTRACT_ELEMENTS (v,i,j,s)	subtrahiert das j-te Element s mal vom i-ten Element des Vektors v	
FORCE0 (A,i,j,p)	macht das Element ij von A gleich 0 durch geeignete Subtraktion der Pivotzeile p	
PIVOT (A,i,j)	macht die Elemente der Spalte j unterhalb der Zeile i gleich 0 durch Pivotsieren	
MATPROD (A,B,i,j)	Element i,j des Produkts von A und B	
COFACTOR (A,i,j)	Zähler des Elements i,j der inversen Matrix von A	
ADJOINT (A)	Adjungierte Matrix zur quadratischen Matrix A	
RANK (A)	Rang der Matrix A	
EXACT_EIGENVECTOR (A,mu)	Eigenvektor von $A < 4 \times 4$ für den Eigenwert mu	

### noch Funktionen in Zusatzdateien

<b>DIF_APPS.MTH</b>	Anwendungen der Differentiation
CURVATURE (y,x)	= Krümmung von y(x)
CENTER_OF_CURVATURE (y,x)	= Krümmungsmittelpunkt von y(x)
TANGENT (y,x,x0)	= Tangente an y(x) an der Stelle x=x0
PERPENDICULAR (y,x,x0)	= Normale zu y(x) an der Stelle x=x0
OSCULATING_CIRCLE (y,x,theta)	= Schmiegekreis an y(x), mit Winkelparameter Theta
PARA_DIF (v,t,n)	= n-te Ableitung von v bzgl. t, wobei v=[x(t),y(t)]
PARA_CURVATURE (v,t)	= Krümmung von v
PARA_CENTER_OF_CURVATURE (v,t)	= Krümmungsmittelpunkt von v
PARA_TANGENT (v,t,t0,x)	= Tangente an v an der Stelle t = t0, in Abhängigkeit von x
PARA_PERPENDICULAR (v,t,t0,x)	= Normale auf v an der Stelle t = t0
PARA_OSCULATING_CIRCLE (v,t,t0,phi)	= Schmiegekreis an v, in Abhängigkeit von phi
IMP_DIF (u,x,y,n)	= implizite Ableitung DIF(y,x,n) für u(x,y)=0
IMP_CURVATURE (u,x,y)	= Krümmung der impliziten Kurve u=0
IMP_CENTER_OF_CURVATURE (u,x,y)	= Krümmungsmittelpunkt für u=0
IMP_TANGENT (u,x,y,x0,y0)	= Tangente für u=0 an der Stelle x=x0 und y=y0
IMP_PERPENDICULAR (u,x,y,x0,y0)	= Normale an u=0 an der Stelle x0 und y0
IMP_OSCULATING_CIRCLE (u,x,y,x0,y0,phi)	= Schmiegekreis in Abhängigkeit von phi
TANGENT_PLANE (u,v,v0)	= Tangentialebene an u(x,y,x)=0 an der Stelle [x,y,x]=v=v0
NORMAL_LINE (u,v,v0,t)	= Normale an u=0 an der Stelle v=v0, mit Parameter t

<b>NUMBER.MTH</b>	Funktionen der Zahlentheorie
CEILING (m,n)	= kleinste ganze Zahl $\geq m/n$ (Grundeinstellung n = 1)
ROUND (m,n)	= nächste ganze Zahl zu m/n (Grundeinstellung n = 1)
SQUARE_WAVE (n)	= 1, falls MOD(n,2)<1, sonst -1
POWER_MOD (n,d,m)	= $n^d \text{ mod } m$
NTH_PRIME (n)	= n-te PrimZahl
FIBONACCI (n)	= n-te FibonacciZahl

<b>MISC.MTH</b>	Verschiedene Zusatzfunktionen
INT_PARTS (u,v,x)	= Stammfunktion von u(x)*v(x) unter Verwendung der partiellen Integration
INT_SUBST (y,x,u)	= Stammfunktion von y(x) durch Substituieren von x durch u(x)
DEF_INT_SUBST (y,x,u,a,b)	= Integral von y(x) von x=a bis b mit Substitutionsmethode
INVERSE (u,x)	= Inverse von u(x) bezüglich x
POLY_COEFF (u,x,n)	= Koeffizient des Glieds $x^n$ im Polynom u(x)
POLY_DEGREE (u,x)	= Grad des Polynoms u(x)
random_sign	= zufällige Wahl von 1 oder -1
RANDOM_POLY (x,d,s)	= Polynom vom Grad d in x mit Zufallszahlen als Koeffizienten von -s bis s
RANDOM_VECTOR (n,s)	= n elementiger Vektor bestehend aus Zufallszahlen von -s bis s
RANDOM_MATRIX (m,n,s)	= m mal n Matrix mit Zufallszahlen als Elementen von -s bis s