

TI-Nspire™ CX CAS Opslagsvejledning

Vigtige oplysninger

Med mindre andet er anført i den Licens, som følger med programmet, stiller Texas Instruments ingen garantier, udtrykte eller underforståede, herunder, men ikke begrænset til enhver underforstået garanti om salgbarhed og egnethed til et bestemt formål for nogen programmer eller bøger, og stiller udelukkende disse materialer til rådighed "som de forefindes." Texas Instruments kan under ingen omstændigheder holdes ansvarlige for nogen særlige, indirekte, påløbne eller følgeskader i forbindelse med eller som følge af købet eller anvendelsen af disse materialer, og det eneste erstatningsansvar, Texas Instruments kan pådrage sig, uanset handlingen, kan ikke overstige købsprisen for dette produkt, som den er angivet i licensen. Texas Instruments kan endvidere ikke holdes ansvarlig for nogen form for krav i forbindelse med nogen tredjeparts anvendelse af disse materialer.

© 2026 Texas Instruments Incorporated

De faktiske produkter kan variere let fra de viste billeder.

Indhold

Udtryksskabeloner	1
Alfabetisk oversigt	8
A	8
B	17
C	21
D	47
E	62
F	72
G	82
I	92
L	101
M	117
N	127
O	135
P	138
Q	147
R	151
S	166
T	193
U	209
V	209
W	211
X	213
Z	214
Symboler	223
TI-Nspire™ CX II - Tegn kommandoer	251
Grafikprogrammering	251
Grafikskærm	251
Standardvisning og indstillinger	252
Fejlmeddelelser på grafikskærmen	253
Ugyldige kommandoer i grafiktilstand	253
C	254
D	255
F	258
G	260
P	261
S	263
U	265

Tomme (ugyldige) elementer	266
Genveje til indtastning af matematiske udtryk	268
Hierarkiet i EOS™ ligningsoperativsystemet (Equation Operating System)	270
TI-Nspire CX II - TI-Grundlæggende programmeringsfunktioner	272
Automatisk indrykning i programmeringseditoren	272
Forbedrede fejlmeddelelser til TI-Basic	272
Konstanter og værdier	275
Fejlkoder og fejlmeddelelser	276
Fejlkoder- og meddelelser	284
Generelle oplysninger	286
Indeks	287

Udtryksskabeloner

Udtryksskabeloner er en nem metode til at indsætte matematiske udtryk i matematisk standardnotation. Når du indsætter en skabelon, optræder den i indtastningslinjen med små blokke på positioner, hvor du kan indsætte elementer. En markør viser, hvilket element, du kan indsætte.

Anvend piletasten eller tryk på **tab** for at flytte markøren til hvert elements position, og skriv en værdi eller et udtryk for hvert element. Tryk på **enter** eller **ctrl enter** for at beregne udtrykket.

Brøkskabelon

ctrl ÷ -taster



Bemærk: Se også / (divider), side 225.

Eksempel:

$$\frac{12}{8 \cdot 2} = \frac{3}{4}$$

Eksponentskabelon

^ -tast



Bemærk: Skriv første værdi, tryk på **^**, og skriv derefter eksponenten. Tryk på højrepilen (**►**) for at hente markøren tilbage til basislinjen.

Bemærk: Se også ^ (potens), side 226.

Eksempel:

$$2^3 = 8$$

Kvadratrodsskabelon

ctrl x² -taster



Bemærk: Se også $\sqrt{()}$ (kvadratrod), side 237.

Eksempel:

$$\sqrt{4} = 2$$
$$\sqrt{\{9, a, 4\}} = \{3, \sqrt{a}, 2\}$$

Nte rod-skabelon

ctrl ^ -taster



Bemærk: Se også root(), side 162.

Eksempel:



Nte rod-skabelon

ctrl ^ -taster

$$\sqrt[3]{8} \quad 2$$

$$\sqrt[3]{\{8, 27, b\}} \quad \left\{ 2, 3, b^{\frac{1}{3}} \right\}$$

e eksponentskabelon

e^x -taster

e

Den naturlige eksponentialfunktion e opløftet til en potens

Bemærk: Se også $e^{\wedge}()$, side 62.

Eksempel:

$$e^1 \quad e$$

$$e^1. \quad 2.71828182846$$

Log-skabelon

ctrl 10^x -tasten

log

Beregner logaritmen med et angivet grundtal. Ved 10-talslogaritmen, der er standard, udelades grundtallet.

Bemærk: Se også $\log()$, side 113.

Eksempel:

$$\log_4(2.) \quad 0.5$$

Stykkevis-skabelon (2 stykker)

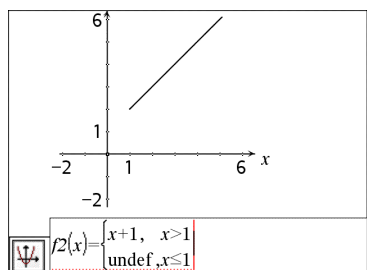
Katalog > $\left\{ \begin{matrix} \end{matrix} \right\}$

$\left\{ \begin{matrix} \end{matrix} \right\}$

Gør det muligt at oprette udtryk og betingelser for en stykkevis funktion med to stykker.- Du kan tilføje et stykke ved at klikke på skabelonen og gentage skabelonen.

Bemærk: Se også $\text{piecewise}()$, side 140.

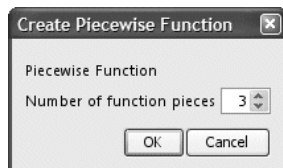
Eksempel:



Stykkevis-skabelon (N stykker)

Katalog > 

Gør det muligt at oprette udtryk og betingelser for en stykkevis funktion med N -stykker. Beder om N .



Bemærk: Se også `piecewise()`, side 140.

Eksempel:

Se eksemplet med stykkevis-skabelonen (2 stykker).

Skabelon til system med 2 ligninger

Katalog > 



Opretter et ligningssystem med to ligninger. Du kan tilføje en række i et eksisterende system ved at klikke i skabelonen og gentage skabelonen.

Bemærk: Se også `system()`, side 192.

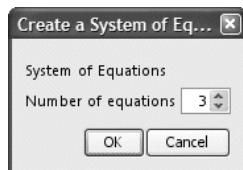
Eksempel:

$$\begin{array}{l} \text{solve} \left(\begin{cases} x+y=0 \\ x-y=5 \end{cases}, x, y \right) \quad x=\frac{5}{2} \text{ and } y=-\frac{5}{2} \\ \text{solve} \left(\begin{cases} y=x^2-2 \\ x+2 \cdot y=-1 \end{cases}, x, y \right) \\ \quad x=-\frac{3}{2} \text{ and } y=\frac{1}{4} \text{ or } x=1 \text{ and } y=-1 \end{array}$$

Skabelon til system med N ligninger

Katalog > 

Gør det muligt at oprette et system af N -ligninger. Beder om N .



Bemærk: Se også `system()`, side 192.

Eksempel:

Se eksemplet med ligningssystemskabelonen (2-ligninger).

Absolut værdi-skabelon

Katalog > 



Bemærk: Se også `abs()`, side 8.

Eksempel:

Absolut værdi-skabelon

Katalog >



$$\left\{ 2, -3, 4, -4^3 \right\} \quad \left\{ 2, 3, 4, 64 \right\}$$

dd°mm'ss.ss''-skabelon

Katalog >



0°0'0"

Eksempel:

Her kan du indtaste vinkler i **gg°mm'ss.ss''** format, hvor **gg** er antallet af decimalgrader, **mm** er antallet af minutter, og **ss.ss** antallet af sekunder.

$$30^{\circ}15'10'' \quad \frac{10891 \cdot \pi}{64800}$$

Matrix-skabelon (2 x 2)

Katalog >



$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

Eksempel:

Opretter en matrix 2 x 2.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot a \quad \begin{bmatrix} a & 2 \cdot a \\ 3 \cdot a & 4 \cdot a \end{bmatrix}$$

Matrix-skabelon (1 x 2)

Katalog >



$\begin{bmatrix} 0 & 0 \end{bmatrix}$

Eksempel:

$$\text{crossP}(\begin{bmatrix} 1 & 2 \end{bmatrix}, \begin{bmatrix} 3 & 4 \end{bmatrix}) \quad \begin{bmatrix} 0 & 0 & -2 \end{bmatrix}$$

Matrix-skabelon (2 x 1)

Katalog >



$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Eksempel:

$$\begin{bmatrix} 5 \\ 8 \end{bmatrix} \cdot 0.01 \quad \begin{bmatrix} 0.05 \\ 0.08 \end{bmatrix}$$

Matrix-skabelon (m x n)

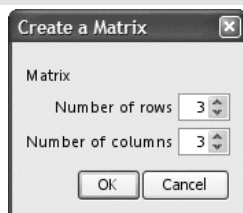
Katalog >



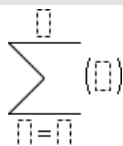
Skabelonen vises, efter at du er blevet bedt om at angive antallet af rækker og kolonner.

Eksempel:

$$\text{diag} \left(\begin{bmatrix} 4 & 2 & 6 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix} \right) \quad \begin{bmatrix} 4 & 2 & 9 \end{bmatrix}$$



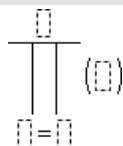
Bemærk: Hvis du opretter en matrix med mange rækker og kolonner, kan det tage et øjeblik, før den kommer frem.

Sum-skabelon (Σ)

Eksempel:

$$\sum_{n=3}^7 (n) \quad 25$$

Bemærk: Se også $\Sigma()$ (`sumSeq`), side 238.

Produkt-skabelon (Π)

Eksempel:

$$\prod_{n=1}^5 \left(\frac{1}{n} \right) \quad \frac{1}{120}$$

Bemærk: Se også $\Pi()$ (`prodSeq`), side 237.

Skabelon til differentialkvotient af første orden



Eksempel:

$$\frac{d}{dx} (x^3) \quad 3 \cdot x^2$$

$$\frac{d}{dx} (x^3) \Big|_{x=3} \quad 27$$

Skabelonerne til differentialkvotienter af første orden kan også anvendes til at beregne differentialkvotienten af første orden i et punkt.

Skabelon til differentialkvotient af første orden

Katalog > 

Bemærk: Se også **d()**
(differentialkvotient), side 234.

Skabelon til differentialkvotient af anden orden

Katalog > 

$$\frac{d^2}{dx^2}(\square)$$

Eksempel:

$$\frac{d^2}{dx^2}(x^3) \quad 6 \cdot x$$

Skabelonerne til differentialkvotienter af anden orden kan også anvendes til at beregne differentialkvotienten af anden orden i et punkt.

$$\frac{d^2}{dx^2}(x^3)|_{x=3} \quad 18$$

Bemærk: Se også **d()**
(differentialkvotient), side 234.

Skabelon til differentialkvotient af nte orden

Katalog > 

$$\frac{d^n}{dx^n}(\square)$$

Eksempel:

$$\frac{d^3}{dx^3}(x^3)|_{x=3} \quad 6$$

Skabelonen til differentialkvotienten af *n*te orden kan anvendes til at beregne differentialkvotienten af *n*te orden.

Bemærk: Se også **d()**
(differentialkvotient), side 234.

Bestemt integral skabelon

Katalog > 

$$\int_a^b \square dx$$

Eksempel:

$$\int_a^b x^2 dx \quad \frac{b^3}{3} - \frac{a^3}{3}$$

Bemærk: Se også **∫()** **integral()**, side 235.

Ubestemt integralskabelon

Katalog > 

$$\int \square dx$$

Eksempel:

Ubestemt integralskabelon

Katalog > 

Bemærk: Se også `∫()` `integral()`, side 235.

$$\int x^2 dx \quad \frac{x^3}{3}$$

Grænseværdi skabelon

Katalog > 

$$\lim_{x \rightarrow a} ()$$

Eksempel:

$$\lim_{x \rightarrow 5} (2 \cdot x + 3) \quad 13$$

Anvend `-` eller `(-)` til grænseværdi fra venstre. Anvend `+` til grænseværdi fra højre.

Bemærk: Se også `limit()`, side 103.

Alfabetisk oversigt

Elementer, hvis navne ikke er alfabetiske (som f.eks. +, ! og >), er anført sidst i dette afsnit, startende (her). Medmindre andet er angivet, udføres alle eksempler i dette afsnit i standard nulstillingstilstand, og alle variable antages at være ikke-defineret.

A

abs()

Katalog >

abs(*UdtrI*)⇒*udtryk*

$\left\{ \frac{\pi}{2}, \frac{\pi}{3} \right\}$

$2-3 \cdot i$

$|z|$

$|x+y \cdot i|$

$\left\{ \frac{\pi}{2}, \frac{\pi}{3} \right\}$

$\sqrt{13}$

$|z|$

$\sqrt{x^2+y^2}$

abs(*ListeI*)⇒*liste*

abs(*MatrixI*)⇒*matrix*

Returnerer den absolutte værdi af argumentet.

Bemærk: Se også **Absolut værdi-skabelon**, side 3.

Hvis argumentet er et komplekst tal, returneres tallets modulus.

Bemærk: Alle udefinerede variable behandles som reelle variable.

amortTbl()

Katalog >

amortTbl(*NPmt*,*N*,*I*,*PV*, [*Pmt*], [*FV*], [*PpY*], [*CpY*], [*PmtAt*], [*afrundVærdi*])⇒*matrix*

amortTbl(12,60,10,5000,,,12,12)

0	0.	0.	5000.
1	-41.67	-64.57	4935.43
2	-41.13	-65.11	4870.32
3	-40.59	-65.65	4804.67
4	-40.04	-66.2	4738.47
5	-39.49	-66.75	4671.72
6	-38.93	-67.31	4604.41
7	-38.37	-67.87	4536.54
8	-37.8	-68.44	4468.1
9	-37.23	-69.01	4399.09
10	-36.66	-69.58	4329.51
11	-36.08	-70.16	4259.35
12	-35.49	-70.75	4188.6

Amortiseringsfunktion, der returnerer en matrix som en amortiseringstabel for et sæt af TVM-argumenter.

NPmt er antallet af betalinger, der skal inkluderes i tabellen. Tabellen starter med den første betaling.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen over TVM-argumenter (side 206).

- Hvis du udelader *Pmt*, sættes den som standard til *Pmt*=**tvmPmt** (*N*,*I*,*PV*,*FV*,*PpY*,*CpY*,*PmtAt*).
- Hvis du udelader *FV*, sættes *FV*=0 som standard.

- Standardværdierne for PpY , CpY og $PmtAt$ er de samme som for TVM-funktionerne.

$afrundVærdi$ angiver antallet af decimaler til afrunding.
Standardværdi=2.

Kolonnerne i resultatmatricen er i denne rækkefølge: Betalingsnummer, beløb betalt til renter, beløb betalt til hovedstol og saldo.

Saldoen, der vises i række n , er saldoen efter betaling n .

Du kan bruge outputmatricen som input for de andre amortiseringsfunktioner $\Sigma Int()$ og $\Sigma Prn()$, side 238 og $bal()$, side 17.

and

Boolsk Udtr1 and Boolsk Udtr2
 $\Rightarrow$ Boolsk udtryk

$x \geq 3$ and $x \geq 4$	$x \geq 4$
$\{x \geq 3, x \leq 0\}$ and $\{x \geq 4, x \leq 2\}$	$\{x \geq 4, x \leq 2\}$

Boolsk Liste1 and Boolsk Liste2
 $\Rightarrow$ Boolsk liste

Boolsk Matrix1 and Boolsk Matrix2
 $\Rightarrow$ Boolsk matrix

Returnerer true eller false eller en forenklet form af den oprindelige indtastning.

Heltal1 and Heltal2 $\Rightarrow$ *heltal*

Sammenligner to heltal bit for bit med en **and**-operation. Internt konverteres begge heltal til 64-bit binære tal med fortegn. Når de tilsvarende bits sammenlignes, er resultatet 1, hvis begge bits er 1. Ellers er resultatet 0. Den returnerede værdi repræsenterer bit-resultaterne og vises i overensstemmelse med den valgte talsystemtilstand.

I hexadecimal tilstand:

0h7AC36 and 0h3D5F	0h2C16
--------------------	--------

Vigtigt: Tallet nul, ikke bogstavet O.

I binær tilstand:

0b100101 and 0b100	0b100
--------------------	-------

I decimal tilstand:

Du kan indtaste heltallene i ethvert talsystem. Til binære eller hexadecimal indtastninger skal du som præfiks benytte henholdsvis 0b eller 0h. Uden præfiks behandles heltallene som decimaltal (10 talssystem).

Hvis du indtaster et decimalt heltal, der er for stort til en 64-bit binær form med fortegn, anvendes en symmetrisk modulooperation til at bringe værdien ind i det korrekte område.

37 and 0b100	4
--------------	---

Bemærk: En binær indtastning kan have op til 64 cifre (præfikset 0b ikke medregnet). En hexadecimal indtastning kan have op til 16 cifre.

angle()

angle(Udtr1)⇒udtryk

Returnerer vinklen på argumentet og fortolker argumentet som et komplekst tal.

Bemærk: Alle udefinerede variable behandles som reelle variable.

I vinkeltilstanden Grader:

angle(0+2·i)	90
--------------	----

I vinkeltilstanden Nygrader:

angle(0+3·i)	100
--------------	-----

I vinkeltilstanden Radian:

angle(1+i)	$\frac{\pi}{4}$
------------	-----------------

angle(z)	$\frac{-\pi \cdot (\text{sign}(z) - 1)}{2}$
----------	---

angle(x+i·y)	$\frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1}\left(\frac{x}{y}\right)$
--------------	--

angle({{1+2·i, 3+0·i, 0-4·i}})	$\left\{\frac{\pi}{2} - \tan^{-1}\left(\frac{1}{2}\right), 0, -\frac{\pi}{2}\right\}$
--------------------------------	---

angle(Liste1)⇒liste

angle(Matrix1)⇒matrix

Returnerer en liste eller matrix med vinkler af elementerne i *Liste1* eller *matrix1*, hvor hvert element fortolkes som et komplekst tal, der repræsenterer et todimensionalt rektangulært koordinatpunkt.

ANOVA

ANOVA *Liste1, Liste2[, Liste3..., Liste20]*
[, *Flag*]

Udfører envejsanalyse af varians til sammenligning af middelværdier for to til 20 populationer. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 187.)

Flag=0 for data, *Flag*=1 for statistik

Output-variabel	Beskrivelse
stat.F	Værdien for F-statistik
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.df	Frihedsgrader i grupperne
stat.SS	Kvadratsum i grupperne
stat.MS	Middelkvadrat for grupperne
stat.dfError	Frihedsgrader for fejl
stat.SSError	Kvadratsum for fejlene
stat.MSError	Middelkvadrat for fejlene
stat.sp	Puljet standardafvigelse
stat.xbarlist	Gennemsnit af input for listerne
stat.CLowerList	95% konfidensintervaller for middelværdien for hver inputliste
stat.CUpperList	95% konfidensintervaller for middelværdien for hver inputliste

ANOVA2-way

ANOVA2way *Liste1, Liste2*
[, *Liste3..., Liste10*][, *levRow*]

Beregner en tovejsanalyse af varians til sammenligning af middelværdier for to til ti populationer. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 187.)

LevRow=0 for Blok

LevRow=2,3,...,*Len*-1, for to-faktor, hvor
Len=length(*List1*)=length(*List2*) = ... = length(*List10*) og *Len* / *LevRow* ∈ {2,3,...}

Output: Blokdesign

Output-variabel	Beskrivelse
stat.F	F statistik for kolonnefaktor
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.df	Frihedsgrader i kolonnefaktoren
stat.SS	Kvadratsum for kolonnefaktoren
stat.MS	Middelkvadrat for kolonnefaktoren
stat.FBlok	F statistik for faktor
stat.PValBlock	Mindste sandsynlighed, ved hvilken nul-hypotesen kan forkastes
stat.dfBlock	Frihedsgrader for faktoren
stat.SSBlock	Kvadratsum for faktoren
stat.MSBlock	Middelkvadrat for faktoren
stat.dfError	Frihedsgrader for fejl
stat.SSError	Kvadratsum for fejlene
stat.MSError	Middelkvadrat for fejlene
stat.s	Standardafvigelse for fejlen

KOLONNEFAKTOR Output

Output-variabel	Beskrivelse
stat.Fcol	F statistik for kolonnefaktor
stat.PValCol	Sandsynlighedsværdi for kolonnefaktoren
stat.dfCol	Frihedsgrader i kolonnefaktoren

Output-variabel	Beskrivelse
stat.SSCol	Kvadratsum for kolonnefaktoren
stat.MSCol	Middelkvadrat for kolonnefaktoren

RÆKKEFAKTOR Output

Output-variabel	Beskrivelse
stat.Frow	F statistik for rækkefaktoren
stat.PValRow	Sandsynlighedsværdi for rækkefaktoren
stat.dfRow	Frihedsgrader for rækkefaktoren
stat.SSRow	Kvadratsum for rækkefaktoren
stat.MSRow	Kvadraternes middelværdi for rækkefaktoren

INTERAKTION-output

Output-variabel	Beskrivelse
stat.FInteract	F statistik for interaktionen
stat.PValInteract	Sandsynlighedsværdi for interaktionen
stat.dfInteract	Frihedsgrader for interaktionen
stat.SSInteract	Kvadratsum for interaktionen
stat.MSInteract	Middelkvadrat for interaktionen

FEJL-output

Output-variabel	Beskrivelse
stat.dfError	Frihedsgrader for fejl
stat.SSError	Kvadratsum for fejlene
stat.MSError	Middelkvadrat for fejlene
s	Standardafvigelse for fejlen

Ans	ctrl (-) -tasten	
Ans⇒ <i>værdi</i>	56	56
Returnerer resultatet af de sidst beregnede udtryk.	56+4	60
	60+4	64

approx()Katalog > **approx(*Udtrl*)⇒*udtryk***

Returnerer beregningen af argumentet som et udtryk med decimale værdier, når det er muligt, uanset den aktuelle indstilling af **Auto eller tilnærmet**.

Dette svarer til at indtaste argumentet og trykke på  .

approx(*Listel*)⇒*liste***approx(*Matrixl*)⇒*matrix***

Returnerer en liste eller *matrix*, hvor hvert element er beregnet til en decimalværdi, hvor det er muligt.

$\text{approx}\left(\frac{1}{3}\right)$	0.333333
$\text{approx}\left(\left\{\frac{1}{3}, \frac{1}{9}\right\}\right)$	{ 0.333333, 0.111111 }
$\text{approx}(\{\sin(\pi), \cos(\pi)\})$	{ 0., -1. }
$\text{approx}(\left[\sqrt{2} \quad \sqrt{3}\right])$	[1.41421 1.73205]
$\text{approx}\left(\left[\frac{1}{3} \quad \frac{1}{9}\right]\right)$	[0.333333 0.111111]
$\text{approx}(\{\sin(\pi), \cos(\pi)\})$	{ 0., -1. }
$\text{approx}(\left[\sqrt{2} \quad \sqrt{3}\right])$	[1.41421 1.73205]

►approxFraction()Katalog > ***Expr* ►approxFraction([*Tol*])⇒*udtryk******List* ►approxFraction([*Tol*])⇒*liste******Matrix* ►approxFraction([*Tol*])⇒*matrix***

Returnere inputtet som en brøk med en tolerance på *Tol*. Hvis *Tol* udelades, anvendes en tolerance på 5.E-14.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive @>**approxFraction**(...).

$\frac{1}{2} + \frac{1}{3} + \tan(\pi)$	0.833333
0.8333333333333333 ►approxFraction(5.E-14)	$\frac{5}{6}$
{ π , 1.5 } ►approxFraction(5.E-14)	$\left\{\frac{5419351}{1725033}, \frac{3}{2}\right\}$

approxRational()Katalog > **approxRational(*Udtryk*[, *Tol*])⇒*udtryk*****approxRational(*Liste*[, *Tol*])⇒*liste*****approxRational(*Matrix*[, *Tol*])⇒*matrix***

Returnerer argumentet som en brøk med en tolerance på *Tol*. Hvis *Tol* udelades, anvendes en tolerance på 5.E-14.

$\text{approxRational}(0.333, 5 \cdot 10^{-5})$	$\frac{333}{1000}$
$\text{approxRational}(\{0.2, 0.33, 4.125\}, 5.E-14)$	$\left\{\frac{1}{5}, \frac{33}{100}, \frac{33}{8}\right\}$

arccos() Se $\cos^{-1}()$, side 33.

arccosh() Se $\cosh^{-1}()$, side 35.

arccot() Se $\cot^{-1}()$, side 36.

arccoth() Se $\coth^{-1}()$, side 37.

arccsc() Se $\csc^{-1}()$, side 39.

arccsch() Se $\operatorname{csch}^{-1}()$, side 40.

arcLen() Katalog > 

arcLen(*Udtr1*, *Var*, *Start*, *S*) $\Rightarrow$ *udtryk*
 Returnerer buelængden for *Udtr1* fra *Start* til *Slut* med hensyn til variabelen *Var*.

Buelængden beregnes som et integral, der forudsætter er funktionsdefinition.

arcLen(*Liste1*, *Var*, *Start*, *Slut*) $\Rightarrow$ *liste*
 Returnerer en liste med buelængder for hvert element i *Liste1* fra *Start* til *Slut* med hensyn til *Var*.

$$\operatorname{arcLen}(\cos(x), x, 0, \pi) \quad 3.8202$$

$$\operatorname{arcLen}(f(x), x, a, b) \quad \int_a^b \sqrt{\left(\frac{d}{dx}(f(x))\right)^2 + 1} dx$$

$$\operatorname{arcLen}(\{\sin(x), \cos(x)\}, x, 0, \pi) \quad \{3.8202, 3.8202\}$$

arcsec() Se $\sec^{-1}()$, side 166.

arcsech() Se $\text{sech}^{-1}()$, side 167.

arcsin() Se $\sin^{-1}()$, side 178.

arcsinh() Se $\sinh^{-1}()$, side 179.

arctan() Se $\tan^{-1}()$, side 194.

arctanh() Se $\tanh^{-1}()$, side 195.

augment()

augment(Liste1, Liste2)⇒liste
Returnerer en ny liste, der er *liste2* føjet til enden af *Liste1*.

augment(Matrix1, Matrix2)⇒matrix
Returnerer en ny matrix, der er *Matrix2* føjet til *Matrix1*. Når tegnet “,” anvendes, skal matricerne have lige store rækkedimensioner, og *Matrix2* føjes til *Matrix1* som nye kolonner. Ændrer ikke *Matrix1* eller *Matrix2*.

Katalog > 

$\text{augment}(\{1, -3, 2\}, \{5, 4\})$	$\{1, -3, 2, 5, 4\}$
--	----------------------

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
$\begin{bmatrix} 5 \\ 6 \end{bmatrix} \rightarrow m2$	$\begin{bmatrix} 5 \\ 6 \end{bmatrix}$
$\text{augment}(m1, m2)$	$\begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 6 \end{bmatrix}$

avgRC()Katalog > 

avgRC(*Udtryk1*, *Var* [=*Værdi*] [, *Trin*])⇒*udtryk*

$$\text{avgRC}(f(x), x, h) \quad \frac{f(x+h) - f(x)}{h}$$

avgRC(*Udtryk1*, *Var* [=*Værdi*] [, *Liste1*])⇒*liste*

$$\text{avgRC}(\sin(x), x, h) | x=2 \quad \frac{\sin(h+2) - \sin(2)}{h}$$

avgRC(*Liste1*, *Var* [=*Værdi*] [, *Trin*])⇒*liste*

$$\text{avgRC}(x^2 - x + 2, x) \quad 2 \cdot (x - 0.4995)$$

$$\text{avgRC}(x^2 - x + 2, x, 0.1) \quad 2 \cdot (x - 0.45)$$

avgRC(*Matrix1*, *Var* [=*Værdi*] [, *Trin*])⇒*matrix*

$$\text{avgRC}(x^2 - x + 2, x, 3) \quad 2 \cdot (x + 1)$$

Returnerer den fremadrettede differenskvotient (gennemsnitlig ændringshastighed).

Udtr1 kan være et brugerdefineret funktionsnavn (se **Func**).

Når *Værdi* er angivet, tilsidesætter den alle forudgående variabeltildelinger og alle nuværende “|” substitutioner for variabelen.

Trin er trinværdien. Hvis *Trin* udelades, er standardværdien 0.001.

Bemærk, at den lignende funktion **centralDiff()** anvender den centrale differenskvotient.

B**bal()**Katalog > 

bal(*NPmt*, *N*, *I*, *PV*, [*Pmt*], [*FV*], [*PpY*], [*CpY*], [*PmtAt*], [*afrundVærdi*])⇒*værdi*

$$\text{bal}(5, 6, 5.75, 5000, , 12, 12) \quad 833.11$$

bal(*NPmt*, *amortTabel*)⇒*værdi*

$$tbl := \text{amortTbl}(6, 6, 5.75, 5000, , 12, 12)$$

Amortiseringsfunktion, der beregner saldo efter en angivet betaling.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen over TVM-argumenter (xxxx).

0	0.	0.	5000.
1	-23.35	-825.63	4174.37
2	-19.49	-829.49	3344.88
3	-15.62	-833.36	2511.52
4	-11.73	-837.25	1674.27
5	-7.82	-841.16	833.11
6	-3.89	-845.09	-11.98

NPmt angiver betalingsnummeret, hvorefter du vil have dataene beregnet.

$$\text{bal}(tbl) \quad 1674.27$$

$N, I, PV, Pmt, FV, PpY, CpY$ og $PmtAt$ er beskrevet i tabellen over TVM-argumenter (xxxx).

- Hvis du udelader Pmt , bliver den som standard $Pmt = \text{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$.
- Hvis du udelader FV , bliver den som standard $FV = 0$.
- Standardværdierne for PpY, CpY og $PmtAt$ er de samme som for TVM-funktionerne.

afroundVærdi angiver antallet af decimaler til afrunding.
Standardværdi=2.

bal($NPmt, \text{amortTabel}$) beregner saldoen efter betaling nummer $NPmt$, baseret på amortiseringstabel amortTabel .
 amortTabel -argumentet skal være en matrix i formen beskrevet under **amortTbl()**, xxxx.

Bemærk: Se også $\Sigma \text{Int}()$ og $\Sigma \text{Prn}()$, xxxx.

►Base2

$\text{Heltal1} \rightarrow \text{Base2} \Rightarrow \text{heltal}$

256 ►Base2	0b100000000
0h1F ►Base2	0b11111

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @►Base2.

Konverterer Heltal1 til et binært tal. Binære eller hexadecimale tal har altid henholdsvis 0b eller 0h som præfiks. Tallet nul, ikke bogstavet O, efterfulgt af b eller h.

0b *binærtTal*

0h *hexadecimaltTal*

Et binært tal kan have op til 64 cifre. Et hexadecimale tal kan have op til 16.

Uden præfiks behandles Heltal1 som decimaltal (10- talssystem). Resultatet vises som binært uanset tilstanden for talsystem.

Negative tal vises på "2-komplement" form. For eksempel:

-1 vises som

0hFFFFFFFFFFFFFF i det hexadecimale talsystem 0b111...111 (64 1-taller) i det binære talsystem

-2^{63} vises som

0h8000000000000000 i det hexadecimale talsystem 0b100...000 (63 nuller) i det binære talsystem

Hvis du indtaster et decimalt heltal, der ligger uden området for en 64-bit binær form med fortegn, anvendes en symmetrisk modulo-operation til at bringe værdien ind i det korrekte område. Undersøg følgende eksempler på værdier uden for området.

2^{63} bliver -2^{63} og vises som

0h8000000000000000 på hexadecimal form 0b100...000 (63 nuller) . På binær form bliver

2^{64} til 0 og vises som

0h0 på hexadecimal form

0b0 på binær form.

$-2^{63} - 1$ bliver $2^{63} - 1$ og vises som

0h7FFFFFFFFFFFFFFF hexadecimal form
0b111...111 (64 1's) på binær form

Heltal ►Base10⇒*heltal*

0b10011►Base10	19
0h1F►Base10	31

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>Base10.

Konverterer *Heltal1* til et decimaltal (i titalssystemet). Binære eller hexadecimal indtastninger skal altid have hhv. 0b eller 0h som præfiks.

0b *binærtTal*

0h *hexadecimaltTal*

Tallet nul, ikke bogstavet O, efterfulgt af b eller h.

Et binært tal kan have op til 64 cifre. Et hexadecimalt tal kan have op til 16.

Uden præfiks behandles *Heltal1* som decimaltal. Resultatet vises som decimaltal uanset tilstanden for talsystem.

Heltal1 ►Base16⇒*heltal*

256►Base16

0h100

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>Base16.

0b111100001111►Base16

0hF0F

Konverterer *Heltal1* til et hexadecimalt tal. Binære eller hexadecimal tal har altid henholdsvis 0b eller 0h som præfiks.

0b *binærtTal*

0h *hexadecimaltTal*

Tallet nul, ikke bogstavet O, efterfulgt af b eller h.

Et binært tal kan have op til 64 cifre. Et hexadecimalt tal kan have op til 16.

Uden præfiks behandles *Heltal1* som decimaltal (10-talssystem). Resultatet vises som hexadecimalt uanset tilstanden for talsystem.

Hvis du indtaster et decimalt heltal, der er for stort til en 64-bit binær form med fortegn, anvendes en symmetrisk modulo-operation til at bringe værdien ind i det korrekte område. Yderligere oplysninger findes under ►Base2, side 18.

binomCdf()

binomCdf(*n*,*p*)⇒*liste*

binomCdf

(*n*,*p*,*nedreGrænse*,*øvreGrænse*)⇒*tal* hvis *nedreGrænse* og *øvreGrænse* er tal, *liste* if *nedreGrænse* og *øvreGrænse* er lister

binomCdf(*n*,*p*,*øvreGrænse*)for $P(0 \leq X \leq \textit{øvreGrænse})$ ⇒*tal* hvis *øvreGrænse* er et tal, *liste* hvis *øvreGrænse* er en liste

Beregner den kumulerede sandsynlighed for den diskrete binomialfordeling med *n* antal forsøg og sandsynligheden *p* for succes ved hvert forsøg.

For $P(X \leq \textit{øvreGrænse})$, sæt *nedreGrænse*=0

binomPdf()

binomPdf(*n*,*p*)⇒*liste*

binomPdf(*n*,*p*,*XVærdi*)⇒*tal* hvis *XVærdi* er et tal, *liste* hvis *XVærdi* er en liste

Beregner en sandsynlighed ved *XVærdi* for den diskrete binomialfordeling med *n* antal forsøg og sandsynligheden *p* for succes ved hvert forsøg.

C**ceiling()**

ceiling(*Udtr1*)⇒*heltal*

ceiling(.456)

1.

Returnerer det nærmeste heltal, der er ≥ argumentet.

Argumentet kan være et reelt eller komplekst tal.

Bemærk: Se også **floor()**.

ceiling(Liste) ⇒ *liste*

ceiling(MatrixI) ⇒ *matrix*

Returnerer en liste eller matrix med oprunding anvendt på hvert element.

$\text{ceiling}(\{-3.1, 1.2, 5\})$	$\{-3., 1.3\}$
$\text{ceiling}\left(\begin{bmatrix} 0 & -3.2 \cdot i \\ 1.3 & 4 \end{bmatrix}\right)$	$\begin{bmatrix} 0 & -3. \cdot i \\ 2. & 4 \end{bmatrix}$

centralDiff()

**centralDiff(Udtr1, Var [=Værdi]
[,Trin])** ⇒ *udtryk*

**centralDiff(Udtr1, Var
[,Trin])** | *Var* = *Værdi* ⇒ *udtryk*

**centralDiff(Udtr1, Var [=Værdi]
[,])** ⇒ *liste*

**centralDiff(1, Var [=Værdi]
[,Trin])** ⇒ *liste*

**centralDiff(MatrixI, Var [=Værdi]
[,Trin])** ⇒ *matrix*

Returnerer den numeriske differentialkvotient udregnet med formlen for den centrale differenskquotient.

Når *Værdi* er angivet, tilsidesætter den alle forudgående variabeltildelinger og alle nuværende “|” substitutioner for variabelen.

Trin er trinværdien. Hvis *Trin* udelades, er standardværdien 0,001.

Ved anvendelse af *Liste1* eller *MatrixI* bliver operationen mappet på tværs af værdierne i listen eller på tværs af matrixelementerne.

Bemærk: Se også **od()**.

$\text{centralDiff}(\cos(x), x, h)$	$\frac{-\cos(x-h) - \cos(x+h)}{2 \cdot h}$
$\lim_{h \rightarrow 0} (\text{centralDiff}(\cos(x), x, h))$	$-\sin(x)$
$\text{centralDiff}(x^3, x, 0.01)$	$3 \cdot (x^2 + 0.000033)$
$\text{centralDiff}(\cos(x), x) \Big _{x=\frac{\pi}{2}}$	$-1.$
$\text{centralDiff}(x^2, x, \{0.01, 0.1\})$	$\{2 \cdot x, 2 \cdot x\}$

cFactor(*Udtr1* [, *Var*]) $\Rightarrow$ *udtryk*

cFactor(*Liste1* [, *Var*]) $\Rightarrow$ *liste*

cFactor(*Matrix1* [, *Var*]) $\Rightarrow$ *matrix*

cFactor(*Udtr1*) returnerer *Udtr1* opløst i faktorer med hensyn til alle dens variable over en fællesnævner.

Udtr1 opløses så meget som muligt i faktorer af 1. grad, også selvom dette indfører nye ikke-reelle tal. Denne mulighed er velegnet, hvis du ønsker opløsning i faktorer med hensyn til mere end en variabel.

cFactor(*Udtr1*, *Var*) returnerer *Udtr1* opløst i faktorer med hensyn til variabelen *Var*.

Udtr1 opløses så meget som muligt i faktorer mod faktorer af 1. grad i *Var*, med mulige ikke-reelle konstanter, også selvom det indfører irrationale konstanter eller deludtryk, der er irrationale i andre variable.

Faktorerne og deres led sorteres med *Var* som hovedvariabel. Ens potenser af *var* samles i hver faktor. Medtag *Var*, hvis opløsning i faktorer kun er nødvendig med hensyn til den pågældende variabel, og du er villig til at acceptere irrationale udtryk i alle andre variable for at øge opløsningen i faktorer med hensyn til *Var*. Der kan forekomme en uforudset faktoropløsning med hensyn til andre variable.

$\text{cFactor}\left(a^3 \cdot x^2 + a \cdot x^2 + a^3 + a \cdot x\right)$	$a \cdot (a^2 + 1) \cdot (x - i) \cdot (x + i)$
$\text{cFactor}\left(x^2 + \frac{4}{9}\right)$	$\frac{(3 \cdot x - 2 \cdot i) \cdot (3 \cdot x + 2 \cdot i)}{9}$
$\text{cFactor}\left(x^2 + 3\right)$	$x^2 + 3$
$\text{cFactor}\left(x^2 + a\right)$	$x^2 + a$

$\text{cFactor}\left(a^3 \cdot x^2 + a \cdot x^2 + a^3 + a \cdot x\right)$	$a \cdot (a^2 + 1) \cdot (x - i) \cdot (x + i)$
$\text{cFactor}\left(x^2 + 3, x\right)$	$(x + \sqrt{3} \cdot i) \cdot (x - \sqrt{3} \cdot i)$
$\text{cFactor}\left(x^2 + a, x\right)$	$(x + \sqrt{a} \cdot i) \cdot (x - \sqrt{a} \cdot i)$

cFactor()

Katalog >

For automatisk indstilling af **Auto eller tilnærmet**-tilstanden tillader medtagningen af *Var* også en approksimation med koefficienter med flydende decimal, hvor irrationale koefficienter ikke kan udtrykkes eksplicit og koncist med de indbyggede funktioner. Også når der kun er en variabel, kan en medtagelse af *Var* give en mere komplet opløsning i faktorer.

Bemærk: Se også **factor()**.

$$\begin{aligned} & \text{cFactor}(x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3) \\ & \quad x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3 \\ & \text{cFactor}(x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3, x) \\ & \quad (x - 0.964673) \cdot (x + 0.611649) \cdot (x + 2.12543) \cdot (x + \end{aligned}$$

Du kan se hele resultatet ved at trykke på **▲** og derefter bruge **◀** og **▶** til at bevæge markøren.

char()

Katalog >

char(Heltal) ⇒ *tegn*

Returnerer en tegnstreng med tegnet nummereret *Heltal* fra grafregnerens tegnsæt. Det gyldige område for *Heltal* er 0–65535.

$$\begin{aligned} & \text{char}(38) \quad \text{"\&"} \\ & \text{char}(65) \quad \text{"A"} \end{aligned}$$

charPoly()

Katalog >

charPoly
(*kvadratMatrix*, *Var*) ⇒ *polynomielt udtryk*

charPoly
(*kvadratMatrix*, *Udtr*) ⇒ *polynomielt udtryk*

charPoly(*kvadratMatrix1*, *Matrix2*) ⇒ *polynomium udtryk*

Returnerer det karakteristiske polynomium af *kvadratMatrix*. Det karakteristiske polynomium af en $n \times n$ matrix *A*, betegnet ved $p_A(\lambda)$, er polynomiet defineret ved

$$p_A(\lambda) = \det(\lambda \cdot I - A)$$

hvor *I* betegner $n \times n$ identitetsmatrixen.

kvadratMatrix1 og *kvadratMatrix2* skal have de samme dimensioner.

$$\begin{aligned} & m := \begin{bmatrix} 1 & 3 & 0 \\ 2 & -1 & 0 \\ -2 & 2 & 5 \end{bmatrix} \quad \begin{bmatrix} 1 & 3 & 0 \\ 2 & -1 & 0 \\ -2 & 2 & 5 \end{bmatrix} \\ & \text{charPoly}(m, x) \quad -x^3 + 5 \cdot x^2 + 7 \cdot x - 35 \\ & \text{charPoly}(m, x^2 + 1) \quad -x^6 + 2 \cdot x^4 + 14 \cdot x^2 - 24 \\ & \text{charPoly}(m, m) \quad 0 \end{aligned}$$

χ^2 2way obsMatrix

chi22way obsMatrix

Beregner en χ^2 test til association på tovejstabelen med tællinger i den observerede matrix *obsMatrix*. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 187.)

Oplysninger om effekten af tomme elementer i en matrix findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat. χ^2	Chi-kvadrat stat: $\text{sum}(\text{observeret} - \text{forventet})^2 / \text{forventet}$
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.df	Frihedsgrader for Chi-kvadrat stat
stat.ExpMat	Matrix med forventet elementtællingstabel, der antager nulhypotese
stat.CompMat	Matrix med bidrag til chi-kvadrat elementbidrag

 χ^2 Cdf()

$\chi^2\text{Cdf}(\text{nedreGrænse}, \text{øvreGrænse}, df) \Rightarrow \text{tal}$
hvis *nedreGrænse* og *øvreGrænse* er tal,
liste hvis *nedreGrænse* og *øvreGrænse* er
lister

$\text{chi2Cdf}(\text{nedreGrænse}, \text{øvreGrænse}, df) \Rightarrow \text{tal}$
hvis *nedreGrænse* og *øvreGrænse* er tal,
liste hvis *nedreGrænse* og *øvreGrænse* er
lister

Beregn χ^2 sandsynlighedsfordelingen mellem *nedreGrænse* og *øvreGrænse* for de angivne frihedsgrader *df*.

For $P(X \leq \text{øvreGrænse})$, sæt
nedreGrænse=0.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

χ^2 GOF *obsListe,forvListe,fg*chi2GOF *obsListe,forvListe,fg*

Udfører en test for at bekræfte, at måledataene er fra en population, der er i overensstemmelse med en angivet distribution. *obsList* er en liste med antal, og skal indeholde heltal. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat. χ^2	Chi-kvadrat stat: $\text{sum}((\text{observeret} - \text{forventet})^2 / \text{forventet})$
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.df	Frihedsgrader for Chi-kvadrat stat
stat.ComplList	Bidrag til chi-kvadrat elementbidrag

 χ^2 Pdf()

$\chi^2\text{Pdf}(XVal,df) \Rightarrow tal$ hvis *XVal* er et tal, *liste* hvis *XVal* er en liste

chi2Pdf(*XVal*,*df*) $\Rightarrow tal$ hvis *XVal* er et tal, *liste*, hvis *XVal* er en liste

Beregner tæthedsfunktionen (pdf) for χ^2 fordelingen ved en angivet *XVal*-værdi for den angivne frihedsgrad *df*.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

ClearAZ

Sletter alle enkelttegnvariable i det aktuelle opgaverum.

Hvis en eller flere af variablene er låst, viser denne kommando en fejlmeddelelse og sletter kun de ulåste variable. Se **unLock**, side 209.

$5 \rightarrow b$	5
b	5
ClearAZ	Done
b	b

ClrErr**ClrErr**

Se et eksempel på **ClrErr**, i Eksempel 2 under **Try**-kommandoen, side 202.

Sletter fejlstatus og indstiller systemvariabel *errCode* til nul.

Else betingelsen i **Try...Else...EndTry**-blokken bør anvende **ClrErr** eller **PassErr**. Brug **ClrErr**, hvis fejlen skal behandles eller ignoreres. Brug **PassErr**, hvis det ikke er kendt, hvad der skal gøres ved fejlen, for at sende den til den næste fejlhåndtering. Hvis der ikke er flere ventende **Try...Else...EndTry**-fejlhåndteringer, vises fejlialogboksen som normalt.

Bemærk: Se også **PassErr**, side 139, og **Try**, side 202.

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

colAugment()

colAugment(*Matrix1*, *Matrix2*) $\Rightarrow$ *matrix*

Returnerer en ny matrix, der er *Matrix2* føjet til *Matrix1*. Matricerne skal have lige store kolonnedimensioner, og *Matrix2* føjes til *Matrix1* som nye rækker. Ændrer ikke *Matrix1* eller *Matrix2*.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
$\begin{bmatrix} 5 & 6 \end{bmatrix} \rightarrow m2$	$\begin{bmatrix} 5 & 6 \end{bmatrix}$
$\text{colAugment}(m1, m2)$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$

colDim()Katalog > **colDim(Matrix)**⇒*udtryk*

$$\text{colDim}\left(\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix}\right) \quad 3$$

Returnerer antallet af kolonner i *Matrix*.**Bemærk:** Se også **rowDim()**.**colNorm()**Katalog > **colNorm(Matrix)**⇒*udtryk*

$$\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix} \rightarrow \text{mat} \quad \begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix}$$

$$\text{colNorm}(\text{mat}) \quad 9$$

Returnerer maksimum for summerne af de absolutte værdier for elementerne i kolonnerne i *Matrix*.**Bemærk:** Udefinerede matricelementer er ikke tilladt. Se også **rowNorm()**.**comDenom()**Katalog > **comDenom(Udtr1,Var)**⇒*udtryk***comDenom(Liste1,Var)**⇒*liste***comDenom(Matrix1,Var)**⇒*matrix*

$$\text{comDenom}\left(\frac{y^2+y}{(x+1)^2}+y^2+y\right)$$

$$\frac{x^2 \cdot y^2 + x^2 \cdot y + 2 \cdot x \cdot y^2 + 2 \cdot x \cdot y + 2 \cdot y^2 + 2 \cdot y}{x^2 + 2 \cdot x + 1}$$

comDenom(Udtryk1) returnerer en forkortet brøk af en fuldt udviklet tæller over en fuldt udviklet nævner.

comDenom(Udtr1,Var) returnerer en brøk med fuldt udviklet tæller og nævner med hensyn til *Var*. Leddene og deres faktorer sorteres med *Var* som hovedvariabel. Ens potenser af *Var* samles. Der kan være tilfældige opløsninger i faktorer i de samlede koefficienter. Sammenlignet med at udelade *Var* sparer dette ofte tid, hukommelse og skærmpads, samtidig med, at det gør udtrykket mere læseligt. Det gør også de efterfølgende operationer på resultatet hurtigere og giver mindre risiko for at fylde hukommelsen op.

$$\text{comDenom}\left(\frac{y^2+y}{(x+1)^2}+y^2+y, x\right)$$

$$\frac{x^2 \cdot y \cdot (y+1) + 2 \cdot x \cdot y \cdot (y+1) + 2 \cdot y \cdot (y+1)}{x^2 + 2 \cdot x + 1}$$

$$\text{comDenom}\left(\frac{y^2+y}{(x+1)^2}+y^2+y, y\right)$$

$$\frac{y^2 \cdot (x^2 + 2 \cdot x + 2) + y \cdot (x^2 + 2 \cdot x + 2)}{x^2 + 2 \cdot x + 1}$$

Hvis *Var* ikke forekommer i *Udtr1*, **comDenom**(*Udtr1*, **returnerer** *Var*) en brøk med udviklet tæller og nævner. Sådanne resultater sparer normalt endnu mere tid, hukommelse og skærmpads. Sådanne resultater, der er delvist opløst i faktorer, gør også de efterfølgende operationer på resultatet hurtigere og mindre tilbøjelige til at fylde hukommelsen op.

Selv hvis der ikke er en nævner, er **comden**-funktionen ofte et hurtigt middel til en delvis faktoroopløsning, hvis **factor()** er for langsom eller bruger hele hukommelsen.

Tip: Indsæt denne **comden**()-funktionsdefinition og prøv den rutinemæssigt som et alternativ til **comDenom()** og **factor()**.

Define *comden*(*exprn*)=**comDenom**(*exprn*,*abc*)
Done

$$\text{comden} \left(\frac{y^2+y}{(x+1)^2} + y^2+y \right) = \frac{(x^2+2 \cdot x+2) \cdot y \cdot (y+1)}{(x+1)^2}$$

$$\text{comden} \left(1234 \cdot x^2 \cdot (y^3-y) + 2468 \cdot x \cdot (y^2-1) \right) = 1234 \cdot x \cdot (x \cdot y+2) \cdot (y^2-1)$$

completeSquare ()

completeSquare(*ExprOrEqn*, *Var*) $\Rightarrow$ udtryk eller ligning

completeSquare(*ExprOrEqn*, *Var*^{Power}) $\Rightarrow$ udtryk eller ligning

completeSquare(*ExprOrEqn*, *Var1*, *Var2* [...]) $\Rightarrow$ udtryk eller ligning

completeSquare(*ExprOrEqn*, {*Var1*, *Var2* [...]}) $\Rightarrow$ udtryk eller ligning

Konverterer et kvadratisk, polynomielt udtryk af formen $a \cdot x^2 + b \cdot x + c$ til formen $a \cdot (x-h)^2 + k$

— eller —

Konverterer en kvadratisk ligning af formen $a \cdot x^2 + b \cdot x + c = d$ til formen $a \cdot (x-h)^2 = k$

Det første argument skal være et kvadratisk udtryk eller en ligning på standardform med hensyn til det andet argument.

$$\text{completeSquare}(x^2+2 \cdot x+3, x) = (x+1)^2+2$$

$$\text{completeSquare}(x^2+2 \cdot x=3, x) = (x+1)^2=4$$

$$\text{completeSquare}(x^6+2 \cdot x^3+3, x^3) = (x^3+1)^2+2$$

$$\text{completeSquare}(x^2+4 \cdot x+y^2+6 \cdot y+3=0, x, y) = (x+2)^2+(y+3)^2=10$$

$$\text{completeSquare}(3 \cdot x^2+2 \cdot y+7 \cdot y^2+4 \cdot x=3, \{x, y\}) = 3 \cdot \left(x+\frac{2}{3}\right)^2 + 7 \cdot \left(y+\frac{1}{7}\right)^2 = \frac{94}{21}$$

$$\text{completeSquare}(x^2+2 \cdot x \cdot y, x, y) = (x+y)^2-y^2$$

Det andet argument skal være en enkelt variabel eller en enkelt variabel opløftet til en rationel potens, f.eks. x, y^2 eller $z^{1/3}$.

Den tredje eller fjerde del af syntaksen forsøger at kvadraterkomplettere med hensyn til variable $Var1, Var2$ [...]).

conj()

conj(*Udtr1*) $\Rightarrow$ *udtryk*

$$\text{conj}(1+2\cdot i) \quad 1-2\cdot i$$

conj(*Liste1*) $\Rightarrow$ *liste*

$$\text{conj}\left(\begin{bmatrix} 2 & 1-3\cdot i \\ -i & -7 \end{bmatrix}\right) \quad \begin{bmatrix} 2 & 1+3\cdot i \\ i & -7 \end{bmatrix}$$

conj(*Matrix1*) $\Rightarrow$ *matrix*

$$\text{conj}(z) \quad z$$

Returnerer kompleks konjugerede af argumentet.

$$\text{conj}(x+i\cdot y) \quad x-y\cdot i$$

Bemærk: Alle udefinerede variable behandles som reelle variable.

constructMat()

constructMat

(*Udtr, Var1, Var2, antalRækker, antalKol*) $\Rightarrow$ *matrix*

$$\text{constructMat}\left(\frac{1}{i+j}, i, j, 3, 4\right) \quad \begin{bmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}$$

Returnerer en matrix baseret på argumenter.

Udtr er et udtryk i variablerne *Var1* og *Var2*. Elementer i den resulterende matrix er dannet ved beregning af *Udtr* for hver forøget værdi af *Var1* og *Var2*.

Var1 er automatisk forøget fra 1 til *antalRækker*. Inden for hver række, *Var2* er forøget fra 1 til *antalKol*.

CopyVar *Var1*, *Var2*

Define $a(x) = \frac{1}{x}$	Done
-----------------------------	------

CopyVar *Var1*, *Var2*.

CopyVar *Var1*, *Var2* kopierer værdien af variabelen *Var1* til variabelen *Var2*, og opretter *Var2* hvis nødvendigt. Variablen *Var1* skal have en værdi

Define $b(x) = x^2$	Done
CopyVar <i>a</i> , <i>c</i> : $c(4)$	$\frac{1}{4}$
CopyVar <i>b</i> , <i>c</i> : $c(4)$	16

Hvis *Var1* er navnet på en eksisterende brugerdefineret funktion, kopieres definitionen af denne funktion til funktionen *Var2*. Funktionen *Var1* skal defineres.

Var1 skal opfylde kravene til navngivning af variable, eller være et indirekte udtryk, der kan reduceres til et variabelnavn, der opfylder betingelserne.

CopyVar *Var1*, *Var2*. kopierer alle elementer af *Var1*. variabelgruppen til *Var2*. gruppen, og opretter *Var2*. hvis nødvendigt.

Var1. skal være navnet på en eksisterende variabelgruppe, så som statistikken *stat.nn* resultater eller variable dannet ved brug af **LibShortcut()**-funktionen. Hvis *Var2*. allerede eksisterer, vil denne kommando udskifte alle elementer, der er fælles i begge grupper, og tilføje de elementer, som ikke allerede eksisterer. Hvis et eller flere elementer i *Var2*. er låst, efterlades alle elementer i *Var2*. uændret.

<i>aa.a</i> :=45	45																
<i>aa.b</i> :=6.78	6.78																
CopyVar <i>aa</i> , <i>bb</i> .	Done																
getVarInfo()	<table> <tr> <td><i>aa.a</i></td> <td>"NUM"</td> <td>"[]"</td> <td>0</td> </tr> <tr> <td><i>aa.b</i></td> <td>"NUM"</td> <td>"[]"</td> <td>0</td> </tr> <tr> <td><i>bb.a</i></td> <td>"NUM"</td> <td>"[]"</td> <td>0</td> </tr> <tr> <td><i>bb.b</i></td> <td>"NUM"</td> <td>"[]"</td> <td>0</td> </tr> </table>	<i>aa.a</i>	"NUM"	"[]"	0	<i>aa.b</i>	"NUM"	"[]"	0	<i>bb.a</i>	"NUM"	"[]"	0	<i>bb.b</i>	"NUM"	"[]"	0
<i>aa.a</i>	"NUM"	"[]"	0														
<i>aa.b</i>	"NUM"	"[]"	0														
<i>bb.a</i>	"NUM"	"[]"	0														
<i>bb.b</i>	"NUM"	"[]"	0														

corrMat()Katalog > **corrMat**(*Liste1*,*Liste2*[,...[,*Liste20*]])

Beregner korrelationsmatricen for den udvidede matrix [*Liste1* *Liste2* . . . *Liste20*].

cosKatalog > *Udtr* **cos**

$(\sin(x))^2 \rightarrow \cos$	$1 - (\cos(x))^2$
--------------------------------	-------------------

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>cos.

Repræsenterer *Expr* i termer af cosinus. Det er en konverteringsoperator for visning. Denne operator kan kun anvendes ved slutningen af indtastningslinjen.

cos reducerer alle potenser af

$$\sin(\dots) \text{ modulo } 1 - \cos(\dots)^2$$

således, at alle tilbageværende potenser af cos(...) har eksponenter i området (0, 2). Således vil resultatet være uden sin (...) hvis, og kun hvis sin(...) nu optræder i det givne udtryk med lige potenser.

Bemærk: Konverteringsoperatoren understøttes ikke i vinkeltilstandene Grader eller Nygrader. Før brug skal man sikre at vinkeltilstanden er indstillet til radianer, og at *Udtr* ikke indeholder eksplicit reference til grader eller nygrader.

cos()

 -tast

cos(*Udtr1*) ⇒ *udtryk*

cos(*Liste1*) ⇒ *list*

cos(*Udtr1*) returnerer cosinus af argumentet som et udtryk.

cos(*Liste1*) returnerer en liste med cosinus til alle elementer i *Liste1*.

Bemærk: Argumentet fortolkes som en vinkel målt i grader, nygrader eller radianer afhængigt af den aktuelt indstillede vinkeltilstand. Du kan bruge °, G eller r til midlertidigt at ignorere vinkeltilstanden.

I vinkeltilstanden Grader:

$\cos\left(\frac{\pi}{4}\right)$	$\frac{\sqrt{2}}{2}$
$\cos(45)$	$\frac{\sqrt{2}}{2}$
$\cos(\{0,60,90\})$	$\left\{1, \frac{1}{2}, 0\right\}$

I vinkeltilstanden Nygrader:

$\cos(\{0,50,100\})$	$\left\{1, \frac{\sqrt{2}}{2}, 0\right\}$
----------------------	---

I vinkeltilstanden Radian:

$\cos\left(\frac{\pi}{4}\right)$	$\frac{\sqrt{2}}{2}$
$\cos(45^\circ)$	$\frac{\sqrt{2}}{2}$

cos(kvadratMatrix1)⇒kvadratMatrix

Returnerer matrixcosinus af *kvadratMatrix1*. Dette er ikke det samme som at beregne cosinus for hvert element.

Når en skalær funktion $f(A)$ opererer på *kvadratMatrix1* (A), beregnes resultatet efter algoritmen:

Beregn egenværdierne (λ_i) og egenvektorer (V_i) af A.

KvadratMatrix1 skal være diagonaliserbar. Den må heller ikke have symbolske variable, der ikke er tildelt en værdi.

Dan matricerne:

$$B = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \text{ and } X = [V_1, V_2, \dots, V_n]$$

Derefter $A = X B X^{-1}$ og $f(A) = X f(B) X^{-1}$. For eksempel $\cos(A) = X \cos(B) X^{-1}$, hvor:

$\cos(B) =$

$$\begin{bmatrix} \cos(\lambda_1) & 0 & \dots & 0 \\ 0 & \cos(\lambda_2) & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \cos(\lambda_n) \end{bmatrix}$$

Alle beregninger udføres aritmetisk med flydende komma.

I vinkeltilstanden Radian:

$\cos\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$	$\begin{bmatrix} 0.212493 & 0.205064 & 0.121389 \\ 0.160871 & 0.259042 & 0.037126 \\ 0.248079 & -0.090153 & 0.218972 \end{bmatrix}$
---	---

cos⁻¹(Udtr1)⇒udtryk

I vinkeltilstanden Grader:

cos⁻¹()**cos⁻¹(Liste1)**⇒liste $\cos^{-1}(1)$ 0**cos⁻¹(Udtr1)** returnerer den vinkel, hvis cosinus er *Udtr1* som et udtryk.**cos⁻¹(Liste1)** returnerer en liste med de inverse cosinusværdier for hvert element af *Liste1*.**Bemærk:** Resultatet returneres som en vinkel i grader eller radianer afhængigt af den aktuelle vinkeltilstand.**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arccos (...)**.**cos⁻¹(kvadratMatrix1)**⇒kvadratMatrixReturnerer den matrixinverse cosinus af *kvadratMatrix1*. Dette er ikke det samme som at beregne den inverse cosinus for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.*KvadratMatrix1* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

I vinkeltilstanden Nygrader:

 $\cos^{-1}(0)$ 100

I vinkeltilstanden Radian:

 $\cos^{-1}(\{0,0,2,0,5\})$ $\left\{\frac{\pi}{2}, 1.36944, 1.0472\right\}$

I vinkeltilstanden radian og rektangulært komplekst format:

 $\cos^{-1}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$
 $\begin{bmatrix} 1.73485+0.064606\cdot i & -1.49086+2.10514 \\ -0.725533+1.51594\cdot i & 0.623491+0.77836\cdot i \\ -2.08316+2.63205\cdot i & 1.79018-1.27182\cdot i \end{bmatrix}$

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

cosh()**Katalog** > **cosh(Udtr1)**⇒udtryk

I vinkeltilstanden Grader:

cosh(Liste1)⇒liste $\cosh\left(\left(\frac{\pi}{4}\right)_r\right)$ cosh(45)**cosh(Udtr1)** returnerer den hyperbolske cosinus af argumentet som et udtryk.**cosh(Liste1)** returnerer en liste med hyperbolsk cosinus for hvert element i *Liste1*.**cosh(kvadratMatrix1)**⇒kvadratMatrix

I vinkeltilstanden Radian:

cosh()

Katalog > 

Returnerer matrix hyperbolsk cosinus af *kvadratMatrix1*. Dette er ikke det samme som at beregne den hyperbolske cosinus for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

KvadratMatrix1 skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

$$\cosh \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 421.255 & 253.909 & 216.905 \\ 327.635 & 255.301 & 202.958 \\ 226.297 & 216.623 & 167.628 \end{bmatrix}$$

cosh⁻¹()

Katalog > 

cosh⁻¹(Udtr1) ⇒ *udtryk*

$$\cosh^{-1}(1) \quad 0$$

cosh⁻¹(Liste1) ⇒ *liste*

$$\cosh^{-1}(\{1, 2, 1, 3\}) \quad \{0, 1.37286, \cosh^{-1}(3)\}$$

cosh⁻¹(Udtr1) returnerer den inverse hyperbolske cosinus af argumentet som et udtryk.

cosh⁻¹(Liste1) returnerer en liste med de inverse hyperbolske cosinusværdier for hvert element i *Liste1*.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arccosh (...)**.

cosh⁻¹(kvadratMatrix1) ⇒ *kvadratMatrix*

Returnerer den matrixinverse hyperbolske cosinus af *kvadratMatrix1*. Dette er ikke det samme som at beregne den inverse hyperbolske cosinus for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

KvadratMatrix1 skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

I vinkeltilstanden Radian og i rektangulært komplekst format:

$$\cosh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 2.52503+1.73485 \cdot i & -0.009241-1.49086 \cdot i \\ 0.486969-0.725533 \cdot i & 1.66262+0.623491 \cdot i \\ -0.322354-2.08316 \cdot i & 1.26707+1.79018 \cdot i \end{bmatrix}$$

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

cot()

 -tast

cot(Udtr1) ⇒ *udtryk*

I vinkeltilstanden Grader:

cot(Liste1) ⇒ *liste*

cot()



Returnerer cotangens til *Udtr1* eller returnerer en liste med cotangens til alle elementer i *Liste1*.

Bemærk: Argumentet fortolkes som en vinkel målt i grader, nygrader eller radianer afhængigt af den aktuelt indstillede vinkeltilstand. Du kan bruge °, G eller r til midlertidigt at ignorere vinkeltilstanden.

$$\text{cot}(45) \quad 1$$

I vinkeltilstanden Nygrader:

$$\text{cot}(50) \quad 1$$

I vinkeltilstanden Radian:

$$\text{cot}(\{1, 2, 1, 3\}) \quad \left\{ \frac{1}{\tan(1)}, 0.584848, \frac{1}{\tan(3)} \right\}$$

cot⁻¹()



cot⁻¹(Udtr1)⇒udtryk

cot⁻¹(Liste1)⇒liste

Returnerer den vinkel, hvis cotangens er *Udtr1*, eller returnerer en liste med den inverse cotangens til hvert element i *Liste1*.

Bemærk: Resultatet returneres som en vinkel i grader eller radianer afhængigt af den aktuelle vinkeltilstand.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arccot (...)**.

I vinkeltilstanden Grader:

$$\text{cot}^{-1}(1) \quad 45.$$

I vinkeltilstanden Nygrader:

$$\text{cot}^{-1}(1) \quad 50.$$

I vinkeltilstanden Radian:

$$\text{cot}^{-1}(1) \quad \frac{\pi}{4}$$

coth()



coth(Udtr1)⇒udtryk

coth(Liste1)⇒liste

Returnerer den hyperbolske cotangens til *Udtr1* eller returnerer en liste med den hyperbolske cotangens til alle elementer i *Liste1*.

$$\text{coth}(1.2) \quad 1.19954$$

$$\text{coth}(\{1, 3, 2\}) \quad \left\{ \frac{1}{\tanh(1)}, 1.00333 \right\}$$

coth⁻¹()**Katalog** > **coth⁻¹(Udtr1)**⇒*udtryk***coth⁻¹(3,5)** 0.293893**coth⁻¹(Liste1)**⇒*liste***coth⁻¹({-2,2,1,6})**
$$\left\{ \frac{-\ln(3)}{2}, 0.518046, \frac{\ln\left(\frac{7}{5}\right)}{2} \right\}$$

Returnerer den inverse hyperbolske cotangens til *Udtr1* eller returnerer en liste med den inverse hyperbolske cotangens til alle elementer i *Liste1*.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arccoth (...)**.

count()**Katalog** > **count(Værdi1 eller Liste1
[,Værdi2 eller Liste2 [...]])**⇒*værdi***count(2,4,6)** 3**count({2,4,6})** 3

Returnerer det akkumulerede antal af alle elementer i argumenterne, der evalueres til numeriske værdier.

count(2,{4,6}, $\begin{bmatrix} 8 & 10 \\ 12 & 14 \end{bmatrix}$) 7

Hvert argument kan være et udtryk, en værdi, en liste eller en matrix. Du kan blande datatyper og anvende argumenter med forskellige dimensioner.

count($\frac{1}{2}$,3+4*i*,undef,"hello",x+5,sign(0))
2

For lister, matricer eller celleområder evalueres hvert element for at bestemme, om det skal inkluderes i tællingen.

I det sidste eksempel tælles kun 1/2 og 3+4*i*. De resterende argumenter evalueres ikke til numeriske værdier, hvis det antages at *x* er udefineret.

I applikationen Lister og regneark kan du anvende et celleområde i stedet for ethvert argument.

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 266.

countif()**Katalog** > **countif(Liste,Kriterie)**⇒*værdi***countif({1,3,"abc",undef,3,1},3)** 2

Returnerer det akkumulerede antal af alle elementer i *Liste*, der opfylder de angivne *Kriterie*.

Tæller antallet af elementer lig med 3.

Kriterie kan være:

- En værdi, et udtryk eller en streng. For eksempel tæller **3** kun de elementer i *Liste*, der reduceres til værdien 3.
- Et Boolsk udtryk, der indeholder symbolet **?** som pladsholder for hvert element. For eksempel **?<5** tæller kun de elementer i *Liste*, der er mindre end 5.

I applikationen Lister og regneark kan du anvende et celleområde i stedet for *Liste*.

Tomme (ugyldige) elementer i listen ignoreres. Yderligere oplysninger om tomme elementer findes på side 266.

Bemærk: Se også **sumIf()**, side 191, og **frequency()**, side 80.

$\text{countIf}(\{ "abc", "def", "abc", 3 \}, "def")$	1
---	---

Tæller antallet af elementer lig med "def."

$\text{countIf}(\{ x^{-2}, x^{-1}, 1, x, x^2 \}, x)$	1
--	---

Tæller antallet af elementer lig med x ; dette eksempel antager, at variabelen x er udefineret.

$\text{countIf}(\{ 1, 3, 5, 7, 9 \}, ? < 5)$	2
--	---

Tæller 1 og 3.

$\text{countIf}(\{ 1, 3, 5, 7, 9 \}, 2 < ? < 8)$	3
--	---

Tæller 3, 5 og 7.

$\text{countIf}(\{ 1, 3, 5, 7, 9 \}, ? < 4 \text{ or } ? > 6)$	4
--	---

Tæller 1, 3, 7 og 9.

cPolyRoots()

cPolyRoots(*Poly*, *Var*) $\Rightarrow$ *liste*

cPolyRoots(*ListeAfKoeff*) $\Rightarrow$ *liste*

Den første syntaks, **cPolyRoots**(*Poly*, *Var*), returnerer en liste med komplekse rødder af polynomiet *Poly* med hensyn til variabelen *Var*.

Poly skal være et polynomium i en variabel.

Den anden syntaks, **cPolyRoots**(*OfCoeffs*), returnerer en liste med komplekse rødder for koefficienterne i *ListeAfKoeff*.

Bemærk: Se også **polyRoots()**, side 144.

$\text{polyRoots}(y^3 + 1, y)$	$\{-1\}$
--------------------------------	----------

$\text{cPolyRoots}(y^3 + 1, y)$	$\left\{ -1, \frac{1}{2} - \frac{\sqrt{3}}{2}i, \frac{1}{2} + \frac{\sqrt{3}}{2}i \right\}$
---------------------------------	---

$\text{polyRoots}(x^2 + 2 \cdot x + 1, x)$	$\{-1, -1\}$
--	--------------

$\text{cPolyRoots}(\{1, 2, 1\})$	$\{-1, -1\}$
----------------------------------	--------------

crossP()Katalog > **crossP(Liste1, Liste2) ⇒ liste**

Returnerer vektorproduktet af *Liste1* og *liste2* som en liste.

Liste1 og *Liste2* skal have ens dimension, og dimensionen skal være 2 eller 3.

crossP(Vektor1, Vektor2) ⇒ vektor

Returnerer en række eller kolonnevektor (afhængigt af argumenterne), der er vektorproduktet af *Vektor1* og *Vektor2*.

Både *Vektor1* og *Vektor2* skal være rækkevektorer, eller begge skal være kolonnevektorer. Begge vektorer skal have ens dimension, og dimensionen skal være enten 2 eller 3.

$$\begin{aligned} \text{crossP}(\{a1, b1\}, \{a2, b2\}) &= \{0, 0, a1 \cdot b2 - a2 \cdot b1\} \\ \text{crossP}(\{0.1, 2.2, -5\}, \{1, -0.5, 0\}) &= \{-2.5, -5., -2.25\} \end{aligned}$$

$$\begin{aligned} \text{crossP}([1 \ 2 \ 3], [4 \ 5 \ 6]) &= [-3 \ 6 \ -3] \\ \text{crossP}([1 \ 2], [3 \ 4]) &= [0 \ 0 \ -2] \end{aligned}$$

csc() **-tast****csc(Udtr1) ⇒ udtryk**

I vinkeltilstanden Grader:

csc(Liste1) ⇒ liste

$$\text{csc}(45) = \sqrt{2}$$

Returnerer cosecansen til *Udtr1* eller returnerer en liste med cosecansen til alle elementer i *Liste1*.

I vinkeltilstanden Nygrader:

$$\text{csc}(50) = \sqrt{2}$$

I vinkeltilstanden Radian:

$$\text{csc}\left(\left\{1, \frac{\pi}{2}, \frac{\pi}{3}\right\}\right) = \left\{\frac{1}{\sin(1)}, 1, \frac{2 \cdot \sqrt{3}}{3}\right\}$$

csc⁻¹() **-tast****csc⁻¹(Udtr1) ⇒ udtryk**

I vinkeltilstanden Grader:

csc⁻¹(Liste1) ⇒ liste

$$\text{csc}^{-1}(1) = 90.$$

I vinkeltilstanden Nygrader:

$\text{csc}^{-1}()$

 **-tast**

Returnerer den vinkel, hvis cosecans er *Udtr1*, eller returnerer en liste med den inverse cosecans til de enkelte elementer på *Liste1*.

Bemærk: Resultatet returneres som en vinkel i nygrader eller radianer afhængigt af den aktuelle vinkeltilstand.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arccsc (...)**.

$$\text{csc}^{-1}(1)$$

$$100.$$

I vinkeltilstanden Radian:

$$\text{csc}^{-1}\left(\left\{1, 4, 6\right\}\right) \quad \left\{\frac{\pi}{2}, \sin^{-1}\left(\frac{1}{4}\right), \sin^{-1}\left(\frac{1}{6}\right)\right\}$$

$\text{csch}()$

Katalog > 

csch(*Udtr1*) $\Rightarrow$ *udtryk*

$$\text{csch}(3)$$

$$\frac{1}{\sinh(3)}$$

csch(*Liste1*) $\Rightarrow$ *liste*

Returnerer den hyperbolske cosecans til *Udtr1* eller returnerer en liste med den hyperbolske cosecans til alle elementer i *Liste1*.

$$\text{csch}\left(\left\{1, 2, 1, 4\right\}\right)$$

$$\left\{\frac{1}{\sinh(1)}, 0.248641, \frac{1}{\sinh(4)}\right\}$$

$\text{csch}^{-1}()$

Katalog > 

csch⁻¹(*Udtr1*) $\Rightarrow$ *udtryk*

$$\text{csch}^{-1}(1)$$

$$\sinh^{-1}(1)$$

csch⁻¹(*Liste1*) $\Rightarrow$ *liste*

Returnerer den hyperbolske cosecans til *Udtr1* eller returnerer en liste med den inverse hyperbolske cosecans til hvert element i *Liste1*.

$$\text{csch}^{-1}\left(\left\{1, 2, 1, 3\right\}\right)$$

$$\left\{\sinh^{-1}(1), 0.459815, \sinh^{-1}\left(\frac{1}{3}\right)\right\}$$

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arccsch (...)**.

cSolve()

Katalog > 

cSolve(*Ligning*, *Var*) $\Rightarrow$ *Boolsk udtryk*

$$\text{cSolve}(x^3=1, x)$$

cSolve(*Ligning*, *Var*=*Gæt*) $\Rightarrow$ *Boolesk udtryk*

$$x=\frac{1}{2}+\frac{\sqrt{3}}{2}\cdot i \text{ or } x=\frac{1}{2}-\frac{\sqrt{3}}{2}\cdot i \text{ or } x=-1$$

cSolve(*Ulighed*, *Var*) $\Rightarrow$ *Boolsk udtryk*

$$\text{solve}(x^3=1, x)$$

$$x=-1$$

Returnerer komplekse løsninger til en ligning eller ulighed for *Var*. Målet er at finde alle reelle og ikke-reelle løsninger. Også selvom *Ligning* er reel, tillader **cSolve()** ikke-reelle resultater i den reelle tilstand.

cSolve() indstiller midlertidigt til det komplekse domæne under løsningen, også selvom det aktuelle domæne er det reelle. I det komplekse domæne anvender brøkpotenser med ulige nævnere hovedområdet i stedet for det reelle område. Som følge heraf er løsninger fra **solve()** af ligninger med sådanne brøkpotenser ikke nødvendigvis en delmængde af løsningerne fra **cSolve()**.

cSolve() starter med eksakte symbolske metoder. Med undtagelse af **Eksakt** tilstand anvender **cSolve()** om nødvendigt også en iterativ tilnærmet kompleks polynomiel faktoropløsning.

Bemærk: Se også **cZeros()**, **solve()**, og **zeros()**.

cSolve(*Eqn1* and *Eqn2* [and...],

VarEllerGæt1, *VarEllerGæt2* [, ...])
⇒ Boolesk udtryk

cSolve(*SystemAfLign*, *VarEllerGæt1*,
VarEllerGæt2 [, ...]) ⇒ Boolesk udtryk

Returnerer mulige komplekse løsninger til de sammenhørende algebraiske ligninger, hvor hvert *varEllerGæt* angiver en variabel, du vil løse for.

Du kan også vælge at angive et initielt gæt til en variabel. Hvert *varEllerGæt* skal have formen:

$\text{cSolve}\left(x^{\frac{1}{3}}=-1, x\right)$	false
$\text{solve}\left(x^{\frac{1}{3}}=-1, x\right)$	$x=-1$

I Vis cifre-tilstand for Fast 2:

$\text{exact}\left(\text{cSolve}\left(x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3=0, x\right)\right)$
$x\cdot\left(x^4+4\cdot x^3+5\cdot x^2-6\right)=3$
$\text{cSolve}\left(\text{Ans}, x\right)$
$x=-1.11+1.07\cdot i$ or $x=-1.11-1.07\cdot i$ or $x=-2.9$

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

variabel

– eller –

variabel = reelt eller ikke-reelt tal

For eksempel er x gyldig, og det er $x=3+i$ også.

Hvis alle ligningerne er polynomier, og hvis du IKKE angiver nogle initiale gæt, benytter **cSolve()** den leksikale Gröbner/Buchberger eliminationsmetode som forsøg på at bestemme **alle** komplekse løsninger.

Komplekse løsninger kan omfatte både reelle og ikke-reelle løsninger som i eksemplet til højre.

$$\text{cSolve}\left(u \cdot v - u = v \text{ and } v^2 = -u, \{u, v\}\right) \\ u = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } v = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ or } u = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i$$

Du kan se hele resultatet ved at trykke på  og derefter bruge  og  til at bevæge markøren.

Sammenhørende polynomielle ligninger kan have ekstra variable, der ikke har nogen værdier men repræsenterer givne numeriske værdier, der kan substitueres efterfølgende.

$$\text{cSolve}\left(u \cdot v - u = c \cdot v \text{ and } v^2 = -u, \{u, v\}\right) \\ u = \frac{-(\sqrt{4 \cdot c - 1} \cdot i + 1)^2}{4} \text{ and } v = \frac{\sqrt{4 \cdot c - 1} \cdot i + 1}{2} \text{ or}$$

Du kan se hele resultatet ved at trykke på  og derefter bruge  og  til at bevæge markøren.

Du kan også medtage løsningsvariable, der ikke optræder i ligningerne. Disse løsninger viser, hvordan løsningsfamilier kan indeholde arbitrære konstanter af formen $c \cdot k$, hvor k er et heltalssuffix fra 1 til og med 255.

$$\text{cSolve}\left(u \cdot v - u = v \text{ and } v^2 = -u, \{u, v, w\}\right) \\ u = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } v = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ and } w = c \cdot d3 \text{ or}$$

Du kan se hele resultatet ved at trykke på  og derefter bruge  og  til at bevæge markøren.

For polynomielle systemer afhænger beregningstiden eller hukommelsesforbrug stærkt af den rækkefølge løsningsvariablene angives i. Hvis det initielle valg kræver for meget hukommelse eller tålmodighed, skal du prøve at omarrangere variablene i ligningerne og/eller *varEllerGæt* listen.

Hvis du ikke medtager nogen gæt, og hvis en ligning er ikke-polynomiell i en variabel, men alle ligninger er lineære i alle løsningsvariable, anvender **cSolve()** en Gauss-eliminering i et forsøg på at bestemme alle løsninger.

$$\text{cSolve}\left(u+v=e^w \text{ and } u-v=i, \{u,v\}\right)$$

$$u=\frac{e^w+i}{2} \text{ and } v=\frac{e^w-i}{2}$$

Hvis et system hverken er polynomielt i alle variable eller lineært i sine løsningsvariable, bestemmer **cSolve()** højst en løsning med en iterativ approksimationsmetode. Dette gøres ved at lade antallet af løsningsvariable være lig med antallet af ligninger og reducere alle andre variable i ligningerne til tal.

$$\text{cSolve}\left(e^z=w \text{ and } w=z^2, \{w,z\}\right)$$

$$w=0.494866 \text{ and } z=0.703467$$

Et ikke-reelt gæt er ofte nødvendigt for at bestemme en ikke-reel løsning. For at opnå konvergens skal et gæt være meget tæt på en løsning.

$$\text{cSolve}\left(e^z=w \text{ and } w=z^2, \{w,z=1+i\}\right)$$

$$w=0.149606+4.8919 \cdot i \text{ and } z=1.58805+1.5402 \cdot i$$

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

CubicReg

CubicReg $X, Y, [Frekv] [, Kategori, Medtag]$

Beregner polynomiell tredjegradsregression $y = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$ på listerne X og Y med frekvens $Frekv$. En sammenfatning af resultaterne lagres i *stat.resultat* variable. (side 187.)

Alle lister skal have samme dimension med undtagelse af *Medtag*.

X og Y er lister med uafhængige og afhængige variable.

Frekv er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hvert tilsvarende X og Y datapunkt. Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

Kategori er en liste, der indeholder kategorikoder for X og Y data.

Medtag er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot x^3 + b \cdot x^2 + c \cdot x + d$
stat.a, stat.b, stat.c, stat.d	Regressionskoefficienter
stat.r ²	Forklaringsgraden
stat.Resid	Residualer fra regressionen
stat.XReg	Liste af datapunkter i den modificerede <i>X-liste</i> , der faktisk bruges i regressionen ud fra begrænsninger af <i>Hyppighed</i> , <i>Kategoriliste</i> , og <i>Medtag kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y-liste</i> , der faktisk bruges i regressionen ud fra begrænsninger af <i>Hyppighed</i> , <i>Kategoriliste</i> , og <i>Medtag kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

cumulativeSum()

cumulativeSum(Liste1) ⇒ liste

$\text{cumulativeSum}(\{1,2,3,4\}) \quad \{1,3,6,10\}$

Returnerer en liste med de kumulerede summer af elementerne i *Liste1*, startende ved element 1.

cumulativeSum(Matrix1) ⇒ matrix

Returnerer en matrix af de kumulerede summer af elementerne i *Matrix1*. Hvert element er den kumulerede sum af kolonnen fra top til bund.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$
$\text{cumulativeSum}(m1)$	$\begin{bmatrix} 1 & 2 \\ 4 & 6 \\ 9 & 12 \end{bmatrix}$

Et tomt (ugyldigt) element i *Liste1* eller *Matrix1* giver et ugyldigt element i den resulterende liste eller matrix. Yderligere oplysninger om tomme elementer findes på side 266.

Cycle

Katalog > 

Cyklus

Overfører kontrol direkte til næste iteration i den aktuelle løkke (**For**, **While** eller **Loop**).

Cycle må ikke ikke benyttes uden for (**For**, **While** eller **Loop**).

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

Funktionsliste, der adderer heltallene fra 1 til 100 og udelader 50.

Define $g()$ =Func	Done
Local $temp,i$	
$0 \rightarrow temp$	
For $i,1,100,1$	
If $i=50$	
Cycle	
$temp+i \rightarrow temp$	
EndFor	
Return $temp$	
EndFunc	

$g()$	5000
-------	------

Cylind

Katalog > 

Vektor Cylind

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>**Cylind**.

Viser række- eller kolonnevektoren i cylindrisk form $[r\angle\theta, z]$.

Vektor skal have nøjagtig tre elementer. Det kan være en række eller en kolonne.

$\begin{bmatrix} 2 & 2 & 3 \end{bmatrix}$ ►Cylind	$\begin{bmatrix} 2\sqrt{2} & \angle \frac{\pi}{4} & 3 \end{bmatrix}$
---	--

cZeros()

Katalog > 

cZeros(Udtr, Var)⇒liste

Returnerer en liste med mulige reelle og ikke-reelle værdier for *Var* som løser $Udtr=0$. **cZeros()** gør dette ved at beregne **exp►liste(cSolve(Udtr=0,Var),Var)**. Ellers ligner **cZeros(zeros())**.

Bemærk: Se også **cSolve()**, **solve()** og **zeros()**.

cZeros({Udtr1, Udtr2 [, ...] }, {varEllerGæt1,varEllerGæt2 [, ...] })⇒matrix

I Vis cifre-tilstand med Fast 3:

$cZeros(x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3,x)$
$\{-1.114+1.073\cdot i, 1.114-1.073\cdot i, -2.125, -0.612, 0\}$

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ► til at bevæge markøren.

Returnerer mulige positioner, hvor udtrykkene samtidigt er nul. Hver *varEllerGæt* angiver en ubekendt, hvis værdi du søger.

Du kan også vælge at angive et initielt gæt til en variabel. Hvert *varEllerGæt* skal have formen:

variabel

– eller –

variabel = reelt eller ikke-reelt tal

For eksempel er x gyldig, og det er $x=3+io$ gså.

Hvis alle ligningerne er polynomier, og hvis du IKKE angiver nogle initielle gæt, benytter **cZeros()** den leksikale Gröbner/Buchberger eliminationsmetode som forsøg på at bestemme **alle** komplekse nulpunkter.

Komplekse nulpunkter kan omfatte både reelle og ikke-reelle nulpunkter som i eksemplet til højre.

Hver række i den resulterende matrix repræsenterer et nulpunkt med komponenterne arrangeret på samme måde som *varEllerGæt*-listen. Du kan udtrække en række og indeksere matricen efter *[række]*.

Sammenhørende polynomier kan have ekstra variable, der ikke har nogen værdier men repræsenterer givne numeriske værdier, der kan substitueres efterfølgende.

$$\left| \begin{array}{c} \text{cZeros}\left(\left\{u \cdot v - u - v, v^2 + u\right\}, \{u, v\}\right) \\ \left[\begin{array}{cc} 0 & 0 \\ \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \\ \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \end{array} \right] \end{array} \right|$$

Udtræk række 2:

$$\text{Ans}[2] \left[\begin{array}{cc} \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \end{array} \right]$$

$$\left| \begin{array}{c} \text{cZeros}\left(\left\{u \cdot v - u - c \cdot v^2, v^2 + u\right\}, \{u, v\}\right) \\ \left[\begin{array}{cc} 0 & 0 \\ -(c-1)^2 & -(c-1) \end{array} \right] \end{array} \right|$$

Du kan også medtage ubekendte variable, der ikke optræder i udtrykkene. Disse nulpunkter viser, hvordan nulpunktsfamilier kan indeholde arbitrære konstanter af formen ϵk , hvor k er et heltalsuffix fra 1 til og med 255.

For polynomielle systemer afhænger beregningstiden eller hukommelsesforbruget stærkt af den rækkefølge, de ubekendte angives i. Hvis det initiale valg kræver for meget hukommelse eller tålmodighed, skal du prøve at omarrangere variablene i udtrykkene og/eller *varEllerGæt* listen.

Hvis du ikke medtager nogen gæt, og hvis en ligning er ikke-polynomiel i en variabel, men alle udtryk er lineære i alle ubekendte, anvender **cZeros()** en Gauss-eliminering i et forsøg på at bestemme alle nulpunkter.

Hvis et system hverken er polynomiel i alle variable eller lineært i sine ubekendte, bestemmer **cZeros()** højst et nulpunkt med en iterativ approksimationsmetode. Dette gøres ved at lade antallet af ubekendte være lig med antallet af udtryk og forkorte alle andre variable i udtrykkene til tal.

Et ikke-reelt gæt er ofte nødvendigt for at bestemme ikke-reelle nulpunkter. For at opnå konvergens skal et gæt være meget tæt på et nulpunkt.

$$\left| \begin{array}{l} \text{cZeros}\left(\left\{u \cdot v - u - v, v^2 + u\right\}, \left\{u, v, w\right\}\right) \\ \text{cZero}\left(\left\{u \cdot (v-1) - v, u + v^2\right\}, \left\{u, v, w\right\}\right) \\ \left[\begin{array}{cc} 0 & 0 \\ \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \\ \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \end{array} \right] \cdot \left[\begin{array}{c} c4 \\ c4 \\ c4 \end{array} \right] \end{array} \right|$$

$$\left| \begin{array}{l} \text{cZeros}\left(\left\{u + v - e^w, u - v - i\right\}, \left\{u, v\right\}\right) \\ \left[\begin{array}{cc} \frac{e^w + i}{2} & \frac{e^w - i}{2} \end{array} \right] \end{array} \right|$$

$$\left| \begin{array}{l} \text{cZeros}\left(\left\{e^z - w, w - z^2\right\}, \left\{w, z\right\}\right) \\ [0.494866 \quad -0.703467] \end{array} \right|$$

$$\left| \begin{array}{l} \text{cZeros}\left(\left\{e^{-z} - w, w - z^2\right\}, \left\{w, z = 1 + i\right\}\right) \\ [0.149606 + 4.8919 \cdot i \quad 1.58805 + 1.54022 \cdot i] \end{array} \right|$$

D

dbd()

dbd(dato1,dato2)⇒værdi

Returnerer antallet af dage mellem *dato1* og *dato2* med tælling af faktiske dage.

dbd(12.3103,1.0104)	1
dbd(1.0107,6.0107)	151
dbd(3112.03,101.04)	1
dbd(101.07,106.07)	151

dato1 og *dato2* kan være tal eller lister med tal inden for området af datoer i en standardkalender. Hvis både *dato1* og *dato2* er lister, skal de have samme længde.

dato1 og *dato2* skal ligge mellem årene 1950 til 2049.

Du kan indtaste datoerne i to formater. Placeringen af decimaler er forskellen mellem datoformaterne.

MM.DDÅÅ (almindeligt format i USA)

DDMM.ÅÅ (almindeligt format i Europa)

►DD

Katalog >

Tal ►DD⇒*værdi*

I vinkeltilstanden Grader:

Liste1 ►DD⇒*liste*

(1.5°) ►DD	1.5°
$(45^\circ 22' 14.3'')$ ►DD	45.3706°
$(\{45^\circ 22' 14.3'', 60^\circ 0' 0''\})$ ►DD	$\{45.3706^\circ, 60^\circ\}$

Matrix1 ►DD⇒*matrix*

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>DD.

Returnerer den decimale ækvivalent til argumentet udtrykt i grader. Argumentet er et tal, en liste eller matrix, som efter den indstillede tilstand af Vinkel tolkes i grader, nygrader eller radianer.

I vinkeltilstanden Nygrader:

1►DD	$\frac{9}{10}^\circ$
------	----------------------

I vinkeltilstanden Radian:

(1.5) ►DD	85.9437°
-------------	----------

►Decimal

Katalog >

Udtryk1 ►Decimal⇒*udtryk*

$\frac{1}{3}$ ►Decimal	0.333333
------------------------	----------

Liste1 ►Decimal⇒*udtryk*

Matrix1 ►Decimal⇒*udtryk*

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>Decimal.

Viser argumentet i decimal form. Denne operator kan kun anvendes ved slutningen af indtastningslinjen.

Define

Define *Var* = *Udtryk*

Define *Funktion*(*Param1*, *Param2*, ...) = *Udtryk*

Definerer variabelen *Var* eller den brugerdefinerede funktion *Funktion*.

Parametre som *Param1* er pladsholdere til at sætte argumenter ind i funktionen. Ved kald af en brugerdefineret funktion skal du angive argumenter (for eksempel værdier eller variable), der svarer til parametrene. Når den kaldes, evaluerer funktionen *Udtryk* med de angivne argumenter.

Var og *Funktion* kan ikke være navnet på en systemvariabel eller en integreret funktion eller kommando.

Bemærk: Denne form for **Define** svarer til at eksekvere udtrykket: *udtryk* → *Funktion*(*Param1*,*Param2*).

Define *Funktion*(*Param1*, *Param2*, ...) = **Func**
Blok
EndFunc

Define *Program*(*Param1*, *Param2*, ...) = **Prgm**
Blok
EndPrgm

I denne form kan den brugerdefinerede funktion eller programmet eksekvere en blok med flere sætninger.

Define $g(x,y)=2 \cdot x-3 \cdot y$	Done
$g(1,2)$	-4
$1 \rightarrow a: 2 \rightarrow b: g(a,b)$	-4
Define $h(x)=\text{when}(x<2,2 \cdot x-3,-2 \cdot x+3)$	Done
$h(-3)$	-9
$h(4)$	-5

Define $g(x,y)=\text{Func}$	Done
If $x>y$ Then	
Return x	
Else	
Return y	
EndIf	
EndFunc	
$g(3,-7)$	3

Define

Katalog > 

Blok kan en være en enkelt sætning eller en række sætninger på separate linjer. *Blok* kan også rumme udtryk og kommandoer (som f.eks. **If**, **Then**, **Else** og **For**).

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

Bemærk: Se også **Define LibPriv**, side 50 og **Define LibPub**, side 50.

```
Define g(x,y)=Prgm
  If x>y Then
    Disp x," greater than ",y
  Else
    Disp x," not greater than ",y
  EndIf
EndPrgm
Done

g(3,-7)
3 greater than -7
Done
```

Define LibPriv

Katalog > 

Define LibPriv *Var* = *Udtryk*

Define LibPriv *Funktion*(*Param1*, *Param2*,
...) = *Udtryk*

Define LibPriv *Funktion*(*Param1*, *Param2*,
...) = **Func**
Blok
EndFunc

Define LibPriv *Program*(*Param1*, *Param2*,
...) = **Prgm**
Blok
EndPrgm

Fungerer på samme måde som **Define** med den undtagelse, at den definerer en privat biblioteksvariabel, funktion, eller et program. Private funktioner og programmer optræder ikke i Katalog.

Bemærk: Se også **Define**, side 49, og **Define LibPub**, side 50.

Define LibPub

Katalog > 

Define LibPub *Var* = *Udtryk*

Define LibPub *Funktion*(*Param1*, *Param2*,
...) = *Udtryk*

Define LibPub *Funktion*(*Param1*, *Param2*,

```
...) = Func
Blok
EndFunc
```

```
Define LibPub Program(Param1, Param2,
...) = Prgm
Blok
EndPrgm
```

Fungerer på samme måde som **Define** med den undtagelse, at den definerer en offentlig biblioteksvariabel, funktion, eller et program. Offentlige funktioner og programmer optræder i Katalog, når biblioteket er gemt eller opdateret.

Bemærk: Se også **Define**, side 49 og **Define LibPriv**, side 50.

deltaList()Se Δ List(), side 109.**deltaTmpCnv()**Se Δ tmpCnv(), side 201.**DelVar**

```
DelVar Var1[, Var2] [, Var3] ...
```

$2 \rightarrow a$	2
-------------------	---

```
DelVar Var.
```

$(a+2)^2$	16
-----------	----

Sletter de angivne variable, eller variabelgruppe fra hukommelse.

DelVar a	Done
$(a+2)^2$	$(a+2)^2$

Hvis en eller flere af variablene er låst, viser denne kommando en fejlmeddelelse og sletter kun de ulåste variable. Se **unLock**, side 209

DelVar

Katalog > 

DelVar *Var*. sletter alle elementer i *Var*. variabelgruppe(så som statistikken *stat.nn* resultater, eller variable dannet ved brug af **LibShortcut()**-funktionen). Punktummet (.) i denne form af **DelVar** -kommandoen begrænser den til at slette en variabelgruppe: den simple variabel *Var* berøres ikke.

$aa.a:=45$	45									
$aa.b:=5.67$	5.67									
$aa.c:=78.9$	78.9									
<code>getVarInfo()</code>	<table><tr><td>$aa.a$</td><td>"NUM"</td><td>"{"</td></tr><tr><td>$aa.b$</td><td>"NUM"</td><td>"{"</td></tr><tr><td>$aa.c$</td><td>"NUM"</td><td>"{"</td></tr></table>	$aa.a$	"NUM"	"{"	$aa.b$	"NUM"	"{"	$aa.c$	"NUM"	"{"
$aa.a$	"NUM"	"{"								
$aa.b$	"NUM"	"{"								
$aa.c$	"NUM"	"{"								
<code>DelVar aa.</code>	<i>Done</i>									
<code>getVarInfo()</code>	"NONE"									

delVoid()

Katalog > 

delVoid(*Liste1*) $\Rightarrow$ *liste*

<code>delVoid({1,void,3})</code>	{1,3}
----------------------------------	-------

Returnerer en liste med indholdet i *Liste1* med alle tomme (ugyldige) elementer fjernet.

Yderligere oplysninger om tomme elementer findes på side 266.

derivative()

Se *d()*, side 234.

deSolve()

Katalog > 

deSolve(*1.Eller2.OrdenODE, Var, afhVar*)
 $\Rightarrow$ en *generel løsning*

<code>deSolve(y''+2·y'+y=x²,x,y)</code>
$y = (c3 \cdot x + c4) \cdot e^{-x} + x^2 - 4 \cdot x + 6$
<code>right(Ans) → temp</code>
$\frac{d^2}{dx^2}(temp) + 2 \cdot \frac{d}{dx}(temp) + temp - x^2$
<code>DelVar temp</code>
<i>Done</i>

Returnerer en ligning, der eksplicit eller implicit angiver en generel løsning til en 1. eller anden ordens ordinær differentialligning (ODE). I ODE'en:

- Anvend et mærketegn (tryk på ) for at betegne differentialkvotienten af første orden af den afhængige variabel med hensyn til den uafhængige variable
- Anvend to mærketegn for at betegne den tilsvarende anden afledede.

Mærketegnet ' anvendes kun til differentialkvotienter i deSolve(). I andre tilfælde anvendes **dl()**.

Den generelle løsning til en 1. grads ligning indeholder en arbitrær konstant af formen $\cdot k$, hvor k er et heltalssuffiks fra 1 til og med 255. Løsningen af en 2. ordens differentialligning indeholder to af disse konstanter.

solve() anvendes på en implicit løsning, hvis du vil prøve at omregne den til en eller flere ækvivalente eksplicitte løsninger.

Ved sammenligning af dine resultater med tekstbogen eller manuelt frembragte løsninger skal du være opmærksom på, at forskellige metoder indfører arbitrære konstanter på forskellige steder i beregningen, hvilket kan frembringe forskellige generelle løsninger.

deSolve(1.OrdenODEandstartBeting, Var, afhVar) $\Rightarrow$ en partikulær løsning

Returnerer en partikulær løsning, som opfylder 1.OrdenODE og startBeting. Dette er normalt nemmere end at bestemme en generel løsning, substituere startværdier, løse for den arbitrære konstant og derefter substituere denne værdi ind i den generelle løsning.

startBeting er en ligning på formen:

afhVar (uafhængigStartværdi) =
afhængigStartværdi

uafhængigStartværdi og

afhængigStartværdi kan være variable som f.eks. x_0 og y_0 , der ikke har nogen lagrede værdier. Implicit differentiation kan hjælpe med at verificere implicitte løsninger.

$$\text{deSolve}\left(y' = (\cos(y))^2 \cdot x, x, y\right) \quad \tan(y) = \frac{x^2}{2} + c_4$$

$$\text{solve}(Ans, y) \quad y = \tan^{-1}\left(\frac{x^2 + 2 \cdot c_4}{2}\right) + n_3 \cdot \pi$$

$$Ans | c_4 = c - 1 \text{ and } n_3 = 0 \quad y = \tan^{-1}\left(\frac{x^2 + 2 \cdot (c - 1)}{2}\right)$$

$$\sin(y) = (y \cdot e^x + \cos(y)) \cdot y' \rightarrow ode$$

$$\sin(y) = (e^x \cdot y + \cos(y)) \cdot y'$$

$$\text{deSolve}(ode \text{ and } y(0) = 0, x, y) \rightarrow soln$$

$$\frac{-(2 \cdot \sin(y) + y^2)}{2} = (e^x - 1) \cdot e^{-x} \cdot \sin(y)$$

$$soln | x = 0 \text{ and } y = 0 \quad \text{true}$$

$$ode | y' = \text{impDif}(soln, x, y) \quad \text{true}$$

$$\text{DelVar } ode, soln \quad Done$$

deSolve

(2.OrdensODEandstartBeting1and startBeting2, Var, depVa)⇒ en partikulær løsning

Returnerer en partikulær løsning, der opfylder 2. ordens ODE og har en angivet værdi for den afhængige variable og dens første afledede i et punkt.

For *startBeting1*, brug formen:

depVar (uafhængigStartværdi) =
afhængigStartværdi

For *startBeting2*, brug formen:

afhVar (uafhængigStartværdi) = start
1.afledede

deSolve

(2.OrdensODEandbegrBeting1and begrBeting2, Var, depVar)⇒ en partikulær løsning

Returnerer en partikulær løsning, der opfylder 2. ordens ODE og har angivne værdier i to forskellige punkter.

$$\text{deSolve}\left(y''=y^{-\frac{1}{2}} \text{ and } y(0)=0 \text{ and } y'(0)=0, t, y\right)$$

$$\frac{\frac{3}{2} \cdot y^{\frac{4}{3}}}{3}=t$$

$$\text{solve}\left(\frac{\frac{3}{2} \cdot y^{\frac{4}{3}}}{3}=t, y\right)$$

$$y=\frac{\frac{1}{3} \cdot \frac{2}{3} \cdot \frac{4}{3}}{\frac{3 \cdot 3^{\frac{1}{3}} \cdot 2^{\frac{2}{3}} \cdot t^{\frac{3}{4}}}{4}} \text{ and } t \geq 0$$

$$\text{deSolve}(y''=x \text{ and } y(0)=1 \text{ and } y'(2)=3, x, y)$$

$$y=\frac{x^3}{6}+x+1$$

$$\text{deSolve}(y''=2 \cdot y' \text{ and } y(3)=1 \text{ and } y'(4)=2, x, y)$$

$$y=e^{2 \cdot x-8}-e^{-2}+1$$

$$\text{deSolve}\left(w''-\frac{2 \cdot w'}{x}+\left(9+\frac{2}{x^2}\right) \cdot w=x \cdot e^x \text{ and } w\left(\frac{\pi}{6}\right)=0 \text{ and } w\left(\frac{\pi}{3}\right)=0, x, w\right)$$

$$w=\frac{x \cdot e^x}{(\ln(e))^{2+9}}+\frac{e^{\frac{3}{2}} \cdot x \cdot \cos(3 \cdot x)}{(\ln(e))^{2+9}}-\frac{e^{\frac{6}{2}} \cdot x \cdot \sin(3 \cdot x)}{(\ln(e))^{2+9}}$$

det()Katalog > 

det(*kvadratMatrix*[,
Tolerance])⇒*udtryk*

Returnerer determinanten af
kvadratMatrix.

Ethvert matricelement kan valgfrit behandles som nul, hvis den absolutte værdi er mindre end *Tolerance*. Denne tolerance anvendes kun, hvis matricen har elementer med flydende decimaler og ikke indeholder symbolske variable, der ikke er tildelt en værdi. Ellers, *Tolerance* ignoreres.

- Hvis du anvender   eller indstiller **Auto eller tilnærmet** - tilstanden til Approximate, foretages beregningerne med aritmetik med flydende komma.
- Hvis *Tolerance* udelades eller ikke anvendes, beregnes standardtolerancen som:

$$5E-14 \cdot \max(\dim(kvadratMatrix)) \cdot \text{rowNorm}(kvadratMatrix)$$

$\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)$	$a \cdot d - b \cdot c$
$\det\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right)$	-2
$\det\left(\text{identity}(3) - x \cdot \begin{bmatrix} 1 & -2 & 3 \\ -2 & 4 & 1 \\ -6 & -2 & 7 \end{bmatrix}\right)$	$-(98 \cdot x^3 - 55 \cdot x^2 + 12 \cdot x - 1)$
$\begin{bmatrix} 1.E20 & 1 \\ 0 & 1 \end{bmatrix} \rightarrow mat1$	$\begin{bmatrix} 1.E20 & 1 \\ 0 & 1 \end{bmatrix}$
$\det(mat1)$	0
$\det(mat1, 1)$	1.E20

diag()Katalog > 

diag(*List*)⇒*matrix*

diag(*rækkeMatrix*)⇒*matrix*

diag(*kolonneMatrix*)⇒*matrix*

Returnerer en matrix med værdierne i argumentlisten eller matricen i hoveddiagonalen.

diag(*kvadratMatrix*)⇒*rækkeMatrix*

Returnerer en rækkematrix, der indeholder elementerne fra hoveddiagonalen i *kvadratMatrix*.

kvadratMatrix skal være kvadratisk.

$\text{diag}([2 \ 4 \ 6])$	$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix}$
$\begin{bmatrix} 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$	$\begin{bmatrix} 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$
$\text{diag}(Ans)$	$\begin{bmatrix} 4 & 2 & 9 \end{bmatrix}$

dim()	Katalog > 
dim(Liste)⇒heltal	$\dim(\{0,1,2\})$ 3
Returnerer dimensionen af <i>liste</i> .	
dim(matrix)⇒liste	$\dim\left(\begin{pmatrix} 1 & -1 \\ 2 & -2 \\ 3 & 5 \end{pmatrix}\right)$ {3,2}
Returnerer dimensionerne af matricen som en liste med to elementer {rækker, kolonner}.	
dim(Streng)⇒heltal	$\dim("Hello")$ 5
Returnerer det antal tegn, der er indeholdt i tegnstrengen <i>Streng</i> .	$\dim("Hello "&"there")$ 11

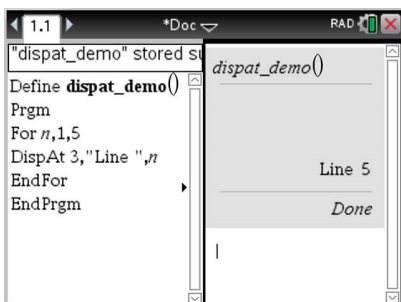
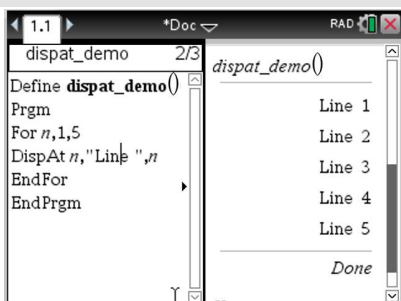
Disp	Katalog > 
Disp udtrykEllerStreng1 [, udtrykEllerStreng2] ...	Define <i>chars</i> (<i>start,end</i>)=Prgm
Viser argumenterne i <i>Calculator</i> historikken. Argumenterne vises efter hinanden med små mellemrum som separator.	For <i>i,start,end</i> Disp <i>i," "</i> ,char(<i>i</i>) EndFor EndPrgm
Anvendes hovedsagelig i programmer og funktioner til at sikre at mellemregninger vises.	Done
Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.	<i>chars</i> (240,243) ----- 240 ð 241 ñ 242 ò 243 ó ----- Done

DispAt	Katalog > 
DispAt int,expr1 [,expr2 ...] ...	DispAt
DispAt tillader dig at angive den linje, hvor det specificerede udtryk eller streng vil blive vist på skærmen.	Eksempel
Linjeantallet kan angives som et udtryk.	
Vær opmærksom på, at linjenummeret ikke gælder hele skærmen, men for området umiddelbart efter kommando/program.	

Denne kommando tillader dashboard-lignende output fra programmer, hvor værdien af et udtryk eller fra en sensor-aflysning bliver opdateret på den samme linje.

DispAt og **Disp** kan bruges indenfor det samme program.

Bemærk: Det maksimale antal er sat til 8, idet det passer til en hel skærm af linjer på den håndholdte skærm - sålænge linjerne ikke har 2D-matematiske udtryk. Det præcise antal linjer afhænger af indholdet af den viste information.



Konkrete eksempler:

Define z()=	Output
Prgm	z()
For n,1,3	Iteration 1:
DispAt 1,\"N: \",n	Linje 1: N:1
Disp \"Hallo\"	Linje 2: Hallo
EndFor	Iteration 2:
EndPrgm	Linje 1: N:2
	Linje 2: Hallo
	Linje 3: Hallo
	Iteration 3:
	Linje 1: N:3
	Linje 2: Hallo
	Linje 3: Hallo

	Linje 4: Hallo
Define z1()=	z1()
Prgm	Linje 1: N:3
For n,1,3	Linje 2: Hallo
DispAt 1,"N: ",n	Linje 3: Hallo
EndFor	Linje 4: Hallo
	Linje 5: Hallo
For n,1,4	
Disp "Hallo"	
EndFor	
EndPrgm	

Fejlbetinger:

Fejlmeddelelse	Beskrivelse
DispAt linjeantal skal være mellem 1 og 8	Udtryk evaluerer linjeantallet udenfor rækken 1-8 (inklusive)
For få argumenter	Funktionen eller kommandoen mangler et eller flere argumenter.
Ingen argumenter	Samme som aktuell syntaksfejl-dialog
For mange argumenter	Begræns argument. Samme fejl som Disp.
Ugyldig datatype	Første argument skal være et tal.
Ugyldig: DispAt ugyldig	"Hallo verden" datatypefejl er fundet ugyldig (hvis tilbagekald er defineret)
Konverteringsoperator: DispAt 2_ft @> _m, "Hallo verden"	CAS: Datatypefejl er fundet ugyldig (hvis tilbagekald er defineret) Numerisk: Konverteringen vil blive evalueret, og hvis resultatet er et gyldigt argument, vil DispAt printe strengen på resultatlinjen.

Udtr ►DMS

I vinkeltilstanden Grader:

List ►DMS

$\{45.371\}$ ►DMS	45°22'15.6"
$\{\{45.371,60\}\}$ ►DMS	$\{45°22'15.6",60°\}$

Matrix ►DMS

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>DMS.

Tolker argumentet som en vinkel og viser tilsvarende tal for grader (D), minutter (M) og sekunder (S/s) (DDDDDD°MM'SS.ss"). Se mere om DMS-formatet for grader, minutter og sekunder °, ', " på side 242.

Bemærk: ►DMS konverterer fra radianer til grader ved anvendelse i radianttilstanden. Hvis inputtet følges af et grader-symbol °, sker der ingen konvertering. Du kan kun anvende ►DMS ved slutningen af en indtastningslinje.

domain()

domain(*udtryk1*, *Var*)⇒*udtryk*

Beregner definitionsområdet for *udtryk1* med hensyn til *Var*.

domain() kan bruges til at undersøge definitionsområdet for funktioner. Den er begrænset til de reelle tal og til et endeligt antal intervaller.

Denne funktionalitet har begrænsninger pga. mangler i computeralgebraens reduktions- og ligningsløsningsalgoritmer.

Visse funktioner kan ikke bruges som argumenter for **domain()**, hvad enten de forekommer eksplicit eller inden for brugerdefinerede variable og funktioner. I det følgende eksempel kan udtrykket ikke reduceres, fordi $\int()$ ikke er en tilladt funktion.

$$\text{domain}\left(\int \begin{pmatrix} x \\ 1 \end{pmatrix} \frac{1}{t} dt, x\right) \rightarrow \text{domain}\left(\int \begin{pmatrix} x \\ 1 \end{pmatrix} \frac{1}{t} dt, x\right)$$

$$\text{domain}\left(\frac{1}{x+y}, y\right) \quad -\infty < y < -x \text{ or } -x < y < \infty$$

$$\text{domain}\left(\frac{x+1}{x^2+2 \cdot x}, x\right) \quad x \neq -2 \text{ and } x \neq 0$$

$$\text{domain}\left(\left(\sqrt{x}\right)^2, x\right) \quad 0 \leq x < \infty$$

$$\text{domain}\left(\frac{1}{x+y}, y\right) \quad -\infty < y < -x \text{ or } -x < y < \infty$$

dominantTerm(*Udtr1*, *Var* [, *Punkt*]) ⇒ *udtryk*

dominantTerm(*Udtr1*, *Var* [, *Punkt*]) | *Var* > *Punkt* ⇒ *udtryk*

dominantTerm(*Udtr1*, *Var* [, *Punkt*]) | *Var* < *Punkt* ⇒ *udtryk*

Returnerer det dominerende led i potensrækkeudviklingen af *Udtr1* udviklet omkring *Punkt*. Det dominerende led er det led, hvis der vokser hurtigst nær *Var* = *Punkt*. Den resulterende potens af (*Var* – *Punkt*) kan have en negativ og/eller brøkeksponent. Koefficienten af denne potens kan indeholde logaritmer af (*Var* – *Punkt*) og andre funktioner af *Var*, der domineres af alle potenser af (*Var* – *Punkt*) med samme eksponentfortegn.

Punkt er som standard 0. *Punkt* kan være ∞ eller $-\infty$, i hvilke tilfælde det dominerende led vil være det led, der har den største eksponent af *Var* frem for den mindste eksponent for *Var*.

dominantTerm(...) returnerer “**dominantTerm(...)**” hvis den ikke kan bestemme en sådan repræsentation, for eksempel for væsentlige singulariteter som $\sin(1/z)$ ved $z=0$, $e^{-1/z}$ ved $z=0$, eller e^z ved $z = \infty$ eller $-\infty$.

$\text{dominantTerm}(\tan(\sin(x)) - \sin(\tan(x)), x)$	$\frac{x^7}{30}$
$\text{dominantTerm}\left(\frac{1 - \cos(x-1)}{(x-1)^3}, x, 1\right)$	$\frac{1}{2 \cdot (x-1)}$
$\text{dominantTerm}\left(x^{-2} \cdot \tan\left(x^{\frac{1}{3}}\right), x\right)$	$\frac{1}{x^3}$
$\text{dominantTerm}(\ln(x^x - 1) \cdot x^{-2}, x)$	$\frac{\ln(x \cdot \ln(x))}{x^2}$

$\text{dominantTerm}\left(e^{\frac{-1}{z}}, z\right)$	$\text{dominantTerm}\left(e^{\frac{-1}{z}}, z, 0\right)$
$\text{dominantTerm}\left(\left(1 + \frac{1}{n}\right)^n, n, \infty\right)$	e
$\text{dominantTerm}\left(\tan^{-1}\left(\frac{1}{x}\right), x, 0\right)$	$\frac{\pi \cdot \text{sign}(x)}{2}$
$\text{dominantTerm}\left(\tan^{-1}\left(\frac{1}{x}\right), x, x > 0\right)$	$\frac{\pi}{2}$

Hvis rækken eller en af dens afledte afbrydes af et spring ved *Punkt*, indeholder resultatet sandsynligvis deludtryk på formen $\text{sign}(\dots)$ eller $\text{abs}(\dots)$ for en reel udviklingsvariabel eller $(-1)^{\text{floor}(\dots \text{angle}(\dots))}$ for en kompleks udviklingsvariabel, der slutter med “ $_$ ”. Hvis du kun vil bruge det dominerende led til værdier på den ene af *Punkt*, skal du til **dominantTerm(...)** tilføje den korrekte af funktionerne “ $| \text{Var} > \text{Punkt}$ ”, “ $| \text{Var} < \text{Punkt}$ ”, “ $| \text{Var} \geq \text{Punkt}$ ”, eller “ $\text{Var} \leq \text{Punkt}$ ” for at opnå et enklere resultat.

dominantTerm() distribuerer over 1. argument lister og matricer.

dominantTerm() er nyttig, når du vil have det enklest mulige udtryk, der nærmer sig asymptotisk til et andet udtryk som $\text{Var} \rightarrow \text{Punkt}$. **dominantTerm()** er også nyttig, når det ikke er klart, hvad graden af det første ikke-nul-led i en række vil være, og du ikke ønsker en række interaktive iterative gæt eller med en programløkke.

Bemærk: Se også **series()**, side 170.

dotP()

dotP(Liste1, Liste2)⇒udtryk

Returnerer “prik” produktet af to lister.

$\text{dotP}(\{a, b, c\}, \{d, e, f\})$	$a \cdot d + b \cdot e + c \cdot f$
$\text{dotP}(\{1, 2\}, \{5, 6\})$	17

dotP(Vektor1, Vektor2)⇒udtryk

Returner “prik” produktet af to vektorer.

$\text{dotP}([a \ b \ c], [d \ e \ f])$	$a \cdot d + b \cdot e + c \cdot f$
$\text{dotP}([1 \ 2 \ 3], [4 \ 5 \ 6])$	32

Begge skal være rækkevektorer, eller begge skal være kolonnevektorer.

e^()

e^x

-tast

e^(Udtr1)⇒udtryk

Returnerer e opløftet til potensen Udtr1.

Bemærk: Se også e Eksponentskabelon, side 2.

Bemærk: At trykke

ctrl

e^x

 for at vise e[^] (er ikke det samme som at trykke på tegnet

E

 på tastaturet.

Du kan indtaste et komplekst tal i rei θ polær form. Anvend dog kun denne form i vinkeltilstanden Radian. Den forårsager en domænefejl i vinkeltilstandene Grader eller Nygrader.

e^(Liste1)⇒liste

Returnerer e opløftet til potensen af hvert element i Liste1.

e^(kvadratMatrix1)⇒kvadratMatrix

Returnerer matrix eksponentialfunktion af kvadratMatrix1. Dette er ikke det samme som at beregne e opløftet til potensen af hvert element. Oplysninger om beregningsmetoden findes i cos().

KvadratMatrix1 skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

e¹

e

e^{1.}

2.71828

e^{3²}

e⁹

e{1,1.,0.5}

{e,2.71828,1.64872}

1

5

3

4

2

1

6

-2

1

782.209

559.617

456.509

680.546

488.795

396.521

524.929

371.222

307.879

eff()

Katalog >

a

2

eff(nominelRente,CpY)⇒værdi

Finansfunktion, der omregner den nominelle rente *nominelRente* til en effektiv årlig rente, hvor *CpY* er antallet af rentetilskrivninger per år.

nominelRente skal være et reelt tal, og *CpY* skal være et reelt tal > 0.

Bemærk: Se også *nom()*, side 130.

eff(5.75,12)

5.90398

62 Alfabetisk oversigt

eigVc(kvadratMatrix)⇒matrix

Returnerer en matrix med egenvektorerne for en reel eller kompleks *kvadratMatrix*, hvor hver kolonne i resultatet svarer til en egenværdi. Bemærk, at en egenvektor ikke er unik. Den kan skaleres af enhver konstantfaktor. Egenvektorerne er normaliseret, dvs. at hvis $V = [x_1, x_2, \dots, x_n]$, så:

$$x_1^2 + x_2^2 + \dots + x_n^2 = 1$$

kvadratMatrix balanceres først med similaritetstransformationer, til række- og kolonnenormer er så tæt som muligt på samme værdi. *KvadratMatrix* reduceres derefter til øvre Hessenberg form, og egenvektorerne beregnes via en Schur faktorisering.

I rektangulært komplekst format:

$$\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix} \rightarrow mI$$

$$\text{eigVc}(mI) \begin{bmatrix} -0.800906 & 0.767947 & (\\ 0.484029 & 0.573804+0.052258 \cdot i & 0.5738 \\ 0.352512 & 0.262687+0.096286 \cdot i & 0.2626 \end{bmatrix}$$

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

eigVI(kvadratMatrix)⇒liste

Returnerer en liste med egenværdier af en reel eller kompleks *kvadratMatrix*.

kvadratMatrix balanceres først med similaritetstransformationer, til række- og kolonnenormer er så tæt som muligt på samme værdi. *KvadratMatrix* reduceres derefter til øvre Hessenberg form, og egenværdierne beregnes fra øvre Hessenberg-matricen.

I rektangulær kompleks formattilstand:

$$\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix} \rightarrow mI$$

$$\text{eigVI}(mI) \{ -4.40941, 2.20471+0.763006 \cdot i, 2.20471-0.763006 \cdot i \}$$

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

```

If Boolsk Udtr1 Then
    Blok1
Elseif Boolsk Udtr2 Then
    Blok2
    :
Elseif Boolsk UdtrN Then
    BlokN
EndIf
    :

```

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

```

Define  $g(x)$ =Func
    If  $x \leq -5$  Then
        Return 5
    ElseIf  $x > -5$  and  $x < 0$  Then
        Return  $-x$ 
    ElseIf  $x \geq 0$  and  $x \neq 10$  Then
        Return  $x$ 
    ElseIf  $x = 10$  Then
        Return 3
    EndIf
EndFunc

```

*Done***EndFor****Se For, side 77.****EndFunc****Se Func, side 81.****EndIf****Se If, side 92.****EndLoop****Se Loop, side 116.****EndPrgm****Se Prgm, side 145.****EndTry****Se Try, side 202.**

euler ()

Katalog > 

euler(Expr, Var, depVar, {Var0, VarMax}, depVar0, VarStep [, eulerStep]) $\Rightarrow$ matrix

euler(SystemOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep [, eulerStep]) $\Rightarrow$ matrix

euler(ListOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep [, eulerStep]) $\Rightarrow$ matrix

Anvender Euler-metoden til at løse systemet

$$\frac{d \text{ depVar}}{d \text{ Var}} = \text{Expr}(\text{Var}, \text{depVar})$$

med $\text{depVar}(\text{Var0}) = \text{depVar0}$ i intervallet $[\text{Var0}, \text{VarMax}]$. Returnerer en matrix, hvor første række definerer Var-outputværdierne, og anden række definerer værdien af første løsningskomponent ved de tilsvarende Var-værdier, osv.

Expr er højresiden, som definerer den ordinære differentialligning (ODE - ordinary differential equation).

SystemOfExpr er systemet af højresider, der definerer ODE'erne (svarer til rækkefølgen af afhængige variable i ListOfDepVars).

ListOfExpr er en liste af højresider, der definerer systemet af ODE'er (svarer til rækkefølgen af afhængige variable i ListOfDepVars).

Var er den uafhængige variable.

ListOfDepVars er en liste med afhængige variable.

Differentialligning:

$$y' = 0.001 \cdot y \cdot (100 - y) \text{ og } y(0) = 10$$

$$\text{euler}\left(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1\right)$$

0.	1.	2.	3.	4.
10.	10.9	11.8712	12.9174	14.042

Du kan se hele resultatet ved at trykke på $\blacktriangle$ og derefter bruge $\blacktriangleleft$ og $\blacktriangleright$ til at bevæge markøren.

Sammenlign resultatet ovenfor med den eksakte CAS løsning opnået ved brug af deSolve() og seqGen():

$$\text{deSolve}\left\{y' = 0.001 \cdot y \cdot (100 - y) \text{ and } y(0) = 10, t, y\right\}$$

$$y = \frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9}$$

$$\text{seqGen}\left(\frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9}, t, y, \{0, 100\}\right)$$

$$\{10., 10.9367, 11.9494, 13.0423, 14.2189\}$$

System af ligninger:

$$\begin{cases} y1' = -y1 + 0.1 \cdot y1 \cdot y2 \\ y2' = 3 \cdot y2 - y1 \cdot y2 \end{cases}$$

med $y1(0) = 2$ og $y2(0) = 5$

$$\text{euler}\left(\begin{cases} y1' + 0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{cases}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1\right)$$

0.	1.	2.	3.	4.	5.
2.	1.	1.	3.	27.	243.
5.	10.	30.	90.	90.	-2070.

$\{Var0, VarMax\}$ er en liste med to elementer, der informerer funktionen om, at integrere fra $Var0$ til $VarMax$.

$ListOfDepVars0$ er en liste med startværdier for afhængige variable.

$VarStep$ er et tal forskelligt fra nul, således at $\text{sign}(VarStep) = \text{sign}(VarMax - Var0)$ og løsninger returneres i $Var0 + i \cdot VarStep$ for alle $i=0,1,2,\dots$ således at $Var0 + i \cdot VarStep$ er i $[var0, VarMax]$ (der vil muligvis ikke være en løsningsværdi ved $VarMax$).

$eulerStep$ er et positivt heltal (som standard er 1), der definerer antallet af euler-trin mellem outputværdier. Den faktiske trinstørrelse, som euler-metoden anvender, er $VarStep / eulerStep$.

eval ()

Hub-menu

$\text{eval}(Expr) \Rightarrow string$

eval() er kun gyldig i TI-Innovator™ Hub Kommandoargument for programmeringskommandoerne **Get**, **GetStr** og **Send**. Softwaren evaluerer udtrykket *Expr* og udskifter **eval()**-udsagnet med resultatet som en tekststreng.

Argumentet *Expr* skal blot være et reelt tal.

Sæt det blå element i RGB LED til halv intensitet.

$lum := 127$	127
Send "SET COLOR.BLUE eval(lum)"	Done

Nulstil det blå element til OFF.

Send "SET COLOR.BLUE OFF"	Done
---------------------------	------

eval()-argument skal blot være et reelt tal.

Send "SET LED eval("4") TO ON"	"Error: Invalid data type"
--------------------------------	----------------------------

Programmet vil langsomt indføre det røde element

```

Define fadein()=
Prgm
For i,0,255,10
Send "SET COLOR.RED eval(i)"
Wait 0.1
EndFor
Send "SET COLOR.RED OFF"
EndPrgm

```

Kør programmet.

<i>fadein()</i>	Done
<i>n:=0.25</i>	0.25
<i>m:=8</i>	8
<i>n·m</i>	2.
Send "SET COLOR.BLUE ON TIME eval(n·m)"	Done
<i>iostr.SendAns</i>	"SET COLOR.BLUE ON TIME 2"

Selvom **eval()** ikke viser et resultat, kan du se den resulterende Hub-kommandostreng, efter at du har udført kommandoen ved at inspicere en af følgende specielle variable.

iostr.SendAns
iostr.GetAns
iostr.GetStrAns

Bemærk: Se også **Get** (side 83), **GetStr** (side 90), og **Send** (side 167).

exact()

Katalog >

exact(*Udtr1* [, *Tolerance*])⇒*udtryk*

exact (*Liste1* [, *Tolerance*])⇒*liste*

exact (*Matrix1* [, *Tolerance*])⇒*matrix*

Anvender Eksakt-tilstandens aritmetik til at returnere det rationale talækvivalent til argumentet, hvis det er muligt.

Tolerance angiver tolerancen for konverteringen; standardindstillingen er 0 (nul).

exact (0.25)	$\frac{1}{4}$
exact (0.333333)	$\frac{333333}{1000000}$
exact (0.333333,0.001)	$\frac{1}{3}$
exact (3.5·x+y)	$\frac{7 \cdot x}{2} + y$
exact ({0.2,0.33,4.125})	$\left\{ \frac{1}{5}, \frac{33}{100}, \frac{33}{8} \right\}$

Exit

Katalog >

Exit

Funktionsliste:

Afslutter den aktuelle **For**, **While**, eller **Loop**-blok.

Exit	Katalog > 	
Exit er ikke tilladt uden for de tre løkkestrukturer (For , While , eller Loop).		
Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.		
	Define $g()$ =Func	Done
	Local $temp,i$	
	$0 \rightarrow temp$	
	For $i,1,100,1$	
	$temp+i \rightarrow temp$	
	If $temp>20$ Then	
	Exit	
	EndIf	
	EndFor	
	EndFunc	
	$g()$	21

$\blacktriangleright$ exp	Katalog > 	
<i>Udtr</i> $\blacktriangleright$ exp	$\frac{d}{dx}(e^x + e^{-x})$	$2 \cdot \sinh(x)$
Repræsenterer <i>Udtr</i> i termer af den naturlige eksponentialfunktion <i>e</i> . Det er en konverteringsoperator til visning. Denne operator kan kun anvendes ved slutningen af indtastningslinjen.	$2 \cdot \sinh(x) \blacktriangleright$ exp	$e^x - e^{-x}$
Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>exp.		

exp()	 -tast	
$\exp(Udtr1) \Rightarrow udtryk$	e^1	e
Returnerer e opløftet til potensen <i>Udtr1</i> .	$e^1.$	2.71828
Returnerer e opløftet til potensen <i>Værdi1</i> .	e^{3^2}	e^9
Bemærk: Se også e eksponentskabelon, side 2.		
Du kan indtaste et komplekst tal i rel θ polær form. Anvend dog kun denne form i vinkeltilstanden Radian. Den forårsager en domænefejl i vinkeltilstandene Grader eller Nygrader.		
$\exp(Liste1) \Rightarrow liste$	$e^{\{1,1.,0.5\}}$	$\{e,2.71828,1.64872\}$
Returnerer e opløftet til potensen af hvert element i <i>Liste1</i> .		

exp()

e^x-tast

exp(kvadratMatrix1)⇒kvadratMatrix

Returnerer matrix eksponentialfunktion af *kvadratMatrix1*. Dette er ikke det samme som at beregne **e** opløftet til potensen af hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

KvadratMatrix1 skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

$$e \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 782.209 & 559.617 & 456.509 \\ 680.546 & 488.795 & 396.521 \\ 524.929 & 371.222 & 307.879 \end{bmatrix}$$

exp►liste()

Katalog >

exp►list(Udtr,Var)⇒liste

Undersøger *Udtr* for ligninger, der er adskilt af ordet "or", og returnerer en liste med ligningernes højresider på formen *Var=udtryk*. Dette er en nem metode til at udtrække løsningsværdier, der er indlejret i resultaterne af funktionerne **solve()**, **cSolve()**, **fMin()**, og **fMax()**.

Bemærk: **exp►list()** er ikke nødvendig med funktionerne **zeros** og **cZeros()**, da de direkte returnerer en liste med løsningsværdier.

Du kan indsætte denne funktion fra computerens tastatur ved at skrive **exp@>list(...)**.

$$\begin{array}{l} \text{solve}(x^2-x-2=0,x) \qquad \qquad \qquad x=-1 \text{ or } x=2 \\ \text{exp►list}(\text{solve}(x^2-x-2=0,x),x) \qquad \qquad \{ -1,2 \} \end{array}$$

expand()

Katalog >

expand(Udtr1 [, Var])⇒udtryk

expand(Liste1 [,Var])⇒liste

expand(Matrix1 [,Var])⇒matrix

expand(Udtr1) returnerer *Udtr1* på ledform med hensyn til alle variable. Ledformen er en polynomiumsudvikling for polynomier og en udvikling i partialbrøker for polynomiumsbrøker.

$$\begin{array}{l} \text{expand}\left((x+y+1)^2\right) \\ \qquad \qquad \qquad x^2+2\cdot x\cdot y+2\cdot x+y^2+2\cdot y+1 \\ \text{expand}\left(\frac{x^2-x+y^2-y}{x^2\cdot y^2-x^2\cdot y-x\cdot y^2+x\cdot y}\right) \\ \qquad \qquad \qquad \frac{1}{x-1}-\frac{1}{x}+\frac{1}{y-1}-\frac{1}{y} \end{array}$$

Målet for **expand()** er at transformere *Udtr1* til en sum og/eller differens for simple led. Som modsætning er målet for **factor()** at transformere *Udtr1* til et produkt og/eller kvotient af simple faktorer.

expand(*Udtr1*,*Var*) returnerer *Udtr1* på ledform med hensyn til *Var*. Ens potenser af *Var* samles. Leddene og deres faktorer sorteres med *Var* som hovedvariabel. Der kan opstå en vis utilsigtet faktoropløsning eller udvikling af de reducerede koefficienter. Sammenlignet med at udelade *Var* sparer dette ofte tid, hukommelse og skærmlads, samtidig med, at det gør udtrykket mere læseligt.

Selv når der kun er en variabel, kan anvendelsen af *Var* gøre faktoropløsningen af nævneren til delvis udvikling af brøker mere fuldstændig.

Tip: For rationale udtryk er **propFrac()** et hurtigere men mindre vidtgående alternativ til **expand()**.

Bemærk: Se også **comDenom()** vedrørende en tæller på ledform over en nævner på ledform.

expand(*Udtr1*, [*Var*]) distribuerer også logaritmer og brøkpotenser, uanset *Var*. For øget distribution af logaritmer og brøkpotenser kan grænser for uligheder være nødvendige for at garantere, at visse faktorer er ikke-negative.

expand(*Udtr1*, [*Var*]) distribuerer også absolutte værdier, **sign()** og eksponentialfunktioner uanset *Var*.

Bemærk: Se også **tExpand()** vedrørende trigonometriske additionsformler og formler for multipel vinkel.

$$\begin{array}{l} \text{expand}\left((x+y+1)^2, y\right) \quad y^2+2\cdot y\cdot(x+1)+(x+1)^2 \\ \text{expand}\left((x+y+1)^2, x\right) \quad x^2+2\cdot x\cdot(y+1)+(y+1)^2 \\ \text{expand}\left(\frac{x^2-x+y^2-y}{x^2\cdot y^2-x^2\cdot y-x\cdot y^2+x\cdot y}, y\right) \\ \quad \frac{1}{y-1}-\frac{1}{y}+\frac{1}{x\cdot(x-1)} \\ \text{expand}(Ans, x) \quad \frac{1}{x-1}-\frac{1}{x}+\frac{1}{y\cdot(y-1)} \\ \text{expand}\left(\frac{x^3+x^2-2}{x^2-2}\right) \quad \frac{2\cdot x}{x^2-2}+x+1 \\ \text{expand}(Ans, x) \quad \frac{1}{x-\sqrt{2}}+\frac{1}{x+\sqrt{2}}+x+1 \end{array}$$

$$\begin{array}{l} \ln(2\cdot x\cdot y)+\sqrt{2\cdot x\cdot y} \quad \ln(2\cdot x\cdot y)+\sqrt{2\cdot x\cdot y} \\ \text{expand}(Ans) \quad \ln(x\cdot y)+\sqrt{2\cdot x\cdot y}+\ln(2) \\ \text{expand}(Ans)|y\geq 0 \quad \ln(x)+\sqrt{2\cdot x}\cdot\sqrt{y}+\ln(y)+\ln(2) \\ \text{sign}(x\cdot y)+|x\cdot y|+e^{2\cdot x+y} \\ \quad e^{2\cdot x+y+\text{sign}(x\cdot y)+|x\cdot y|} \\ \text{expand}(Ans) \quad \text{sign}(x)\cdot\text{sign}(y)+|x|\cdot|y|+(e^x)^2\cdot e^y \end{array}$$

expr()	Katalog > 	
expr(<i>Streng</i>) ⇒ <i>udtryk</i>		
Returnerer den tegnstreng, der er indeholdt i <i>Streng</i> som et udtryk og eksekverer den straks.		
	expr("1+2+x^2+x")	x^2+x+3
	expr("expand((1+x)^2)")	$x^2+2\cdot x+1$
	"Define cube(x)=x^3" → <i>funcstr</i>	
	"Define cube(x)=x^3"	
	expr(<i>funcstr</i>)	<i>Done</i>
	cube(2)	8

ExpReg	Katalog > 	
ExpReg <i>X</i> , <i>Y</i> [, [<i>Frekv</i>][, <i>Kategori</i> , <i>Medtag</i>]]		

Beregner polynomielle tredjegradsregression $y = a \cdot (b)^x$ på listerne *X* og *Y* med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

X og *Y* er lister med uafhængige og afhængige variable.

Frekv er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende *X* og *Y* datapunkt. Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

Kategori er en liste, der indeholder kategorikoder for *X* og *Y* data.

Medtag er en liste med en eller flere af kategorikoderne. Kun de dataelementer, hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot (b)^x$
stat.a, stat.b	Regressionskoefficienter
stat.r ²	Koefficient af en lineær forklaringsgrad til transformerede data

Output-variablel	Beskrivelse
stat.r	Korrelationskoefficient til transformerede data ($x, \ln(y)$)
stat.Resid	Residualer af kurvetilpasningen $= y - a \cdot (b)^x$
stat.ResidTrans	Residualer associeret med lineær tilpasning af transformerede data
stat.XReg	Liste af datapunkter i den modificerede <i>X-liste</i> , der faktisk bruges i regressionen ud fra begrænsninger af <i>Hyppighed</i> , <i>Kategoriliste</i> og <i>Medtag kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y-liste</i> , der faktisk bruges i regressionen ud fra begrænsninger af <i>Hyppighed</i> , <i>Kategoriliste</i> og <i>Medtag kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

F

factor()

Katalog > 

factor(*Udtr1*[, *Var*]) $\Rightarrow$ *udtryk*

factor(*Liste1*[, *Var*]) $\Rightarrow$ *liste*

factor(*Matrix1*[, *Var*]) $\Rightarrow$ *matrix*

factor(*Udtr1*) returnerer *Udtr1* opløst i faktorer med hensyn til alle dens variable over en fællesnævner.

Udtr1 opløses mest muligt mod lineære rationale faktorer uden at indføre nye ikke-reelle deludtryk. Denne mulighed er velegnet, hvis du ønsker opløsning i faktorer med hensyn til mere end en variabel.

factor(*Udtr1*, *Var*) returnerer *Udtr1* opløst i faktorer med hensyn til variabelen *Var*.

Udtr1 opløses mest muligt i faktorer, der er lineære i *Var*, også hvis den indfører irrationale konstanter eller deludtryk, der er irrationale i andre variable.

$$\begin{array}{l} \text{factor}\left(a^3 \cdot x^2 - a \cdot x^2 - a^3 + a\right) \\ a \cdot (a-1) \cdot (a+1) \cdot (x-1) \cdot (x+1) \\ \text{factor}\left(x^2+1\right) \\ x^2+1 \\ \text{factor}\left(x^2-4\right) \\ (x-2) \cdot (x+2) \\ \text{factor}\left(x^2-3\right) \\ x^2-3 \\ \text{factor}\left(x^2-a\right) \\ x^2-a \end{array}$$

$$\begin{array}{l} \text{factor}\left(a^3 \cdot x^2 - a \cdot x^2 - a^3 + a, x\right) \\ a \cdot \left(a^2-1\right) \cdot (x-1) \cdot (x+1) \\ \text{factor}\left(x^2-3, x\right) \\ (x+\sqrt{3}) \cdot (x-\sqrt{3}) \\ \text{factor}\left(x^2-a, x\right) \\ (x+\sqrt{a}) \cdot (x-\sqrt{a}) \end{array}$$

Faktorerne og deres led sorteres med *Var* som hovedvariabel. Ens potenser af *Var* samles i hver faktor. Medtag *Var*, hvis opløsning i faktorer kun er nødvendig med hensyn til den pågældende variabel, og du er villig til at acceptere irrationale udtryk i alle andre variable for at øge opløsningen i faktorer med hensyn til *Var*. Der kan forekomme en uforudset faktoropløsning med hensyn til andre variable.

For Auto-indstillingen af

Auto eller tilnærmet tilstanden muliggør medtagelsen af *Var* også en approksimering med koefficienter med flydende decimaler, hvor irrationale koefficienter ikke kan udtrykkes med de indbyggede funktioner. Også når der kun er en variabel, kan en medtagelse af *Var* give en mere komplet opløsning i faktorer.

Bemærk: Se også **comDenom()** for en hurtig metode til at opnå partiel faktoropløsning, når **factor()** ikke er hurtig nok, eller den bruger hele hukommelsen.

Bemærk: Se også **cFactor()** om faktoropløsning med komplekse koefficienter for at opnå lineære faktorer.

factor(rationalTal) returnerer det rationale tal opløst i primtal. Ved sammensatte tal øges beregningstiden eksponentielt med antallet af cifre i den næststørste faktor. Opløsning af et 30-cifret heltal kan for eksempel være længere end en dag, og opløsning af et 100-cifret tal kan være længere end et århundrede.

Sådan stopper du en beregning manuelt,

- **Håndholdt:** Hold tasten  nede, mens du gentagne gange trykker på

$$\frac{\text{factor}(x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3)}{x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3}$$

$$\frac{\text{factor}(x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3, x)}{(x - 0.964673) \cdot (x + 0.611649) \cdot (x + 2.12543) \cdot (x^2 + 1.964673x + 0.964673)}$$

<code>factor(152417172689)</code>	123457 · 1234577
<code>isPrime(152417172689)</code>	false

enter .

- **Windows®:** Hold tasten **F12** nede, mens du gentagne gange trykker på **Enter**.
- **Macintosh®:** Hold tasten **F5** nede, mens du gentagne gange trykker på **Enter**.
- **iPad®:** App'en viser en meddelelse. Du kan fortsat vente eller annullere.

Hvis du kun vil bestemme, om et tal er et primtal, skal du anvende **isPrime()** i stedet. Det er meget hurtigere, især hvis *RationalTal* ikke er et primtal, og den næststørste faktor har mere end fem cifre.

F Cdf()**F Cdf**

(nedreGrænse, øvreGrænse, fgTæller, fgNævner) ⇒ tal hvis *nedreGrænse* og *øvreGrænse* er tal, *liste* hvis *nedreGrænse* og *øvreGrænse* er lister

FCdf

(nedreGrænse, øvreGrænse, fgTæller, fgNævner) ⇒ tal hvis *nedreGrænse* og *øvreGrænse* er tal, *liste* hvis *nedreGrænse* og *øvreGrænse* er lister

Beregner sandsynligheden hørende til **F**-fordelingen for intervallet mellem *nedreGrænse* og *øvreGrænse* med de angivne frihedsgrader *dfTæller* og *dfNævner*.

For den kumulerede fordeling $P(X \leq \text{øvreGrænse})$ skal du sætte *nedreGrænse* = 0.

Fill

Fill *Udtr*, *matrixVar* ⇒ *matrix*

Erstatter hvert element i variabelen *matrixVar* med *Udtr*.

matrixVar skal eksistere i forvejen.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow amatrix$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
Fill 1.01, amatrix	Done
amatrix	$\begin{bmatrix} 1.01 & 1.01 \\ 1.01 & 1.01 \end{bmatrix}$

Fill
Katalog >

Fill $Udtr, listeVar \Rightarrow liste$

$\{1,2,3,4,5\} \rightarrow alist$
 $\{1,2,3,4,5\}$

Erstatter hvert element i variablen $Listevar$ med $Udtr$.

Fill 1.01, $alist$
Done

$alist$
 $\{1.01,1.01,1.01,1.01,1.01\}$

$Listevar$ skal eksistere i forvejen.

FiveNumSammendrag
Katalog >

FiveNumSummary X [, [$Frekv$][$Kategori, Medtag$]]

Frembringer en forkortet version af 1-variabelstatistikken på listen X . En sammenfatning af resultaterne lagres i $stat.results$ variable. (side 187.)

X repræsenterer en liste med dataene.

$Frekv$ er en valgfri liste med hyppigheder. Hvert element i $Frekv$ angiver hyppigheden af hver tilsvarende X værdi. Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

$Kategori$ er en liste med numeriske kategorikoder for tilsvarende X værdier.

$Medtag$ er en liste med en eller flere af kategorikoderne. Kun de dataelementer, hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Et tomt (ugyldigt) element i en af listerne X , $Freq$ eller $Category$ resulterer i at det tilsvarende element i alle disse lister bliver ugyldigt. Yderligere oplysninger om tomme elementer findes på side 266

Output-variabel	Beskrivelse
stat.MinX	Minimum af x-værdier
stat.Q1X	1. kvartil af x
stat.MedianX	Median af x
stat.Q3X	3. kvartil af x
stat.MaxX	Maksimum af x-værdier

floor()Katalog > **floor**(*Udtr1*) $\Rightarrow$ *heltal*

$\text{floor}(-2.14)$	-3.
-----------------------	-----

Returnerer det største heltal, der $\leq$ argumentet. Denne funktion er identisk med **int()**.

Argumentet kan være et reelt eller komplekst tal.

floor(*Liste1*) $\Rightarrow$ *liste*

$\text{floor}\left(\left\{\frac{3}{2}, 0, -5.3\right\}\right)$	$\{1, 0, -6\}$
--	----------------

floor(*Matrix1*) $\Rightarrow$ *matrix*

$\text{floor}\left(\begin{bmatrix} 1.2 & 3.4 \\ 2.5 & 4.8 \end{bmatrix}\right)$	$\begin{bmatrix} 1. & 3. \\ 2. & 4. \end{bmatrix}$
---	--

Returnerer en liste eller matrix med nedrunding af hvert element.

Bemærk: Se også **ceiling()** og **int()**.

fMax()Katalog > **fMax**(*Udtr*, *Var*) $\Rightarrow$ *Boolsk udtryk*

$\text{fMax}\left(1-(x-a)^2-(x-b)^2, x\right)$	$x=\frac{a+b}{2}$
--	-------------------

fMax(*Udtr*, *Var*, *nedreGrænse*)

$\text{fMax}\left(.5 \cdot x^3 - x - 2, x\right)$	$x=\infty$
---	------------

fMax(*Udtr*,
Var, *nedreGrænse*, *øvreGrænse*)**fMax**(*Udtr*, *Var*) | *nedreGrænse* $\leq$ *Var*
 $\leq$ *øvreGrænse*

Returnerer et Boolsk udtryk, der angiver mulige værdier for *Var*, der maksimerer *udtryk* eller finder dets mindste øvre grænse.

Med ("|") betingelses-operatoren kan du indskrænke løsningsintervallet og/eller angive andre begrænsninger.

$\text{fMax}\left(0.5 \cdot x^3 - x - 2, x\right) x \leq 1$	$x=-0.816497$
---	---------------

I Tilnærmet-indstillingen af

Auto eller **tilnærmet**-tilstanden søger **fMax()** iterativt efter et tilnærmet lokalt maksimum. Dette er ofte hurtigere, især hvis du anvender "|" -operatoren til at begrænse søgningen til et relativt lille interval, der indeholder nøjagtigt et lokalt maksimum.

Bemærk: Se også **fMin()** og **max()**.

fMin(*Udtr*, *Var*) $\Rightarrow$ Boolsk udtryk

fMin(*Udtr*, *Var*, *nedreGrænse*)

fMin(*Udtr*,
Var, *nedreGrænse*, *øvreGrænse*)

fMin(*Udtr*, *Var*) | *nedreGrænse* $\leq$ *Var*
 $\leq$ *øvreGrænse*

Returnerer et Boolsk udtryk, der angiver mulige værdier for *var*, der minimerer *Udtr* eller finder dets største nedre grænse.

Med ("|") betingelses-operatoren kan du indskrænke løsningsintervallet og/eller angive andre begrænsninger.

Til Approx-indstillingen af **Auto eller tilnærmet** tilstanden søger **fMin()** iterativt for et enkelt tilnærmet lokalt minimum. Dette er ofte hurtigere, især hvis du anvender "|" -operatoren til at begrænse søgningen til et relativt lille interval, der indeholder nøjagtigt et lokalt minimum.

Bemærk: Se også **fMax()** og **min()**.

$fMin(1-(x-a)^2-(x-b)^2, x)$	$x=-\infty$ or $x=\infty$
$fMin(0.5 \cdot x^3 - x - 2, x) x \geq 1$	$x=1.$

For

For *Var*, *Lav*, *Høj* [, *Trin*]

Blok

EndFor

Eksekverer sætningerne i *blok* iterativt for hver værdi af *Var* fra *Lav* til *Høj* i intervaller på *Trin*.

Var må ikke være en systemvariabel.

Trin kan være positiv eller negativ.
Standardværdien er 1.

Blok kan enten være en enkelt sætning eller en serie sætninger adskilt med kolon.

Define $g()$ = Func	Done
Local <i>tempsum</i> , <i>step</i> , <i>i</i>	
$0 \rightarrow tempsum$	
$1 \rightarrow step$	
For <i>i</i> , 1, 100, <i>step</i>	
$tempsum + i \rightarrow tempsum$	
EndFor	
EndFunc	
$g()$	5050

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

format()

format(*Udtr*[, *formatStreng*])⇒*streng*

Returnerer *Udtr* som en tegnstring baseret på formatskabelonen.

Udtr skal kunne omregnes til et tal.

formatStreng er en streng og skal være på formen: "F[n]", "S[n]", "E[n]", "G[n][c]", hvor [] angiver valgfrie dele.

F[n]: Fast format. n er det antal cifre, der vises efter decimalpunktet.

S[n]: Videnskabeligt format. n er det antal cifre, der vises efter decimalpunktet.

E[n]: Teknisk format. n er antallet af cifre efter det betydende ciffer. Eksponenter er justeret til et multiplum af tre, og decimalpunktet flyttes til højre med nul, en eller to pladser.

G[n][c]: Samme som fast format men skiller også cifrene til venstre for decimalpunktet i grupper på tre. c angiver gruppeskilletegnet og er som standard et komma. Hvis c er et punktum, vises grundtallet som et komma.

[Rc]: Alle ovennævnte angivelser kan udvides med Rc-grundtalflaget, hvor c er et enkelt tegn, der angiver, hvad der skal substitueres for grundtalspunktet.

format(1.234567, "f3")	"1.235"
format(1.234567, "s2")	"1.23E0"
format(1.234567, "e3")	"1.235E0"
format(1.234567, "g3")	"1.235"
format(1234.567, "g3")	"1,234.567"
format(1.234567, "g3,r")	"1:235"

fPart()Katalog > **fPart**(*Udtr1*) $\Rightarrow$ *udtryk* $\text{fPart}(-1.234)$ -0.234**fPart**(*Liste1*) $\Rightarrow$ *liste* $\text{fPart}(\{1, -2.3, 7.003\})$ $\{0, -0.3, 0.003\}$ **fPart**(*Matrix1*) $\Rightarrow$ *matrix*

Returnerer decimaldelen af argumentet.

For en liste eller matrix returneres decimaldelen af elementerne.

Argumentet kan være et reelt eller komplekst tal.

FPdf()Katalog > **FPdf**(*XVærdi*,*dfTæller*,*dfNævner*)**FPdf**(*XVærdi*,*dfTæller*,*dfNævner*)

Beregner F sandsynlighedsfordelingen på *XVal* for den angivne *dfTæller* (frihedsgrader i tælleren) og *dfNævner* (frihedsgrader i nævneren).

freqTable►list()Katalog > **freqTable►liste****(Liste1**,*frekvHeltalListe*) $\Rightarrow$ *liste*
 $\text{freqTable} \blacktriangleright \text{list}(\{1, 2, 3, 4\}, \{1, 4, 3, 1\})$
 $\{1, 2, 2, 2, 2, 3, 3, 3, 4\}$

Returnerer en liste indeholdende elementerne fra *Liste1* udvidet i henhold til hyppighederne i *frekvHeltalListe*. Denne funktion kan anvendes til at danne en frekvenstabel for Data- & Statistikapplikationer.

 $\text{freqTable} \blacktriangleright \text{list}(\{1, 2, 3, 4\}, \{1, 4, 0, 1\})$
 $\{1, 2, 2, 2, 4\}$
Liste1 kan være enhver gyldig liste.

frekvHeltalListe skal have den samme dimension som *Liste1* og må kun indeholde ikke-negative heltalselementer. Hvert element angiver det antal gange det tilsvarende *Liste1* element vil blive gentaget i resultatlisten. En nul-værdi udelukker det tilsvarende *Liste1* element.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive `freqTable@>list(...)`.

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 266.

frequency()

frequency(Liste1, binsListe) ⇒ liste

Returnerer en liste, der indeholder optælling af elementerne i *Liste1*. Antallene er baseret på områder (bins), som du definerer i *binsListe*.

Hvis *binsListe* er $\{b(1), b(2), \dots, b(n)\}$, er de specificerede områder $\{? \leq b(1), b(1) < ? \leq b(2), \dots, b(n-1) < ? \leq b(n), b(n) > ?\}$. Den resulterende liste er et element længere end *binsListe*.

Hvert element af resultatet svarer til antallet af elementer fra *Liste1*, der er i dette område. Udtrykt med **countlf()**-funktionen er resultatet $\{\text{countlf}(\text{liste}, ? \leq b(1)), \text{countlf}(\text{liste}, b(1) < ? \leq b(2)), \dots, \text{countlf}(\text{liste}, b(n-1) < ? \leq b(n)), \text{countlf}(\text{liste}, b(n) > ?)\}$.

Elementer i *Liste1*, der ikke kan "placeres i en størrelse" ignoreres. Tomme (ugyldige) elementer ignoreres også. Yderligere oplysninger om tomme elementer findes på side 266.

I applikationen Lister og regneark kan du anvende et celleområde i stedet for begge argumenter.

Bemærk: Se også **countlf()**, side 37.

$\text{datalist} = \{1, 2, e, 3, \pi, 4, 5, 6, \text{"hello"}, 7\}$
$\{1, 2, 2.71828, 3, 3.14159, 4, 5, 6, \text{"hello"}, 7\}$
$\text{frequency}(\text{datalist}, \{2.5, 4.5\})$
$\{2, 4, 3\}$

Forklaring af resultatet:

2 elementer fra *Dataliste* er ≤ 2.5

4 elementer fra *Dataliste* er > 2.5 og ≤ 4.5

3 elementer fra *Dataliste* er > 4.5

Elementet "hello" er en streng og kan ikke placeres i nogen af de definerede områder.

FTest_2Samp

FTest_2Samp *Liste1, Liste2[, Hyppighed1[, Hyppighed2[, Hypot]]]*

FTest_2Samp *Liste1, Liste2[, Hyppighed1*

[,Hyppighed2[,Hypot]]]

(Datalisteinput)

FTest_2Samp *sx1,n1,sx2,n2[,Hypot]*

FTest_2Samp *sx1,n1,sx2,n2[,Hypot]*

(Sammenfatning, stat input)

Udfører en F test med to målinger. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 187.)

eller $H_a: \sigma_1 > \sigma_2$, sæt *Hypot*>0

Til $H_a: \sigma_1 \neq \sigma_2$ (standard), sæt *Hypot* =0

Til $H_a: \sigma_1 < \sigma_2$, sæt *Hypot*<0

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.F	Beregnet $\hat{U}$ statistik for datasekvensen
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.dfTæller	frihedsgrader for tæller = $n_1 - 1$
stat.dfNævner	tæller, frihedsgrader = $n_2 - 1$
stat.sx1, stat.sx2	Stikprøve standardafvigelse for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.x1_bar stat.x2_bar	Middelværdi af stikprøver for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Størrelse på stikprøverne

Func

Blok

EndFunc

Definition af en stykvis funktion:

Skabelon til oprettelse af en brugerdefineret funktion.

Blok kan være en enkelt sætning, en serie sætninger adskilt med kolon eller en serie sætninger på separate linjer. Funktionen kan anvende **Return**-instruktionen til at returnere et specifikt resultat.

Bemærk indtastning af

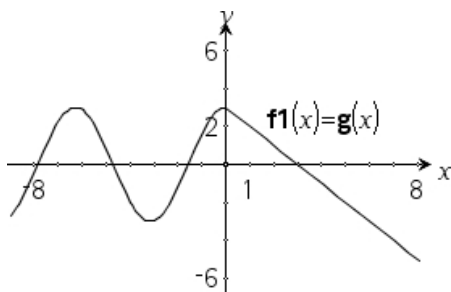
eksempel: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

Define $g(x)$ =Func

Done

```
If x<0 Then
Return 3*cos(x)
Else
Return 3-x
EndIf
EndFunc
```

Tegnet resultat af grafen $g(x)$



G

gcd()

gcd(*Værdi1*, *Værdi2*)⇒*udtryk*

gcd(18,33)

3

Returnerer den største fælles divisor af to argumenter. **Gcd** for to brøker er **gcd** af deres tællere divideret med **lcm** af deres nævnere.

I **Auto** eller **tilnærmet**-tilstand er **gcd** af flydende decimalbrøker 1.0.

gcd(*Liste1*, *Liste2*)⇒*liste*

gcd({12,14,16},{9,7,5})

{3,7,1}

Returnerer de største fælles divisorer af de tilsvarende elementer i *Liste1* og *Liste2*.

gcd(*Matrix1*, *Matrix2*)⇒*matrix*

gcd($\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}, \begin{bmatrix} 4 & 8 \\ 12 & 16 \end{bmatrix}$)

$\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$

Returnerer de største fælles divisorer af de tilsvarende elementer i *Matrix1* og *Matrix2*.

geomCdf

$(p, \text{nedreGrænse}, \text{øvreGrænse}) \Rightarrow \text{tal}$, hvis *nedreGrænse* og *øvreGrænse* er tal, *liste*, hvis *nedreGrænse* og *øvreGrænse* er lister

geomCdf($p, \text{øvreGrænse}$) for $P(1 \leq X \leq \text{øvreGrænse}) \Rightarrow \text{tal}$ hvis *øvreGrænse* er et tal, *liste*, hvis *øvreGrænse* er en liste

Beregner den kumulerede geometriske sandsynlighed fra *nedreGrænse* til *øvreGrænse* med den angivne sandsynlighed for succes *p*.

For $P(X \leq \text{øvreGrænse})$, sæt *nedreGrænse*=1.

geomPdf()

geomPdf($p, \text{XVærdi}$) $\Rightarrow \text{tal}$ hvis *XVærdi* er et tal, *liste* hvis *XVærdi* er en liste

Beregner sandsynligheden i *XVærdi*, nummeret på den forsøgsgang hvor den første succes forekommer, for den diskrete geometrisk distribution med den angivne sandsynlighed for succes, *p*.

Get**Hub-menu**

Get[*promptString*,] *var*[, *statusVar*]

Get[*promptStreng*,] *func*(*arg1*, ...*argn*)[, *statusVar*]

Programmeringskommando: Henter en værdi fra en tilsluttet TI-Innovator™ Hub og tildeler værdien til den variable *var*.

Der skal anmodes om værdien:

- i forvejen gennem en **Send "READ ..."** kommando.
— eller —
- ved at indlejre en **"READ ..."** anmodning som det valgfrie *promptString*-argument. Med denne

Eksempel: Anmod om den nuværende værdi for hubbens indbyggede lysniveausensor. Brug **Get** for at modtage værdien, og tildel den til den variable *lightval*.

Send "READ BRIGHTNESS"	Done
Get <i>lightval</i>	Done
<i>lightval</i>	0.347922

Indlejrer READ-anmodningen med **Get**-kommandoen.

Get "READ BRIGHTNESS", <i>lightval</i>	Done
<i>lightval</i>	0.378441

metode kan du bruge en enkelt kommando til at anmode om værdien og modtage den.

Implicit reduktion finder sted. For eksempel fortolkes en modtaget streng på "123" som en numerisk værdi. Brug **GetStr** i stedet for **Get** for at bevare strengen.

Hvis du inkluderer det valgfrie argument *statusVar*, tildeles det en værdi baseret på operationens succes. En værdi på nul betyder, at ingen data blev modtaget.

I den anden syntaks lader argumentet *func()* et program gemme den modtagne streng som en funktionsdefinition. Denne syntaks virker, ligesom hvis programmet udførte kommandoen:

Define *func(arg1, ...argn) = modtaget streng*

Så kan programmet bruge den definerede funktion *func()*.

Bemærk: Du kan bruge kommandoen **Get** i et brugerdefineret program, men ikke i en funktion.

Bemærk: Se også **GetStr**, side 90 og **Send**, side 167.

getDenom()

Katalog > 

getDenom(Udtr1) ⇒ udtryk

Transformerer argumentet til et udtryk med en forkortet fællesnævner og returnerer derefter dens nævner.

$\text{getDenom}\left(\frac{x+2}{y-3}\right)$	y^{-3}
$\text{getDenom}\left(\frac{2}{7}\right)$	7
$\text{getDenom}\left(\frac{1}{x} + \frac{y^2+y}{y^2}\right)$	$x \cdot y$

getKey()

Katalog > 

getKey([0|1]) ⇒ returnString

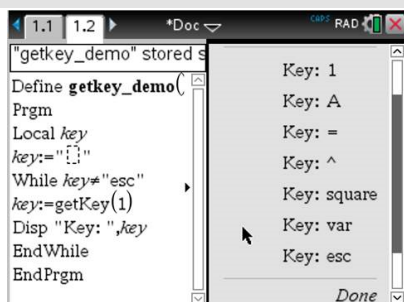
getKey()

Eksempel:

Beskrivelse: `getKey()` - tillader et TI-Basic-program at få tastaturinput - håndholdt, stationær og emuleret på stationær.

Eksempel:

- tasttrykket := `getKey()` vil returnere en tast eller en tom streng, hvis ingen tast blev trykket. Denne kommando vil straks returnere.
- tasttrykket := `getKey(1)` venter til en tast bliver trykt ned. Denne kommando vil sætte udførelsen af programmet på pause, indtil en tast er trykket ned.



Håndtering af tasttryk:

Håndholdte enheder/emulatortast	Skrivebord	Returværdi
Esc	Esc	"esc"
Touchpad - klik øverst	n/a	"op"
Til	n/a	"til top"
Scratchapps	n/a	"kladde"
Touchpad - venstreklik	n/a	"venstre"
Touchpad - klik i midten	n/a	"midten"
Touchpad - højreklik	n/a	"højre"
Dok	n/a	"dok"
Tabulator	Tabulator	"tabulator"
Touchpad - klik nederst	Ned pil	"ned"
menu	n/a	"menu"
Ctrl	Ctrl	ingen retur
Skift	Skift	ingen retur
Var	n/a	"var"
Del	n/a	"del"

Håndholdte enheder/emulatortast	Skrivebord	Returværdi
=	=	"="
trigonometri	n/a	"trigonometri"
0 til 9	0-9	"0" ... "9"
Skabeloner	n/a	"skabelon"
Katalog	n/a	"kat"
^	^	"^"
X^2	n/a	"kvadrat"
/ (divisionstast)	/	"/"
* (gangetast)	*	"*"
e^x	n/a	"eksp"
10^x	n/a	"10potens"
+	+	"+"
-	-	"_"
(	(	"("
)	)	")"
.	.	"."
(-)	n/a	"-" (negativ fortegn)
Indtast	Indtast	"indtast"
ee	n/a	"E" (eksponentiel notation E)
a - z	a - z	alpha = bogstav trykket ned (lille bogstav) ("a" - "z")
skift a - z	skift a - z	alpha = bogstav trykket ned "A" - "Z"
		Bemærk: ctrl-shift fungerer som lock-caps
?!	n/a	"?!"
pi	n/a	"pi"

Håndholdte enheder/emulatortast	Skrivebord	Returværdi
Flag	n/a	ingen retur
,	,	" "
Return	n/a	"retur"
mellemrum	mellemrum	" " (mellemrum)
Utilgængelig	Specielle tegntaster som @, !^, osv.	Tegnet returneres
n/a	Funktionstaster	Ingen returnerede tegn
n/a	Specielle skrivebordskontrolltaster	Ingen returnerede tegn
Utilgængelig	Andre skrivebordstaster, som ikke er tilgængelige på beregneren, når getkey () venter på et tastetryk. {, }, ,, :, ...)	Samme tegn du får i noter (ikke i et matematik-felt)

Bemærk: Det er vigtigt at være opmærksom på, at tilstedeværelsen af **getKey()** i et program ændrer, hvordan visse hændelser, bliver håndteret af systemet. Nogle af disse er beskrevet nedenfor.

Afslut program og håndter hændelse - Præcis som hvis en bruger ville afbryde et program ved at trykke på tasten **ON**

"**Support**" nedenfor betyder - system fungerer som forventet - program kører stadig.

Hændelse	Enhed	Skrivebord - TI-Nspire™ Student Software
Hurtig svar	Afslut program, håndter hændelse	Samme som den håndholdte (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-kun)
Fjern fil mgmt (Inkl. afsendelse af "exit tryk 2 test" fil fra en anden håndholdt eller skrivebordshåndholdt)	Afslut program, håndter hændelse	Samme som den håndholdte. (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-kun)
Afslut klasse	Afslut program, håndter hændelse	Support (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-kun)

Hændelse	Enhed	Skrivebords - TI-Nspire™ alle versioner
TI-Innovator™ Hub forbind/afbryd	Support - kan med held udstede kommandoer til TI-Innovator™ Hub. Når du går ud af programmet, arbejder TI-Innovator™ Hub stadig med den håndholdte.	Samme som den håndholdte

getLangInfo()

Katalog > 

getLangInfo()⇒*streng*

getLangInfo()

"en"

Returnerer en streng, som svarer til det korte navn af det aktuelle aktive sprog. Man kan, for eksempel, bruge det i et program eller funktion til at bestemme det aktuelle sprog.

Engelsk = "en"

Dansk = "da"

Tysk = "de"

Finsk = "fi"

Fransk = "fr"

Italiensk = "it"

Hollandsk = "nl"

Belgisk Hollandsk = "nl_BE"

Norsk = "no"

Portugisisk = "pt"

Spansk = "es"

Svensk = "sv"

getLockInfo()

Katalog > 

getLockInfo(Var)⇒*værdi*

a:=65

65

Returnerer den aktuelle låste/oplåste tilstand på variablen *Var*.

Lock a

Done

getLockInfo(a)

1

værdi =0: *Var* er ulåst eller findes ikke.

a:=75

"Error: Variable is locked."

DelVar a

"Error: Variable is locked."

værdi =1: *Var* er låst og kan ikke ændres eller slettes.

Unlock a

Done

a:=75

75

DelVar a

Done

Se **Lock**, side 112, og **unlock**, side 209.

getMode(TilstandNavnHeltal)⇒*værdi*

getMode(0)⇒*liste*

getMode(TilstandNavnHeltal)

returnerer en værdi, der repræsenterer den aktuelle indstilling for tilstanden *TilstandNavnHeltal*.

getMode(0) returnerer en liste, der indeholder talpar. Hvert par består af et tilstandsheltal og et indstillingsheltal.

Se tabellen nedenfor for en oversigt over tilstande og deres indstillinger.

Hvis du gemmer indstillingerne med **getMode(0)** → *var*, kan du anvende **setMode(var)** i en funktion eller et program for midlertidigt at gendanne indstillingerne under eksekveringen af funktionen eller programmet. Se **setMode()**, side 171.

getMode(0)	{ 1,7,2,1,3,1,4,1,5,1,6,1,7,1,8,1 }
getMode(1)	7
getMode(8)	1

Tilstands-navn	Tilstands-heltal	Indstillingsheltal
Viste cifre	1	1=Float, 2=Float1, 3=Float2, 4=Float3, 5=Float4, 6=Float5, 7=Float6, 8=Float7, 9=Float8, 10=Float9, 11=Float10, 12=Float11, 13=Float12, 14=Fix0, 15=Fix1, 16=Fix2, 17=Fix3, 18=Fix4, 19=Fix5, 20=Fix6, 21=Fix7, 22=Fix8, 23=Fix9, 24=Fix10, 25=Fix11, 26=Fix12
Vinkel	2	1=Radian, 2=Grader, 3=Gradian
Eksponentielt format	3	1=Normal, 2=Videnskabelig, 3=Teknisk
Reel eller kompleks	4	1=Reel, 2=Rektangulær, 3=Polær
Auto eller tilnærmet	5	1=Auto, 2=Tilnærmet, 3=Eksakt
Vektorformat	6	1=Rektangulær, 2=Cylindrisk, 3=Sfærisk
Talsystem	7	1=Decimal, 2=Hex, 3=Binær
Enhedssystem	8	1=SI, 2=Eng/US

getNum()	Katalog > 	
getNum (<i>Udtr1</i>) $\Rightarrow$ <i>udtryk</i>		
Transformerer argumentet til et udtryk med en forkortet fællesnævner og returnerer derefter dens tæller.		
	$\text{getNum}\left(\frac{x+2}{y-3}\right)$	$x+2$
	$\text{getNum}\left(\frac{2}{7}\right)$	2
	$\text{getNum}\left(\frac{1}{x}+\frac{1}{y}\right)$	$x+y$

GetStr	Hub-menu
GetStr <i>[promptStreng,] var[, statusVar]</i>	Se Get for eksempler.
GetStr <i>[promptStreng,] func(arg1, ...argn) [, statusVar]</i>	
Programmeringskommando: Fungerer identisk med kommandoen Get, bortset fra at den hentede værdi altid fortolkes som en streng. I modsætning hertil fortolker kommandoen Get svarene som et udtryk, medmindre det er omsluttet af citationstegn ("").	
Bemærk: Se også Get, side 83 og Send, side 167.	

getType()	Katalog > 	
getType (<i>var</i>) $\Rightarrow$ <i>streng</i>		
Returnerer en streng, som angiver datatypen for variablen <i>var</i> .		
Hvis <i>var</i> ikke er defineret, returneres strengen "NONE".		
	$\{1,2,3\} \rightarrow temp$	$\{1,2,3\}$
	$\text{getType}(temp)$	"LIST"
	$3 \cdot i \rightarrow temp$	$3 \cdot i$
	$\text{getType}(temp)$	"EXPR"
	$\text{DelVar } temp$	<i>Done</i>
	$\text{getType}(temp)$	"NONE"

getVarInfo() $\Rightarrow$ matrix eller streng

getVarInfo(BibNavnStreng) $\Rightarrow$ matrix eller streng

getVarInfo() returnerer en matrix med informationer (variabelnavn, type, bibliotekets tilgængelighed og låst/ulåst-status) for alle variable og biblioteksobjekter defineret i den aktuelle opgave.

Hvis der ikke er defineret nogen variable, returnerer **getVarInfo()** strengen "NONE"

getVarInfo

(*Biblioteksnavnstreng*) returnerer en matrix med oplysninger for alle biblioteksobjekter, der er defineret i biblioteket *BibNavnStreng*.

BibNavnStreng skal være en streng (tekst omsluttet af citationstegn) eller en strengvariabel.

Hvis biblioteket *BibNavnStreng* ikke findes, opstår der en fejl.

Bemærk eksemplet til venstre, i hvilket resultatet af **getVarInfo()** er tilknyttet til variabel *vs*. Forsøg på at vise række 2 eller række 3 af *vs* returnerer en "ugyldig liste eller matrix" fejl, fordi mindst et af elementerne i disse rækker (variable *b*, f.eks) reevaluerer til en matrix.

Denne fejl kan også opstå, når *Ans* bruges til at evaluere et **getVarInfo()** resultat.

Systemet giver den ovenfor nævnte fejl, fordi den aktuelle version af softwaren ikke understøtter en generaliseret matrixstruktur, hvor et element i en matrix enten kan være en matrix eller en liste.

getVarInfo()	"NONE"												
Define x=5	Done												
Lock x	Done												
Define LibPriv y={1,2,3}	Done												
Define LibPub z(x)=3·x ² -x	Done												
getVarInfo()	<table><tr><td>x</td><td>"NUM"</td><td>"{"}</td><td>1</td></tr><tr><td>y</td><td>"LIST"</td><td>"LibPriv"</td><td>0</td></tr><tr><td>z</td><td>"FUNC"</td><td>"LibPub"</td><td>0</td></tr></table>	x	"NUM"	"{"}	1	y	"LIST"	"LibPriv"	0	z	"FUNC"	"LibPub"	0
x	"NUM"	"{"}	1										
y	"LIST"	"LibPriv"	0										
z	"FUNC"	"LibPub"	0										
getVarInfo(tmp3)	"Error: Argument must be a string"												
getVarInfo("tmp3")	<table><tr><td>volcyl2</td><td>"NONE"</td><td>"LibPub"</td><td>0</td></tr></table>	volcyl2	"NONE"	"LibPub"	0								
volcyl2	"NONE"	"LibPub"	0										

$a:=1$	1			
$b:=[1\ 2]$	[1 2]			
$c:=[1\ 3\ 7]$	[1 3 7]			
$vs:=\text{getVarInfo}()$	$\begin{bmatrix} a \\ b \\ c \end{bmatrix}$	$\begin{bmatrix} \text{"NUM"} \\ \text{"MAT"} \\ \text{"MAT"} \end{bmatrix}$	$\begin{bmatrix} \text{"{"}} \\ \text{"{"}} \\ \text{"{"}} \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
$vs[1]$	[1	"NUM"	"{"	0]
$vs[1,1]$	1			
$vs[2]$	"Error: Invalid list or matrix"			
$vs[2,1]$	[1 2]			

Goto		Katalog > 
Goto <i>etiketnavn</i>	Define $g()$ =Func	Done
Overfører kontrol til etiketten <i>etiketnavn</i> .	Local $temp,i$	
	$0 \rightarrow temp$	
	$1 \rightarrow i$	
<i>Etiketnavn</i> skal defineres i den samme funktion med en Lbl -kommando.	Lbl <i>top</i>	
	$temp+i \rightarrow temp$	
	If $i < 10$ Then	
	$i+1 \rightarrow i$	
Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.	Goto <i>top</i>	
	EndIf	
	Return $temp$	
	EndFunc	
	$g()$	55

►Grad		Katalog > 
<i>Udtr1</i> ► Grad ⇒ <i>udtryk</i>	I vinkeltilstanden Grader:	
Konverterer <i>Udtr1</i> til vinkelmål i nygrader.	$(1.5) \blacktriangleright \text{Grad}$	$(1.66667)^g$
Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @> Grad .	I vinkeltilstanden Radian:	
	$(1.5) \blacktriangleright \text{Grad}$	$(95.493)^g$

/

identitet()		Katalog > 
identitet (<i>heltal</i>) ⇒ <i>matrix</i>	identity(4)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Returnerer identitetsmatrixen med en dimension af <i>heltal</i> .		
<i>Heltal</i> skal være et positivt heltal.		

If		Katalog > 
If <i>BooleanExpr</i> <i>Statement</i>	Define $g(x)$ =Func	Done
	If $x < 0$ Then	
	Return x^2	
If <i>BooleanExpr</i> Then <i>Block</i>	EndIf	
EndIf	EndFunc	
	$g(-2)$	4

Hvis *BooleanExpr* evalueres som sand, eksekveres enkelt sætningen *Statement* eller sætningsblokken *Block*, før eksekveringen fortsættes.

Hvis *BooleanExpr* evalueres som falsk, fortsættes eksekveringen uden eksekvering af sætningen eller sætningsblokken.

Block kan enten være en enkelt sætning eller en række sætninger adskilt af ":-"-tegn

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

If *BooleanExpr* **Then**
 Block1

Else
 Block2

EndIf

Hvis *BooleanExpr* evalueres som sand, eksekveres *Block1*, og *Block2* springes over.

Hvis *BooleanExpr* evalueres som falsk, springes over *Block1*, men *Block2* eksekveres.

Block1 og *Block2* kan være en enkelt sætning.

Define $g(x) =$	Func	Done
	If $x < 0$ Then	
	Return $-x$	
	Else	
	Return x	
	EndIf	
	EndFunc	
$g(12)$		12
$g(-12)$		12

If BooleanExpr1 Then*Block1***ElseIf BooleanExpr2 Then***Block2*

:

ElseIf BooleanExprN Then*BlockN***EndIf**

Muliggør en forgrening. Hvis *BooleanExpr1* evalueres som sand, eksekveres *Block1*. Hvis *BooleanExpr1* evalueres som falsk, evalueres *BooleanExpr2*, og så videre.

Define $g(x) = \text{Func}$ If $x < 5$ Then

Return 5

ElseIf $x > 5$ and $x < 0$ ThenReturn $\neg x$ ElseIf $x \geq 0$ and $x \neq 10$ ThenReturn x ElseIf $x = 10$ Then

Return 3

EndIf

EndFunc

Done

$g(-4)$	4
$g(10)$	3

ifFn()Katalog >

**ifFn(BooleanExpr, Value If true
[, Value If false [, Value If unknown]])**
 $\Rightarrow$ udtryk, liste eller matrix

Beregner det boolske udtryk *BooleanExpr* (eller hvert element i *BooleanExpr*) og giver et resultat baseret på følgende regler:

- *BooleanExpr* kan teste en enkelt værdi, en liste eller en matrix.
- Hvis et element i *BooleanExpr* evaluerer sandt, returneres det tilsvarende element fra *Value If true*.
- Hvis et element i *BooleanExpr* evaluerer falsk, returneres det tilsvarende element fra *Value If false*. Hvis du udelader *Value If false*, returneres undef.
- Hvis et element i *BooleanExpr* hverken er sandt eller falsk, returneres det tilsvarende element *Value If unknown*. Hvis du udelader *Value If unknown*, returneres undef.
- Hvis det andet, tredje eller fjerde argument i **ifFn()**-funktionen er et enkelt udtryk, udføres den Boolske test på hver position i *BooleanExpr*.

ifFn($\{1,2,3\} < 2.5, \{5,6,7\}, \{8,9,10\}$)
 $\{5,6,10\}$

Testværdien for **1** er mindre end 2,5, så dens tilsvarende

Value If True-element på **5** kopieres til resultatlisten.

Testværdien for **2** er mindre end 2,5, så dens tilsvarende

Value If True-element på **6** kopieres til resultatlisten.

Testværdien for **3** er ikke mindre end 2,5, så det tilhørende *Value If False*-element på **10** kopieres til resultatlisten.

ifFn($\{1,2,3\} < 2.5, 4, \{8,9,10\}$)
 $\{4,4,10\}$

Value If True er en enkelt værdi og svarer til enhver valgt position.

ifFn($\{1,2,3\} < 2.5, \{5,6,7\}$)
 $\{5,6,\text{undef}\}$

ifFn()Katalog > 

Bemærk: Hvis den reducerede *BooleanExpr*-sætning indeholder en liste eller matrix, skal alle andre liste- eller matrixargumenter have de samme dimensioner, og resultatet vil have de samme dimensioner.

Value_If_False er ikke specificeret. Undef anvendes.

$$\text{ifFn}(\{2, "a" \} < 2.5, \{6, 7\}, \{9, 10\}, "err") \\ \{6, "err" \}$$

Ét element valgt fra *Value_If_True*. Ét element valgt fra *Value_If_unknown*.

imag()Katalog > 

imag(*ExprI*) $\Rightarrow$ *udtryk*

Returnerer imaginærdelen af argumentet.

$\text{imag}(1+2 \cdot i)$	2
$\text{imag}(z)$	0
$\text{imag}(x+i \cdot y)$	y

Bemærk: Alle udefinerede variable behandles som reelle variable. Se også reel(), page 154

imag(*ListI*) $\Rightarrow$ *liste*

Returnerer en liste med imaginærdelen af elementerne.

$$\text{imag}(\{-3, 4-i, i\}) \quad \{0, -1, 1\}$$

imag(*MatrixI*) $\Rightarrow$ *matrix*

Returnerer en matrix med imaginærdelene af elementerne.

$$\text{imag}\left(\begin{bmatrix} a & b \\ i \cdot c & i \cdot d \end{bmatrix}\right) \quad \begin{bmatrix} 0 & 0 \\ c & d \end{bmatrix}$$

impDif()Katalog > 

impDif(*Equation*, *Var*, *dependVar* [, *Ord*]) $\Rightarrow$ *udtryk*

hvor ordenen *Ord* som standard er 1.

$$\text{impDif}(x^2+y^2=100, x, y) \quad \frac{-x}{y}$$

Beregner den implicitte differentialkvotient til ligninger, hvor en variabel defineres implicit ud fra en anden.

Henvisning

Se #(), side 240.

inString()Katalog > 

inString(*srcString*, *subString*[, *Start*]) ⇒ *heltal*

<code>inString("Hello there","the")</code>	7
<code>inString("ABCEFG","D")</code>	0

Returnerer tegnpositionen i strengen *srcString*, hvor første forekomst af strengen *subString* begynder.

Start, hvis medtaget, angiver den position i *srcString*, hvor søgningen begynder. Standard = 1 (første tegn i *srcString*).

Hvis *srcString* ikke indeholder *subString*, eller *Start* er > længden på *srcString*, returneres nul.

int()Katalog > 

int(*Expr*) ⇒ *heltal*

int(*List1*) ⇒ *liste*

int(*Matrix1*) ⇒ *matrix*

<code>int(-2.5)</code>	-3.
<code>int([-1.234 0 0.37])</code>	<code>[-2. 0 0.]</code>

Returnerer det største heltal, der er mindre end eller lig med argumentet. Denne funktion identisk med **floor()**.

Argumentet kan være et reelt eller komplekst tal.

For lister og matrixer returneres det største heltal mindre end eller lig med hvert element.

intDiv()Katalog > 

intDiv(*Number1*, *Number2*) ⇒ *heltal*

intDiv(*List1*, *List2*) ⇒ *liste*

intDiv(*Matrix1*, *Matrix2*) ⇒ *matrix*

<code>intDiv(-7,2)</code>	-3
<code>intDiv(4,5)</code>	0
<code>intDiv({12,-14,-16},{5,4,-3})</code>	<code>{2,-3,5}</code>

Returnerer den heltalsdel med fortegn, der er en del af (*Number1* ÷ *Number2*).

Returnerer for lister og matrixer heltalsdelen med fortegn, der er en del af (argument 1 ÷ argument 2), for hvert elementpar.

interpoler ()

Katalog >

interpoler(*xValue*, *xList*, *yList*, *yPrimeList*) $\Rightarrow$ *liste*

Denne funktion gør følgende:

Givet *xList*, *yList*=**f**(*xList*) og *yPrimeList*=**f'**(*xList*) for en ukendt funktion **f** anvendes en kubisk interpolation til at approksimere funktionen **f** ved *xValue*. Det antages, at *xList* er en liste med monotont voksende eller aftagende tal, men denne funktion kan returnere en værdi selvom det ikke er tilfældet. Denne funktion gennemløber *xList* i søgningen efter et interval [*xList*[*i*], *xList*[*i*+1]] der indeholder *xValue*. Hvis den finder et sådan interval, returnerer den en interpoleret værdi for **f**(*xValue*); i modsat fald returnerer den **undef**.

xList, *yList* og *yPrimeList* skal have samme dimension ≥ 2 og indeholde udtryk, der reducerer til tal.

xValue kan være en ikke-defineret variabel, et tal eller en liste af tal.

Differentialligning:

$$y' = -3 \cdot y + 6 \cdot t + 5 \text{ og } y(0) = 5$$

$rk := rk23(-3 \cdot y + 6 \cdot t + 5, y, \{0, 10\}, 5, 1)$					
0.	1.	2.	3.	4.	
5.	3.19499	5.00394	6.99957	9.00593	10.

Du kan se hele resultatet ved at trykke på $\blacktriangle$ og derefter bruge $\blacktriangleleft$ og $\blacktriangleright$ til at bevæge markøren.

Brug funktionen **interpolate()** til at beregne funktionsværdierne for listen med *x*-værdier:

$xvalueList := seq(i, 0, 10, 0.5)$	
{ 0, 0.5, 1., 1.5, 2., 2.5, 3., 3.5, 4., 4.5, 5., 5.5, 6., 6.5, 7., 7.5, 8., 8.5, 9., 9.5, 10. }	
$xlist := mat \blacktriangleright list(rk[1])$	
{ 0., 1., 2., 3., 4., 5., 6., 7., 8., 9., 10. }	
$ylist := mat \blacktriangleright list(rk[2])$	
{ 5., 3.19499, 5.00394, 6.99957, 9.00593, 10.9979 }	
$yprimeList := -3 \cdot y + 6 \cdot t + 5 y = ylist \text{ og } t = xlist$	
{ -10., 1.41503, 1.98819, 2.00129, 1.98221, 2.0067 }	
$interpolate(xvalueList, xlist, ylist, yprimeList)$	
{ 5., 2.67062, 3.19499, 4.02782, 5.00394, 6.0001 }	

inv χ^2 ()

Katalog >

inv χ^2 (*Areal*, *df*)

invChi2(*Areal*, *df*)

Beregner den inverse kumulerede χ^2 (chi-kvadrat) sandsynlighedsfunktion angivet ved frihedsgrad, *fg* for et givet *Areal* under kurven.

invF()

Katalog >

invF(*Areal*, *dfNumer*, *dfDenom*)

invF(Area,dfNumer,dfDenom)

Beregner den inverse kumulerede F fordelingsfunktion angivet ved *dfNumer* og *dfDenom* for et givet *Areal* under kurven.

invBinom()

invBinom
(*CumulativeProb*,*NumTrials*,*Prob*,
OutputForm) $\Rightarrow$ skalar eller matrix

Givet antallet af forsøg (*NumTrials*) og sandsynligheden for succes i hvert forsøg (*Prob*) vil denne funktion returnere det minimale antal succeser, *k*, således at den kumulerede sandsynlighed for *k* succeser er større end eller lig med den givne kumulerede sandsynlighed (*CumulativeProb*).

OutputForm=0, viser resultatet som en skalar (standard).

OutputForm=1, viser resultatet som en matrix.

Eksempel: Mary og Kevin spiller et spil terninger. Mary skal gætte det maksimale antal gange, der rulles en sekser i 30 slag. Hvis der rulles en sekser så mange eller færre gange, vinder Mary. Desuden vinder hun mere, jo mindre det tal, hun gætter på, er. Hvad er det mindste tal, Mary kan gætte på, hvis hun vil have en sandsynlighed for at vinde på over 77 %?

$\text{invBinom}\left(0.77, 30, \frac{1}{6}\right)$	6
$\text{invBinom}\left(0.77, 30, \frac{1}{6}, 1\right)$	$\begin{bmatrix} 5 & 0.616447 \\ 6 & 0.776537 \end{bmatrix}$

invBinomN()

invBinomN(*CumulativeProb*,*Prob*,
NumSuccess,*OutputForm*) $\Rightarrow$ skalar
eller matrix

Givet sandsynligheden for succes i hvert forsøg (*Prob*) og antallet af succeser (*NumSuccess*) vil denne funktion returnere det minimale antal forsøg, *N*, således at den kumulerede sandsynlighed for *x* succeser er mindre end eller lig med den givne kumulerede sandsynlighed (*CumulativeProb*).

OutputForm=0, viser resultatet som en skalar (standard).

OutputForm=1, viser resultatet som en matrix.

Eksempel: Monique øver sig i målskud i kurvebold. Hun ved fra erfaring, at hendes chance for at ramme målet ved et kast er 70 %. Hun har tænkt sig at øve, til hun har fået 50 mål. Hvor mange skud skal hun forsøge for at være sikker på, at sandsynligheden for at få mindst 50 mål er større end 0,99?

$\text{invBinomN}(0.01, 0.7, 49)$	86
$\text{invBinomN}(0.01, 0.7, 49, 1)$	$\begin{bmatrix} 85 & 0.010451 \\ 86 & 0.00709 \end{bmatrix}$

invNorm()Katalog > **invNorm**(*Areal*[, μ],[σ])

Beregner den inverse kumulerede normalfordelingsfunktion for et givet *Areal* under normalfordelingskurven angivet ved μ og σ .

invT()Katalog > **invT**(*Areal*,*df*)

Beregner den inverse kumulerede student-t sandsynlighedsfunktion angivet ved frihedsgrad *fg* for et givet *Area* under kurven.

iPart()Katalog > **iPart**(*Number*) $\Rightarrow$ *heltal***iPart**(*List1*) $\Rightarrow$ *liste***iPart**(*Matrix1*) $\Rightarrow$ *matrix*

iPart (-1.234)	-1.
iPart ($\left\{\frac{3}{2}, -2.3, 7.003\right\}$)	$\{1, 2., 7.\}$

Returnerer heltalsdelen af argumentet.

Returnerer heltalsdelen af hvert element for lister og matrixer.

Argumentet kan være et reelt eller komplekst tal.

irr()Katalog > **irr**(*CF0*,*CFList* [*,CFFreq*]) $\Rightarrow$ *værdi*

Finansfunktion, der beregner den interne rente af en investering.

CF0 er startpengestrømmen på tidspunkt 0; den skal være et reelt tal.

CFList er en liste over pengestrømsbeløb efter startpengestrømmen *CF0*.

<i>list1</i> := { 6000, -8000, 2000, -3000 }	{ 6000, -8000, 2000, -3000 }
<i>list2</i> := { 2, 2, 2, 1 }	{ 2, 2, 2, 1 }
irr (5000, <i>list1</i> , <i>list2</i>)	-4.64484

CFFreq er en valgfri liste, hvor hvert element angiver hyppigheden for et grupperet (fortløbende) pengestrømsbeløb, som er det tilsvarende element i *CFList*. Standardværdien er 1; Hvis du indtaster værdier, skal de være positive heltal < 10.000.

Bemærk: Se også *mirr()*, side 122.

isPrime()

isPrime(*Number***)** $\Rightarrow$ *boolsk konstantudtryk*

Returnerer sand eller falsk for at vise, om *Number* er et helt tal ≥ 2 , der kun kan divideres med sig selv og 1.

Hvis *Number* har flere end ca. 306 cifre og ikke har nogen faktorer ≤ 1021 , viser **isPrime(***Number***)** en fejlmeddelelse.

Hvis du kun vil bestemme, om *Number* er et primtal, skal du anvende **isPrime()** i stedet for **factor()**. Det er meget hurtigere, især hvis *Number* ikke er et primtal, og den næststørste faktor har mere end fem cifre.

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

isPrime(5)	true
isPrime(6)	false

Funktion til søgning af det næste primtal efter det angivne tal:

Define <i>nextprim</i> (<i>n</i>)=Func	Done
Loop	
$n+1 \rightarrow n$	
If isPrime(<i>n</i>)	
Return <i>n</i>	
EndLoop	
EndFunc	
<i>nextprim</i> (7)	11

isVoid()

isVoid(*Var***)** $\Rightarrow$ *boolsk konstantudtryk*
isVoid(*Expr***)** $\Rightarrow$ *boolsk konstantudtryk*
isVoid(*List***)** $\Rightarrow$ *liste over boolske konstantudtryk*

Returnerer sand eller falsk for at angive, om argumentet er en ugyldig datatype.

For yderligere oplysninger om ugyldige elementer se side 266.

<i>a</i> :=_	_
isVoid(<i>a</i>)	true
isVoid({ 1,_,3 })	{ false,true,false }

Lbl	Katalog > 																								
Lbl <i>etiketNavn</i> Definerer en etiket med navnet <i>etiketNavn</i> i en funktion. Du kan anvende en Goto <i>etiketNavn</i>-kommando til at videregive kontrollen til kommandoen lige efter etiketten. <i>EtiketNavn</i> skal opfylde de samme navngivningskrav som et variabelnavn. Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.	<table border="0"> <tr> <td>Define <i>g()</i>=Func</td><td>Done</td></tr> <tr> <td>Local <i>temp,i</i></td><td></td></tr> <tr> <td>$0 \rightarrow temp$</td><td></td></tr> <tr> <td>$1 \rightarrow i$</td><td></td></tr> <tr> <td>Lbl <i>top</i></td><td></td></tr> <tr> <td>$temp+i \rightarrow temp$</td><td></td></tr> <tr> <td>If $i < 10$ Then</td><td></td></tr> <tr> <td>$i+1 \rightarrow i$</td><td></td></tr> <tr> <td>Goto <i>top</i></td><td></td></tr> <tr> <td>EndIf</td><td></td></tr> <tr> <td>Return <i>temp</i></td><td></td></tr> <tr> <td>EndFunc</td><td></td></tr> </table> <i>g()</i> 55	Define <i>g()</i> =Func	Done	Local <i>temp,i</i>		$0 \rightarrow temp$		$1 \rightarrow i$		Lbl <i>top</i>		$temp+i \rightarrow temp$		If $i < 10$ Then		$i+1 \rightarrow i$		Goto <i>top</i>		EndIf		Return <i>temp</i>		EndFunc	
Define <i>g()</i> =Func	Done																								
Local <i>temp,i</i>																									
$0 \rightarrow temp$																									
$1 \rightarrow i$																									
Lbl <i>top</i>																									
$temp+i \rightarrow temp$																									
If $i < 10$ Then																									
$i+1 \rightarrow i$																									
Goto <i>top</i>																									
EndIf																									
Return <i>temp</i>																									
EndFunc																									

lcm()	Katalog > 				
lcm(<i>Værdi1</i>, <i>Værdi2</i>)⇒<i>udtryk</i> lcm(<i>Liste1</i>, <i>Liste2</i>)⇒<i>liste</i> lcm(<i>Matrix1</i>, <i>Matrix2</i>)⇒<i>matrix</i> Returnerer det mindste fælles multiplum af to argumenter. lcm af to brøker er lcm af deres tællere divideret med gcd af deres nævnere. lcm af brøker med flydende komma er deres produkt. For to lister eller matricer returneres det mindste fælles multiplum af deres tilsvarende elementer.	<table border="0"> <tr> <td>lcm(6,9)</td><td>18</td></tr> <tr> <td>lcm($\left\{\frac{1}{3}, -14, 16\right\}, \left\{\frac{2}{15}, 7, 5\right\}$)</td><td>$\left\{\frac{2}{3}, 14, 80\right\}$</td></tr> </table>	lcm (6,9)	18	lcm ($\left\{\frac{1}{3}, -14, 16\right\}, \left\{\frac{2}{15}, 7, 5\right\}$)	$\left\{\frac{2}{3}, 14, 80\right\}$
lcm (6,9)	18				
lcm ($\left\{\frac{1}{3}, -14, 16\right\}, \left\{\frac{2}{15}, 7, 5\right\}$)	$\left\{\frac{2}{3}, 14, 80\right\}$				

left()	Katalog > 		
left(<i>kildeStreng</i>[, <i>Tal</i>])⇒<i>streng</i> Returnerer <i>Antal</i>-tegn fra venstre i tegnstrengen <i>kildeStreng</i>. Hvis du udelader <i>Antal</i>, returneres alle <i>kildeStreng</i>-variable.	<table border="0"> <tr> <td>left("Hello",2)</td><td>"He"</td></tr> </table>	left ("Hello",2)	"He"
left ("Hello",2)	"He"		

left(*Liste1*[, *Antal*])⇒*liste* $\text{left}(\{1, 3, -2, 4\}, 3)$ $\{1, 3, -2\}$

Returnerer *Antal*-elementer til venstre i *Liste1*.

Hvis du udelader *Antal*, returneres hele *Liste1*.

left(*Sammenligning*)⇒*udtryk* $\text{left}(x < 3)$ x

Returnerer venstre af en ligning eller ulighed.

libShortcut()**libShortcut**(*BibNavneStreng*,
GenvejNavneStreng[, *BibPrivFlag*])⇒ *liste med variable*

Opretter en variabelgruppe i den aktuelle opgave, som indeholder referencer til alle objekter i det specificerede biblioteksdokument *bibNavneStreng*. Tilføjer også gruppemedlemmerne til variabelmenuen. Du kan henvise til hvert objekt ved brug af *GenvejNavneStreng*.

Sæt *BibPrivFlag* = **0** for at udelukke private biblioteksobjekter (standard)

Sæt *BibPrivFlag* = **1** for at medtage private biblioteksobjekter

For at kopiere en variabelgruppe, se **CopyVar** på side 31.

For at slette en variabelgruppe, se **DelVar** på side 51.

Dette eksempel forudsætter et korrekt gemt og opdateret biblioteksdokument med navnet, **linalg2**, som indeholder objekter defineret som *clearmat*, *gauss1*, og *gauss2*.

```
getVarInfo("linalg2")
[ clearmat  "FUNC"  "LibPub "
  gauss1    "PRGM"  "LibPriv "
  gauss2    "FUNC"  "LibPub " ]

libShortcut("linalg2", "la")
{ la.clearmat, la.gauss2 }

libShortcut("linalg2", "la", 1)
{ la.clearmat, la.gauss1, la.gauss2 }
```

limit(*Udtr1*, *Var*, *Punkt* [, *Retning*]) $\Rightarrow$ *udtryk*

$$\lim_{x \rightarrow 5} (2 \cdot x + 3) = 13$$

limit(*Liste1*, *Var*, *Punkt* [, *Retning*]) $\Rightarrow$ *liste*

$$\lim_{x \rightarrow 0^+} \left(\frac{1}{x} \right) = \infty$$

limit(*Matrix1*, *Var*, *Punkt* [, *Retning*]) $\Rightarrow$ *matrix*

$$\lim_{x \rightarrow 0} \left(\frac{\sin(x)}{x} \right) = 1$$

Returnerer den ønskede grænseværdi.

$$\lim_{h \rightarrow 0} \left(\frac{\sin(x+h) - \sin(x)}{h} \right) = \cos(x)$$

Bemærk: Se også **Grænseværdi skabelon**, side 7.

$$\lim_{n \rightarrow \infty} \left(\left(1 + \frac{1}{n} \right)^n \right) = e$$

Retning: negativ=fra venstre, positiv=fra højre, ellers =begge. (Ved udeladelse er *Retning* som standard begge.)

Grænseværdier i $+\infty$ og $-\infty$ konverteres altid til ensidige grænseværdier fra den begrænsede side.

Afhængigt af omstændighederne returnerer **limit()** sig selv eller undef hvis der ikke kan bestemmes en entydig grænseværdi. Dette betyder ikke nødvendigvis, at der ikke findes en entydig grænseværdi. Undef betyder, at resultatet enten er et ukendt tal af endelig eller uendelig størrelse eller en hel mængde af sådanne tal.

limit() benytter metoder som L'Hopital's regel, så der findes entydige grænseværdier, der ikke kan bestemmes. Hvis *Udtr1* indeholder andre udefinerede variable end *var*, kan det være nødvendigt at begrænse dem for at opnå et mere nøjagtigt resultat.

$$\lim_{x \rightarrow \infty} (a^x) = \text{undef}$$

$$\lim_{x \rightarrow \infty} (a^x) | a > 1 = \infty$$

$$\lim_{x \rightarrow \infty} (a^x) | a > 0 \text{ and } a < 1 = 0$$

Grænseværdier kan være meget følsomme over for afrundingsfejl. Undgå indstillingen Tilnærmet i

Auto eller tilnærmet-tilstanden og tilnærmede tal ved beregning af grænseværdier, hvor det er muligt Ellers vil grænseværdier, der skulle være nul eller uendelige og grænseværdier, der skulle have været endelige ikke-nul, sandsynligvis ikke være det.

LinRegBx $X, Y, [Frekv][Kategori, Medtag]$

Beregner den lineære regression $y = a + b \cdot x$ på listerne X og Y med hyppigheder $Frekv$. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Alle lister skal have ens dimensioner med undtagelse af undtagelse af *Medtag*.

X og Y er lister med uafhængige og afhængige variable.

Frekv er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende X og Y datapunkt. Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

Kategori er en liste, der indeholder kategorikoder for X og Y data.

Medtag er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a + b \cdot x$
stat.a, stat.b	Regressionskoefficienter
stat.r ²	Forklaringsgraden
stat.r	Korrelationskoefficient
stat.Resid	Residualer fra regressionen
stat.XReg	Liste af datapunkter i den modificerede <i>X</i> -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y</i> -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

LinRegMx *X*, *Y*, [*Frekv*][*Kategori*, *Medtag*]

Beregner den lineære regression $y = m \cdot x + b$ på listerne *X* og *Y* med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Alle lister skal have ens dimensioner med undtagelse af undtagelse af *Medtag*.

X og *Y* er lister med uafhængige og afhængige variable.

Frekv er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende *X* og *Y* datapunkt. Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

Kategori er en liste, der indeholder kategorikoder for *X* og *Y* data..

Medtag er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $m \cdot x + b$
stat.m, stat.b	Regressionskoefficienter
stat.r ²	Forklaringsgraden
stat.r	Korrelationskoefficient
stat.Resid	Residualer fra regressionen
stat.XReg	Liste af datapunkter i den modificerede <i>X</i> -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y</i> -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

LinRegtIntervals $X, Y[, F[, 0[, CNiveau]]]$

Til hældning. Beregner et niveau C konfidensinterval for hældningen.

LinRegtIntervals $X, Y[, F[, 1, Xværdi[, CNiveau]]]$

Åbent svar Beregner en forudset y-værdi, et niveau C forudsigelsesinterval for enkle observationer, og et niveau C konfidensinterval til gennemsnits-responsen.

En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187)

Alle lister skal have samme dimension.

X og Y er lister med uafhængige og afhængige variable.

F er en valgfri liste med hyppighedsværdier. Hvert element i F angiver hyppigheden af forekomster for hvert tilsvarende datapunkt for X og Y . Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variable	Beskrivelse
stat.RegEqn	Regressionsligning: $a+b \cdot x$
stat.a, stat.b	Regressionskoefficienter
stat.fg	Frihedsgrader
stat.r ²	Forklaringsgraden
stat.r	Korrelationskoefficient
stat.Resid	Residualer fra regressionen

Kun for hældningstypen

Output-variable	Beskrivelse
[stat.CLower, stat.CUpper]	Konfidensinterval for hældningen
stat.ME	Konfidensinterval, fejlmargen

Output-variabel	Beskrivelse
stat.SESlope	Standardfejl for hældning
stat.s	Standardfejl for linjen

Kun for svartype

Output-variabel	Beskrivelse
[stat.CLower, stat.CUpper]	Konfidensinterval for en middelværdi
stat.ME	Konfidensinterval, fejlmargen
stat.SE	Standardfejl for middelværdi
[stat.LowerPred, stat.UpperPred]	Prædiktionsinterval for en enkelt observation
stat.MEPred	Prædiktionsintervallsmargin for fejl
stat.SEPred	standardfejl for prædiktion
Statistik. $\hat{y}$	$a + b \cdot X$ Værdi

LinRegtTest

[katalog](#) > 

LinRegtTest $X, Y[, Frekv[, Hypot]]$

Beregner en lineær regression ud fra X og Y listerne og en t test på værdien af hældningen β og korrelationskoefficienten ρ for ligningen $y = \alpha + \beta x$. Den tester nulhypotesen $H_0: \beta = 0$ (ækvivalent, $\rho = 0$) mod en af tre alternative hypoteser.

Alle lister skal have samme dimension.

X og Y er lister med uafhængige og afhængige variable.

$Frekv$ er en valgfri liste med hyppigheder. Hvert element i $Frekv$ angiver hyppigheden af hændelse for hver tilsvarende X og Y datapunkt. Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

$Hypot$ er en valgfri værdi, som angiver en af tre alternative hypoteser, mod hvilken nulhypotesen ($H_0: \beta = \rho = 0$) vil blive testet.

Til $H_a: \beta \neq 0$ og $\rho \neq 0$ (standard), sæt $Hypot = 0$

Til $H_a: \beta < 0$ og $\rho < 0$, sæt $Hypot \leq 0$

Til $H_a: \beta > 0$ og $\rho > 0$, sæt $Hypot > 0$

En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a + b \cdot x$
stat.t	t -Statistik for signifikantest
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.fg	Frihedsgrader
stat.a, stat.b	Regressionskoefficienter
stat.s	Standardfejl for linjen
stat.SESlope	Standardfejl for hældning
stat.r ²	Forklaringsgraden
stat.r	Korrelationskoefficient
stat.Resid	Residualer fra regressionen

linSolve()

Katalog >

linSolve(*SystemafLineæreLigninger*, *Var1*, *Var2*, ...) $\Rightarrow$ liste

$$\text{linSolve}\left(\begin{cases} 2 \cdot x + 4 \cdot y = 3 \\ 5 \cdot x - 3 \cdot y = 7 \end{cases}, \{x, y\}\right) \quad \left\{ \frac{37}{26}, \frac{1}{26} \right\}$$

linSolve(*LineærLign1* og *LineærLign2* og ..., *Var1*, *Var2*, ...) $\Rightarrow$ liste

$$\text{linSolve}\left(\begin{cases} 2 \cdot x = 3 \\ 5 \cdot x - 3 \cdot y = 7 \end{cases}, \{x, y\}\right) \quad \left\{ \frac{3}{2}, \frac{1}{6} \right\}$$

linSolve(*{LineærLign1, LineærLign2, ...}*, *Var1*, *Var2*, ...) $\Rightarrow$ liste

$$\text{linSolve}\left(\begin{cases} \text{apple} + 4 \cdot \text{pear} = 23 \\ 5 \cdot \text{apple} - \text{pear} = 17 \end{cases}, \{\text{apple}, \text{pear}\}\right) \quad \left\{ \frac{13}{3}, \frac{14}{3} \right\}$$

linSolve(*SystemafLineæreLigninger*, *{Var1, Var2, ...}*) $\Rightarrow$ liste

$$\text{linSolve}\left(\begin{cases} \text{apple} \cdot 4 + \frac{\text{pear}}{3} = 14 \\ -\text{apple} + \text{pear} = 6 \end{cases}, \{\text{apple}, \text{pear}\}\right) \quad \left\{ \frac{36}{13}, \frac{114}{13} \right\}$$

linSolve(*LineærLign1* og *LineærLign2* og ..., *{Var1, Var2, ...}*) $\Rightarrow$ liste

linSolve(*{LineærLign1, LineærLign2, ...}*, *{Var1, Var2, ...}*) $\Rightarrow$

Returnerer en løsning for variablene
Var1, Var2, ...

Det første argument skal kunne
beregnes til et system af lineære
ligninger eller en enkelt lineær ligning.
Eller opstår der en argumentfejl.

For eksempel giver beregningen af
linSolve(x=1 og x=2,x) resultatet
"Argumentfejl".

Δlist()

Δlist(*Liste1*)⇒*liste*

$\Delta\text{List}(\{20,30,45,70\})$	$\{10,15,25\}$
--------------------------------------	----------------

Bemærk: Du kan indsætte denne
funktion fra computerens tastatur ved at
skrive **deltaList**(...).

Returnerer en liste med differenserne
mellem konsekutive elementer i *Liste1*.
Hvert element i *Liste1* er subtraheret fra
det næste element i *Liste1*. Den
resulterende liste er altid et element
kortere end den oprindelige *Liste1*.

list▶mat()

list▶mat(*Liste* [,
elementerPrRække])⇒*matrix*

$\text{list}\blacktriangleright\text{mat}(\{1,2,3\})$	$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$
$\text{list}\blacktriangleright\text{mat}(\{1,2,3,4,5\},2)$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 0 \end{bmatrix}$

Returnerer en matrix fyldt rækkevis med
elementerne fra *Liste*.

ElementerPrRække angiver antallet af
elementer pr. række, hvis den er
medtaget. Standard er antallet af
elementer i *Liste* (en række).

Hvis *Liste* ikke udfylder den resulterende
matrix, tilføjes nuller.

Bemærk: Du kan indsætte denne
funktion fra computerens tastatur ved at
skrive **list@>mat**(...).

Udtr ►ln⇒*udtryk*

Konverterer inputtet *Udtr* til et udtryk, der kun indeholder naturlige logaritmer (ln).

$$\left(\log_{10}(x)\right) \rightarrow \ln \frac{\ln(x)}{\ln(10)}$$

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>ln.

ln()

  -taster

ln(*Udtr1*)⇒*udtryk*

$$\ln(2.) \quad 0.693147$$

ln(*Liste1*)⇒*liste*

Returnerer den naturlige logaritme til argumentet.

Hvis kompleks formattilstand er reel:

Til en liste returneres de naturlige logaritmer af elementerne.

$$\ln\{-3, 1.2, 5\}$$

"Error: Non-real calculation"

Hvis kompleks formattilstand er rektangulær:

$$\ln\{-3, 1.2, 5\} \quad \{\ln(3) + \pi \cdot i, 0.182322, \ln(5)\}$$

ln(*kvadratMatrix1*)⇒*kvadratMatrix*

Returnerer den naturlige matrixlogaritme af *kvadratMatrix1*. Dette er ikke det samme som at beregne den naturlige logaritme af hvert element. Oplysninger om beregningsmetoden findes under **cos()**.

I vinkeltilstanden radian og rektangulært komplekst format:

$$\ln \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 1.83145+1.73485 \cdot i & 0.009193-1.49086 \\ 0.448761-0.725533 \cdot i & 1.06491+0.623491 \cdot i \\ -0.266891-2.08316 \cdot i & 1.12436+1.79018 \cdot i \end{bmatrix}$$

KvadratMatrix1 skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

Du kan se hele resultatet ved at trykke på ► og derefter bruge ◀ og ▶ til at bevæge markøren.

LnReg

Katalog > 

LnReg *X*, *Y*, [*Frekv*] [, *Kategori*, *Medtag*]

Beregner den lineære regression $y = a + b \cdot \ln(x)$ på liste X og Y med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

X og Y er lister med uafhængige og afhængige variable.

Frekv er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende X og Y datapunkt. Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

Category er en liste, der indeholder kategorikoder for X og Y data..

Medtag er en liste med en eller flere af kategorikoderne. Kun de dataelementer, hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a + b \cdot \ln(x)$
stat.a, stat.b	Regressionskoefficienter
stat.r ²	Koefficient af en lineær forklaringsgrad til transformerede data
stat.r	Korrelationskoefficient til transformerede data ($\ln(x)$, y)
stat.Resid	Residualer forbundet med eksponentielmodellen
stat.ResidTrans	Residualer associeret med lineær tilpasning af transformerede data
stat.XReg	Liste af datapunkter i den modificerede <i>X-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

Local *Var1* [, *Var2*] [, *Var3*] ...

Erklærer de angivne *var* som lokale variable. Disse variable eksisterer kun under beregning af en funktion og slettes, når eksekveringen af funktionen afsluttes.

Bemærk: Lokale variable sparer hukommelse, fordi kun eksisterer midlertidigt. De forstyrrer heller ikke de eksisterende globale variabelværdier. Lokale variable skal anvendes til **For**-løkker og midlertidig lagring af variabelværdier i en flerlinjefunktion da modifikationer af globale ikke er tilladt i en funktion.

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

```
Define rollcount()=Func
```

```
Local i
1 → i
Loop
If randInt(1,6)=randInt(1,6)
Goto end
i+1 → i
EndLoop
Lbl end
Return i
EndFunc
```

Done

<i>rollcount()</i>	16
--------------------	----

<i>rollcount()</i>	3
--------------------	---

Lock**Lock** *Var1* [, *Var2*] [, *Var3*] ...**Lock** *Var*.

Låser de angivne variable eller den angivne variabelgruppe. Låste variable kan ikke redigeres eller slettes.

Du kan ikke låse eller oplåse systemvariabeln *Ans*, og du kan ikke låse systemvariabelgrupperne *stat*. eller *tvm*.

Bemærk: Kommandoen **Lås (Lock)** rydder Fortryd/Annuller Fortryd-historikken, når den anvendes på ulåste variable.

Se **unLock**, side 209 og **getLockInfo()**, side 88.

<i>a:=65</i>	65
--------------	----

Lock <i>a</i>	<i>Done</i>
---------------	-------------

getLockInfo(<i>a</i>)	1
-------------------------	---

<i>a:=75</i>	"Error: Variable is locked."
--------------	------------------------------

DelVar <i>a</i>	"Error: Variable is locked."
-----------------	------------------------------

Unlock <i>a</i>	<i>Done</i>
-----------------	-------------

<i>a:=75</i>	75
--------------	----

DelVar <i>a</i>	<i>Done</i>
-----------------	-------------

log()

ctrl 10^x -taster

log(*Udtr1* [, *Udtr2*]) ⇒ *udtryk*

$$\log_{10} (2.) \quad 0.30103$$

log(*Liste1* [, *Udtr2*]) ⇒ *liste*

$$\log_4 (2.) \quad 0.5$$

Returner -*Udtr2*-talslogaritmen til argumentet.

$$\log_3 (10) - \log_3 (5) \quad \log_3 (2)$$

Bemærk: Se også **Log-skabelon**, side 2.

Ved en liste returneres *Udtr2*-talslogaritmen til elementerne.

Hvis kompleks formattilstand er reel:

$$\log_{10} (\{-3, 1.2, 5\}) \quad \text{Error: Non-real result}$$

Hvis *Udtr2* udelades, anvendes 10-talslogaritmen.

Hvis kompleks formattilstand er rektangulær:

$$\log_{10} (\{-3, 1.2, 5\}) \\ \left\{ \log_{10} (3) + 1.36438 \cdot i, 0.079181, \log_{10} (5) \right\}$$

log(*kvadratMatrix1* [, *Udtr2*]) ⇒ *kvadratMatrix*

I vinkeltilstanden radian og rektangulært komplekst format:

$$\log_{10} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \\ \begin{bmatrix} 0.795387 + 0.753438 \cdot i & 0.003993 - 0.6474 \cdot i \\ 0.194895 - 0.315095 \cdot i & 0.462485 + 0.2707 \cdot i \\ -0.115909 - 0.904706 \cdot i & 0.488304 + 0.7774 \cdot i \end{bmatrix}$$

Returnerer *Udtr2*-talsmatrixlogaritmen til *kvadratMatrix1*. Dette er ikke det samme som at beregne *Udtr2*-talslogaritmen til hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

KvadratMatrix1 skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

Hvis tal-argumentet udelades, anvendes 10-talslogaritmen.

logbase

Katalog > 

Udtr1 ▶ **logbase**(*Udtr2*) ⇒ *udtryk*

Omformer inputudtrykket til et udtryk, der anvender *Udtr2*-talslogaritmen.

$$\log_3 (10) - \log_5 (5) \blacktriangleright \log_{\text{base}(5)} \left(\frac{\log_5 \left(\frac{10}{3} \right)}{\log_5 (3)} \right)$$

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>**logbase** (...).

Logistik X , Y , [*Frekv*] [, *Kategori*, *Medtag*]

Beregner den logistiske regression $y = c / (1 + a \cdot e^{-bx})$ på listerne X og Y med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

X og Y er lister med uafhængige og afhængige variable.

Frekv er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende X og Y datapunkt. Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

Kategori er en liste, der indeholder kategorikoder for X og Y data..

Medtag er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $c / (1 + a \cdot e^{-bx})$
stat.a, stat.b, stat.c	Regressionskoefficienter
stat.Resid	Residualer fra regressionen
stat.XReg	Liste af datapunkter i den modificerede <i>X-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

LogisticD X, Y [, [*Iterationer*], [*Frekv*] [, *Kategori*, *Medtag*]]

Beregner den logistiske regression $y = (c / (1 + a \cdot e^{-bx}) + d)$ på listerne X og Y med hyppighed *Frekv*, ved brug af et angivet tal fra *Iterationer*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

X og Y er lister med uafhængige og afhængige variable.

Iterationer er en valgfri værdi, som angiver det maksimale antal gange en løsning vil forsøges. Hvis udeladt, anvendes 64. Typisk resulterer større værdier i større nøjagtighed men længere eksekveringstider og omvendt.

Frekv er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende X og Y datapunkt. Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

Kategori er en liste, der indeholder kategorikoder for X og Y data..

Medtag er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $c / (1 + a \cdot e^{-bx}) + d$
stat.a, stat.b, stat.c, stat.d	Regressionskoefficienter
stat.Resid	Residualer fra regressionen

Output-variabel	Beskrivelse
stat.XReg	Liste af datapunkter i den modificerede <i>X-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>Kategori liste</i> og <i>Medtag Kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>Kategori liste</i> og <i>Medtag Kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

Loop

Katalog > 

Loop

Blok

EndLoop

Eksekverer gentagne gange sætningerne i *Blok*. Bemærk, at løkken eksekveres uendeligt, medmindre en **Goto** eller **Exit**-kommando eksekveres i *Blok*.

Blok er en sekvens af sætninger adskilt med kolon.

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

```

Define rollcount()=Func
    Local i
    1 → i
    Loop
    If randInt(1,6)=randInt(1,6)
        Goto end
    i+1 → i
    EndLoop
    Lbl end
    Return i
EndFunc

```

	Done
<i>rollcount()</i>	16
<i>rollcount()</i>	3

LU *Matrix*, *lMatrix*, *uMatrix*, *pMatrix* [, *Tol*]

Beregner Doolittle LU (nedre-øvre) opløsningen af en reel eller kompleks matrix. Den nedre triangulære matrix lagres i *lMatrix*, den øvre triangulære matrix i *uMatrix*, og permutationsmatricen (der beskriver de foretagne rækkeombytninger under beregningen) i *pMatrix*.

$$lMatrix \cdot uMatrix = pMatrix \cdot matrix$$

Ethvert matricelement kan valgfrit behandles som nul, hvis dets absolutte værdi er mindre end *Tol*. Denne tolerance anvendes kun, hvis matricen har elementer med flydende decimaler og ikke indeholder symbolske variable, der ikke er tildelt en værdi. Ellers ignoreres *Tol*.

- Hvis du anvender   eller indstiller **Auto eller tilnærmet** - tilstanden til Approximate, foretages beregningerne med aritmetik med flydende komma.
- Hvis *Tol* udelades eller ikke anvendes, beregnes standardtolerancen som:
- $5E-14 \cdot \max(\dim(Matrix)) \cdot \text{rækkeNorm}(Matrix)$

LU-algoritmen til faktoropløsning anvender partiel pivotering med rækkeombytninger.

$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix}$
---	--

LU *m1*, *lower*, *upper*, *perm* Done

<i>lower</i>	$\begin{bmatrix} 1 & 0 & 0 \\ \frac{5}{6} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$
--------------	---

<i>upper</i>	$\begin{bmatrix} 6 & 12 & 18 \\ 0 & 4 & 16 \\ 0 & 0 & 1 \end{bmatrix}$
--------------	--

<i>perm</i>	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
-------------	---

$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} m & n \\ o & p \end{bmatrix}$
---	--

LU *m1*, *lower*, *upper*, *perm* Done

<i>lower</i>	$\begin{bmatrix} 1 & 0 \\ \frac{m}{o} & 1 \\ o & \end{bmatrix}$
--------------	---

<i>upper</i>	$\begin{bmatrix} o & p \\ 0 & n - \frac{m \cdot p}{o} \\ o & \end{bmatrix}$
--------------	---

<i>perm</i>	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
-------------	--

M

mat▶list()

mat▶list(*Matrix*) ⇒ *liste*

Returner en liste bestående af elementerne i *Matrix*. Elementerne kopieres fra *Matrix* række for række.

mat▶list ($\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$)	$\{1, 2, 3\}$
---	---------------

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
---	--

mat▶list (<i>m1</i>)	$\{1, 2, 3, 4, 5, 6\}$
-------------------------------	------------------------

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive `mat@>list(...)`.

max()

Katalog >

max(*Udtr1*, *Udtr2*) $\Rightarrow$ *udtryk*

$\max\{2.3, 1.4\}$	2.3
--------------------	-----

max(*Liste1*, *Liste2*) $\Rightarrow$ *liste*

$\max\{\{1, 2\}, \{-4, 3\}\}$	$\{1, 3\}$
-------------------------------	------------

max(*Matrix1*, *Matrix2*) $\Rightarrow$ *matrix*

Returnerer maksimum af de to argumenter. Hvis argumenterne er to lister eller matricer, returneres en liste eller matrix med maksimumsværdier for hvert sammenhørende elementpar.

max(*List*) $\Rightarrow$ *udtryk*

$\max\{0, 1, -7, 1.3, 0.5\}$	1.3
------------------------------	-----

Returnerer det største element i *liste*.

max(*Matrix1*) $\Rightarrow$ *matrix*

$\max\left(\begin{bmatrix} 1 & -3 & 7 \\ -4 & 0 & 0.3 \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 0 & 7 \end{bmatrix}$
---	---

Returnerer en rækkevektor med det største element i hver kolonne i *Matrix1*.

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 266

Bemærk: Se også **fMax()** og **min()**.

mean()

Katalog >

mean(*Liste*[, *hyppighedsliste*]) $\Rightarrow$ *udtryk*

$\text{mean}(\{0.2, 0.1, -0.3, 0.4\})$	0.26
--	------

Returnerer middelværdien for elementerne i *Liste*.

$\text{mean}(\{1, 2, 3\}, \{3, 2, 1\})$	$\frac{5}{3}$
---	---------------

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

mean(*Matrix1*[, *Hyppighedsmatrix*]) $\Rightarrow$ *matrix*

I rektangulært vektorformat:

Returnerer en rækkevektor af middelværdierne af alle kolonner i *Matrix1*.

mean()Katalog > 

Hvert *Hyppighedsmatrix*-element tæller antallet af konsekutive forekomster af det tilsvarende element i *Matrix1*.

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 266.

mean	$\begin{pmatrix} 0.2 & 0 \\ -1 & 3 \\ 0.4 & -0.5 \end{pmatrix}$	$\begin{bmatrix} -0.133333 & 0.833333 \end{bmatrix}$
mean	$\begin{pmatrix} \frac{1}{5} & 0 \\ -1 & 3 \\ \frac{2}{5} & \frac{-1}{2} \end{pmatrix}$	$\begin{bmatrix} \frac{-2}{15} & \frac{5}{6} \end{bmatrix}$
mean	$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}, \begin{pmatrix} 5 & 3 \\ 4 & 1 \\ 6 & 2 \end{pmatrix}$	$\begin{bmatrix} \frac{47}{15} & \frac{11}{3} \end{bmatrix}$

median()Katalog > 

median(*Liste*[,
Hyppighedsliste])⇒*udtryk*

Returnerer medianen af elementerne i *Liste*.

Hvert *Hyppighedsliste*-element tæller antallet af forekomster i ubrudt rækkefølge for de tilsvarende elementer i *Liste*.

median(*Matrix1*[,
Hyppighedsmatrix])⇒*matrix*

Returnerer en rækkevektor med medianerne af kolonnerne i *Matrix1*.

Hvert *Hyppighedsmatrix*-element tæller antallet af forekomster i en ubrudt rækkefølge af det tilsvarende element i *Matrix1*.

median	$\{0.2, 0.1, -0.3, 0.4\}$	0.2
median	$\begin{pmatrix} 0.2 & 0 \\ 1 & -0.3 \\ 0.4 & -0.5 \end{pmatrix}$	$\begin{bmatrix} 0.4 & -0.3 \end{bmatrix}$

Noter:

- Alle elementer i listen eller matricen skal kunne omregnes til tal.
- tomme (ugyldige) elementer i listen eller matricen ignoreres. Yderligere oplysninger om tomme elementer findes på side 266.

MedMed X, Y [, *Frekv*] [, *Kategori*, *Medtag*]

Beregner median-median linje $y = (m \cdot x + b)$ på listerne X og Y med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

X og Y er lister med uafhængige og afhængige variable.

Frekv er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende X og Y datapunkt. Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

Kategori er en liste, der indeholder kategorikoder for X og Y data..

Medtag er en liste med en eller flere af kategorikoderne. Kun de dataelementer, hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.RegEqn	Median-median-linjeligning: $m \cdot x + b$
stat.m, stat.b	Modelkoefficienter
stat.Resid	Residualer fra median-median-linjen
stat.XReg	Liste af datapunkter i den modificerede X -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede Y -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

mid()

Katalog > 

mid(kildeStreng, Start[, Antal])⇒streng

Returnerer *Antal* tegn fra tegnstrengen *kildeStreng*, startende med tegn nummer *Start*.

Hvis *Antal* udelades eller er større end dimension på *kildeStreng*, returneres alle tegn fra *kildeStreng*, begyndende med tegn nummer *Start*.

Antal skal være ≥ 0 . Hvis *Antal* = 0, returneres en tom streng.

mid(kildeListe, Start [, Antal])⇒liste

Returnerer *Antal* elementer fra *kildeListe*, begyndende med element nummer *Start*.

Hvis *Antal* udelades eller er større end dimensionen på *kildeListe*, returneres alle elementer fra *kildeListe*, begyndende med element nummer *Start*.

Antal skal være ≥ 0 . Hvis antal = 0, returneres en tom liste.

mid(kildeStrengListe, Start[, Antal])⇒liste

Returnerer *Antal* strenge fra listen med strenge *kildeStrengListe* begyndende med element nummer *Start*.

mid("Hello there",2)	"ello there"
mid("Hello there",7,3)	"the"
mid("Hello there",1,5)	"Hello"
mid("Hello there",1,0)	" "

mid({9,8,7,6},3)	{7,6}
mid({9,8,7,6},2,2)	{8,7}
mid({9,8,7,6},1,2)	{9,8}
mid({9,8,7,6},1,0)	{ }

mid({"A","B","C","D"},2,2)	{"B","C"}
----------------------------	-----------

min()

Katalog > 

min(Udtr1, Udtr2)⇒udtryk

min(Liste1, Liste2)⇒liste

min(Matrix1, Matrix2)⇒matrix

Returnerer minimum af de to argumenter. Hvis argumenterne er to lister eller matricer, returneres en liste eller matrix med minimumværdi af hvert sammenhørende elementpar.

min(Liste)⇒udtryk

min(2,3,1,4)	1.4
min({1,2},{-4,3})	{-4,2}

min({0,1,-7,1.3,0.5})	-7
-----------------------	----

Returnerer det mindste element af *Liste*.

min(*Matrix1*) $\Rightarrow$ *matrix*

Returnerer en rækkevektor med det mindste element i hver kolonne i *Matrix1*.

$$\min \begin{pmatrix} 1 & -3 & 7 \\ -4 & 0 & 0.3 \end{pmatrix} \quad \begin{bmatrix} -4 & -3 & 0.3 \end{bmatrix}$$

Bemærk: Se også **fMin()** og **max()**.

mirr

(*finansRente*, *geninvestRente*, *CF0*, *CFListe*, *CFFrekv*)

Finansfunktion, der returnerer den modificerede interne rente af en investering.

$$\begin{aligned} list1 &:= \{6000, -8000, 2000, -3000\} \\ &\quad \{6000, -8000, 2000, -3000\} \\ list2 &:= \{2, 2, 2, 1\} \\ &\quad \{2, 2, 2, 1\} \\ mirr(4.65, 12, 5000, list1, list2) &\quad 13.41608607 \end{aligned}$$

finansRente er rentesatsen, du betaler for pengestrømsbeløbene.

geninvestRente er rentesatsen, som pengestrømmen geninvesteres til.

CF0 er startpengestrømmen på tidspunkt 0. Den skal være et reelt tal.

CFListe er en liste over pengestrømsbeløb efter startpengestrømmen *CF0*.

CFFrekv er en valgfri liste, hvor hvert element angiver hyppigheden for et grupperet (fortløbende) pengestrømsbeløb, som er det tilsvarende element i *CFListe*. Standardværdien er 1. Hvis du indtaster værdier, skal de være positive heltal < 10.000.

Bemærk: Se også **irr()**, side 99.

mod()Katalog > **mod**(*Udtr1*, *Udtr2*) $\Rightarrow$ *udtryk* $\text{mod}(7,0)$ 7**mod**(*Liste1*, *Liste2*) $\Rightarrow$ *liste* $\text{mod}(7,3)$ 1**mod**(*Matrix1*, *Matrix2*) $\Rightarrow$ *matrix* $\text{mod}(-7,3)$ 2

Returnerer det første argument modulo andet argument som defineret efter definitionen:

 $\text{mod}(7,-3)$ -2 $\text{mod}(-7,-3)$ -1 $\text{mod}(\{12,-14,16\},\{9,7,-5\})$ $\{3,0,-4\}$ $\text{mod}(x,0) = x$ $\text{mod}(x,y) = x - y \text{ floor}(x/y)$

Når det andet argument er ikke-nul, er resultatet periodisk i det pågældende argument. Resultatet er enten nul eller har samme fortegn som det andet argument.

Hvis argumenterne er to lister eller to matricer, returneres en liste eller matrix med modulo af hvert par af sammenhørende elementer.

Bemærk: Se også **remain()**, side 157

mRow()Katalog > **mRow**(*Udtr*, *Matrix1*, *Indeks*) $\Rightarrow$ *matrix*

Returnerer en kopi af *Matrix1* med hvert element i rækken *Indeks* af *Matrix1* ganget med *Udtr*.

$$\text{mRow}\left(\frac{-1}{3}, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 2\right) \quad \begin{bmatrix} 1 & 2 \\ -1 & -4 \\ 3 & \end{bmatrix}$$
mRowAdd()Katalog > **mRowAdd**(*Udtr*, *Matrix1*, *Indeks1*, *Indeks2*) $\Rightarrow$ *matrix*

Returnerer en kopi af *Matrix1* med hvert element i rækken *Indeks2* af *Matrix1* erstattet med:

$$\text{mRowAdd}\left(-3, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 1, 2\right) \quad \begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix}$$

$$\text{mRowAdd}\left(n, \begin{bmatrix} a & b \\ c & d \end{bmatrix}, 1, 2\right) \quad \begin{bmatrix} a & b \\ a \cdot n + c & b \cdot n + d \end{bmatrix}$$

$Udtr \times \text{række } Indeks1 + \text{række } Indeks2$

MultReg $Y, X1[,X2[,X3,...[,X10]]]$

Beregner multiple lineære regressioner af listen Y på listerne $X1, X2, \dots, X10$. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Alle lister skal have samme dimension.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $b_0+b_1 \cdot x_1+b_2 \cdot x_2+ \dots$
stat.b0, stat.b1, ...	Regressionskoefficienter
stat.r ²	Koefficient af multipel forklaringsgrad
stat. $\hat{y}$ List	$\hat{y}$ Liste = $b_0+b_1 \cdot x_1+ \dots$
stat.Resid	Residualer fra regressionen

MultRegIntervals**MultRegIntervals** $Y, X1[,X2[,X3,...[,X10]]], XValListe[,CNiveau]$

Beregner en forudset y -værdi, et niveau C forudsigelsesinterval for enkle observationer, og et niveau C konfidensinterval til gennemsnits-responsen.

En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Alle lister skal have samme dimension.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $b_0+b_1 \cdot x_1+b_2 \cdot x_2+ \dots$
Statistik. $\hat{y}$	Et punktestimat: $\hat{y} = b_0 + b_1 \cdot x_1 + \dots$ for <i>XValListe</i>
stat.dfFejl (stat.dfError)	Frihedsgrader for fejl

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval for en middelværdi
stat.ME	Konfidensinterval, fejlmargen
stat.SE	Standardfejl for middelværdi
stat.LowerPred, stat.UpperPred	Prædiktionsinterval for en enkelt observation
stat.MEPred	Prædiktionsintervalsmargin for fejl
stat.SEPred	Standardfejl for prædiktion
stat.bList	Liste med regressionskoefficienter, {b0,b1,b2,...}
stat.Resid	Residualer fra regressionen

MultRegTests

Katalog > 

MultRegTests $Y, X1[,X2[,X3,...[,X10]]]$

Multipel lineær regressionstest beregner en multipel lineær regression fra de givne data, og danner den globale F teststatistik og t teststatistikker for koefficienterne.

En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $b_0+b_1 \cdot x_1+b_2 \cdot x_2+ \dots$
stat.F	Global F teststatistik
stat.PVal	P-værdi tilknyttet global F statistik
stat.r ²	Koefficient af multipel forklaringsgrad
stat.AdjR ²	Justeret koefficient af multipel forklaringsgrad
stat.s	Standardafvigelse for fejlen
stat.DW	Durbin-Watson-statistik; Anvendes til at bestemme, om første-ordens auto-korrelationen er tilstede i modellen

Output-variabel	Beskrivelse
stat.dfReg	Frihedsgrader i regressionen
stat.SSReg	Kvadraternes regressionsum
stat.MSReg	Middelkvadrat af regression
stat.dfFejl	Frihedsgrader for fejl
stat.SSError	fejl, kvadratsum
stat.MSError	fejl, middelkvadrat
stat.bList	{b0,b1,...} Liste med koefficienter
stat.tList	Liste med t statistikker for hver koefficient i bListen
stat.PList	Liste P-værdier for hver t-statistik
stat.SEList	Liste med standardfejl for koefficienter i bListe
stat.ŷList	$\hat{y}$ Liste = $b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Residualer fra regressionen
stat.sResid	Standardiserede residualer; Værdi opnået ved at dividere en residual med dens standardafvigelse
stat.CookDist	Cooks distance; Mål for påvirkningen af en observation baseret på residual og udnyttelse
stat.Leverage	Mål for, hvor langt værdierne for de uafhængige variable er fra deres middelværdier

nand

  -taster

BoolskUdtryk1 **nand** *BoolskUdtryk2*
returnerer *boolsk udtryk*

$x \geq 3$ and $x \geq 4$

$x \geq 4$

BoolskListe1 **nand** *BoolskListe2*
returnerer *Boolsk liste*

$x \geq 3$ nand $x \geq 4$

$x < 4$

BoolskMatrix1 **nand** *BoolskMatrix2*
returnerer *Boolsk matrix*

Returnerer negationen af en logisk **and** operation anvendt på de to argumenter. Returnerer true eller false eller en forenklet form af ligningen.

For lister og matricer returneres sandhedsværdierne element for element.

Heltal1 **nand** *Heltal2* $\Rightarrow$ *heltal*

Sammenligner to reelle heltal bit for bit med en **nand** operation. Internt konverteres begge heltal til 64-bit binære tal med fortegn. Når de tilsvarende bits sammenlignes, er resultatet 1, hvis begge bits er 1. Ellers er resultatet 0. Den returnerede værdi repræsenterer bit-resultaterne og vises i overensstemmelse med den valgte talsystemtilstand.

Du kan indtaste heltallene i ethvert talsystem. Til binære eller hexadecimalt indtastninger skal du som præfiks benytte henholdsvis 0b eller 0h. Uden præfiks behandles heltallene som decimaltal (10 talsystem).

3 and 4	0
3 nand 4	-1
$\{1,2,3\}$ and $\{3,2,1\}$	$\{1,2,1\}$
$\{1,2,3\}$ nand $\{3,2,1\}$	$\{-2,-3,-2\}$

N

nCr()

Katalog > 

nCr(Udtr1, Udtr2) $\Rightarrow$ *udtryk*

For heltal *Udtr1* og *Udtr2* med $Udtr2 \geq Udtr1 \geq 0$, **nCr()** er antallet af kombinationer af *Udtr1* ting taget *Udtr2* ad gangen. (Dette kendes også som en binomial koefficient). Begge argumenter kan være heltal eller symbolske udtryk.

$nCr(z,3)$	$\frac{z \cdot (z-2) \cdot (z-1)}{6}$
$Ans z=5$	10
$nCr(z,c)$	$\frac{z!}{c! \cdot (z-c)!}$
$\frac{Ans}{nPr(z,c)}$	$\frac{1}{c!}$

nCr(Udtr, 0) $\Rightarrow$ 1

nCr(Udtr, negativtHeltal) $\Rightarrow$ 0

nCr(Udtr, positivtHeltal) $\Rightarrow Udtr \cdot (Udtr-1) \cdot \dots \cdot (Udtr-positivtHeltal+1) / positivtHeltal!$

nCr(Udtr, ikkeHeltal) $\Rightarrow$ *udtryk*!
 $((Udtr-ikkeHeltal)! \cdot ikkeHeltal!)$

nCr(Liste1, Liste2) $\Rightarrow$ *liste*

$nCr(\{5,4,3\}, \{2,4,2\})$	$\{10,1,3\}$
-----------------------------	--------------

nCr()Katalog > 

Returnerer en liste med kombinationer baseret på de sammenhørende elementpar i de to lister. Argumenterne skal være lister af samme størrelse.

nCr(*Matrix1*, *Matrix2*) $\Rightarrow$ *matrix*

Returnerer en matrix af kombinationer baseret på de sammenhørende elementpar i de to matricer. Argumenterne skal være matricer af samme størrelse.

$$\text{nCr}\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right) \quad \begin{bmatrix} 15 & 10 \\ 6 & 3 \end{bmatrix}$$

nDerivative()Katalog > 

nDerivative(*Udtr1*, *Var*=*Værdi* [*,Orden*]) $\Rightarrow$ *værdi*

nDerivative(*Udtr1*, *Var*, [*Orden*]) | *Var*=*Værdi* $\Rightarrow$ *værdi*

Returnerer den numeriske differentialkvotient udregnet med en automatisk differentiationsmetode.

Når *Værdi* er angivet, tilsidesætter den alle forudgående variabeltildelinger og alle nuværende “|” substitutioner for variabelen.

Differentialkvotientens orden skal være 1 eller 2.

$\text{nDerivative}(x , x=1)$	1
$\text{nDerivative}(x , x) _{x=0}$	undef
$\text{nDerivative}(\sqrt{x-1}, x) _{x=1}$	undef

newList()Katalog > 

newList(*antalElementer*) $\Rightarrow$ *liste*

Returnerer en liste med en dimension af *antalElementer*. Hvert element er nul.

$$\text{newList}(4) \quad \{0, 0, 0, 0\}$$

newMat()Katalog > 

newMat(*antalRækker*, *antalKolonner*) $\Rightarrow$ *matrix*

Returnerer en matrix med nulpunkter af dimensionen *antalRækker* gange *antalKolonner*.

$$\text{newMat}(2, 3) \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

nfMax()Katalog > **nfMax**(*Udtr*, *Var*)⇒*værdi*

$\text{nfMax}\left(-x^2-2\cdot x-1,x\right)$	-1.
--	-----

nfMax(*Udtr*, *Var*,
nedreGrænse)⇒*værdi*

$\text{nfMax}\left(0.5\cdot x^3-x-2,x,-5,5\right)$	5.
--	----

nfMax(*Udtr*, *Var*, *nedreGrænse*,
øvreGrænse)⇒*værdi***nfMax**(*Udtr*, *Var*) | *nedreGrænse*≤*Var*
≤*øvreGrænse*⇒*værdi*

Returnerer mulig numerisk værdi for
variablen *Var*, hvor det lokale
maksimum for *Udtr* optræder.

Hvis du opgiver *nedreGrænse* og
øvreGrænse, søger funktionen i det
lukkede interval
[*nedreGrænse*,*øvreGrænse*] efter det
lokale maksimum.

Bemærk: Se også **fMax()** og **d()**.**nfMin()**Katalog > **nfMin**(*Udtr*, *Var*)⇒*værdi*

$\text{nfMin}\left(x^2+2\cdot x+5,x\right)$	-1.
---	-----

nfMin(*Udtr*, *Var*, *nedreGrænse*)⇒*værdi*

$\text{nfMin}\left(0.5\cdot x^3-x-2,x,-5,5\right)$	-5.
--	-----

nfMin(*Udtr*, *Var*, *nedreGrænse*,
øvreGrænse)⇒*værdi***nfMin**(*Udtr*, *Var*) | *nedreGrænse*≤*Var*
≤*øvreGrænse*⇒*værdi*

Returnerer mulig numerisk værdi for
variablen *Var*, hvor det lokale minimum
for *Udtr* optræder.

Hvis du opgiver *nedreGrænse* og
øvreGrænse, søger funktionen i det
lukkede interval
[*nedreGrænse*,*øvreGrænse*] efter det
lokale minimum.

Bemærk: Se også **fMin()** og **d()**.

nInt()

Katalog > 

nInt(Udtr1, Var, nedre, øvre)⇒udtryk

$$\text{nInt}\left(e^{-x^2}, x, -1, 1\right) \quad 1.49365$$

Hvis integranden *Udtr1* ikke indeholder andre variable end *Var*, og hvis *Nedre* og *Øvre* er konstante, $+\infty$ eller $-\infty$, så returnerer **nInt()** en tilnærmet værdi af $\int (Udtr1, Var, Nedre, Øvre)$. Denne tilnærmede værdi er et vægtet gennemsnit af nogle eksempelverdier af integranden i intervallet $Nedre < Var < Øvre$.

Målet er seks betydende cifre.

Algoritmen, der kan tilpasses, afsluttes hvis det virker sandsynligt, at målet er nået, eller når det virker usandsynligt, at ydeligere eksempler vil give en væsentlig forbedring.

$$\text{nInt}\left(\cos(x), x, \pi, \pi+1.E-12\right) \quad -1.04144E-12$$

$$\int_{\pi}^{\pi+10^{-12}} \cos(x) dx \quad -\sin\left(\frac{1}{1000000000000}\right)$$

Der vises en advarsel ("Tvivl om nøjagtighed") når målet ikke ser ud til at være nået.

Indskyd flere **nInt()**, for at foretage numerisk integration i flere variable. Integrationsgrænser kan afhænge af integrationsvariable uden for dem.

$$\text{nInt}\left(\text{nInt}\left(\frac{e^{-x \cdot y}}{\sqrt{x^2 - y^2}}, y, -x, x\right), x, 0, 1\right) \quad 3.30423$$

Bemærk: Se også **j()**, side 235.

nom()

Katalog > 

nom(effektivRente, CpY)⇒værdi

$$\text{nom}(5.90398, 12) \quad 5.75$$

Finansfunktion, der omregner den effektive årlige rente *effektivRente* til en nominal rente, hvor *CpY* er antallet af rentetilskrivninger per år.

effektivRente skal være et reelt tal, og *CpY* skal være et reelt tal > 0 .

Bemærk: Se også **eff()**, side 62.

nor

-taster

BoolskUdtryk1 **nor** *BoolskUdtryk2*
returnerer *boolsk udtryk*

$$x \geq 3 \text{ or } x \geq 4 \quad x \geq 3$$

BoolskListe1 **nor** *BoolskListe2*

$$x \geq 3 \text{ nor } x \geq 4 \quad x < 3$$

returnerer *Boolsk liste*

BoolskMatrix1 **nor** *BoolskMatrix2*

returnerer *Boolsk matrix*

Returnerer negationen af en logisk **or** operation anvendt på de to argumenter. Returnerer true eller false eller en forenklet form af ligningen.

For lister og matricer returneres sandhedsværdierne element for element.

Heltal1 **nor** *Heltal2* $\Rightarrow$ *heltal*

Sammenligner to reelle heltal bit for bit med en **nor** operation. Internt konverteres begge heltal til 64-bit binære tal med fortegn. Når de tilsvarende bits sammenlignes, er resultatet 0, hvis begge bits er 1. Ellers er resultatet 1. Den returnerede værdi repræsenterer bit-resultaterne og vises i overensstemmelse med den valgte talsystemtilstand.

Du kan indtaste heltallene i ethvert talsystem. Til binære eller hexadecimale indtastninger skal du som præfiks benytte henholdsvis 0b eller 0h. Uden præfiks behandles heltallene som decimaltal (10 talssystem)

3 or 4	7
3 nor 4	-8
$\{1,2,3\}$ or $\{3,2,1\}$	$\{3,2,3\}$
$\{1,2,3\}$ nor $\{3,2,1\}$	$\{-4,-3,-4\}$

norm()

Katalog > 

norm(*Matrix*) $\Rightarrow$ *udtryk*

norm(*Vektor*) $\Rightarrow$ *udtryk*

Returnerer Frobenius-normen.

$\text{norm}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)$	$\sqrt{a^2+b^2+c^2+d^2}$
$\text{norm}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right)$	$\sqrt{30}$
$\text{norm}\left(\begin{bmatrix} 1 & 2 \end{bmatrix}\right)$	$\sqrt{5}$
$\text{norm}\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$	$\sqrt{5}$

normalLine()Katalog > **normalLine**(*Udtr1*,*Var*,*Punkt*)⇒*udtryk***normalLine**(*Udtr1*,*Var*=*Punkt*)⇒*udtryk*

Returnerer normallinjen til kurven repræsenteret ved *Udtr1* i punktet angivet i *Var*=*Punkt*.

Vær sikker på, at den uafhængige variabel ikke er defineret. For eksempel, Hvis $f1(x)=5$ og $x:=3$, så returnerer **normalLine**($f1(x)$, x ,2) "false."

$\text{normalLine}(x^2, x, 1)$	$\frac{3}{2} - \frac{x}{2}$
$\text{normalLine}((x-3)^2 - 4, x, 3)$	$x=3$
$\text{normalLine}(\frac{1}{x^3}, x=0)$	0
$\text{normalLine}(\sqrt{ x }, x=0)$	undef

normCdf()Katalog > 

normCdf(*nedreGrænse*,*øvreGrænse*[, μ [σ]])⇒*tal* hvis *nedreGrænse* og *øvreGrænse* er tal, *liste* hvis *nedreGrænse* og *øvreGrænse* er lister

Beregner normalfordelingens sandsynlighed mellem *nedreGrænse* og *øvreGrænse* for de angivne μ (standard=0) og σ (standard=1).

For $P(X \leq \text{øvreGrænse})$, sæt *nedreGrænse*= $-\infty$.

normPdf()Katalog > 

normPdf(*XVærdi*[, μ , σ])⇒*tal* hvis *XVærdi* er et tal, *liste* hvis *XVærdi* er en liste

Beregner tæthedsfunktionen for normalfordelingen i en angivet *XVærdi* for de angivne μ og σ .

notKatalog > **not** *Boolsk udtr1*⇒*Boolsk udtryk*

Returnerer true eller false eller en forenklet form af argumentet.

not *Heltal1*⇒*heltal*

$\text{not}(2 \geq 3)$	true
$\text{not}(x < 2)$	$x \geq 2$
not not innocent	<i>innocent</i>

I hexadecimal tilstand:

Vigtigt: Tallet nul, ikke bogstavet O.

Returnerer 1's komplement til et reelt heltal. Internt konverteres *Heltal1* til et 64-bit binært tal med fortegn. Værdien af hver bit vendes (0 bliver 1, og omvendt) for 1's komplement. Resultatet vises i den valgte tilstand for talsystem.

Du kan indtaste heltallet i ethvert talsystem. Til binære eller hexadecimal indtastninger skal du som præfiks benytte henholdsvis 0b eller 0h. Uden præfiks behandles heltallet som decimaltal (10-talssystem).

Hvis du indtaster et decimalt heltal, der er for stort til en 64-bit binær form med fortegn, anvendes en symmetrisk modulo-operation til at bringe værdien ind i det korrekte område. Yderligere oplysninger findes under ►Base2, side 18.

not 0h7AC36	0hFFFFFFFFF853C9
-------------	------------------

I binær tilstand:

0b100101 ►Base10	37
------------------	----

not 0b100101

0b11111111111111111111111111111111 ►

not 0b100101 ►Base10	-38
----------------------	-----

Du kan se hele resultatet ved at trykke på ► og derefter bruge ◀ og ▶ til at bevæge markøren.

Bemærk: En binær indtastning kan have op til 64 cifre (præfikset 0b ikke medregnet). En hexadecimal indtastning kan have op til 16 cifre.

nPr()

nPr(*Udtr1*, *Udtr2*) ⇒ *udtryk*

For heltal er *Udtr1* og *Udtr2* med $Udtr1 \geq Udtr2 \geq 0$, **nPr**() antallet af permutationer af *Udtr1* ting taget *Udtr2* ad gangen. Begge argumenter kan være heltal eller symbolske udtryk.

nPr(*Udtr*, 0) ⇒ 1

nPr(*Udtr*, *negHeltal*) ⇒ $1 / ((Udtr+1) \cdot (Udtr+2) \dots (Udtr-negHeltal))$

nPr(*Udtr*, *posHeltal*) ⇒ $Udtr \cdot (Udtr-1) \dots (Udtr-posHeltal+1)$

nPr(*Udtr*, *ikkeHetal*) ⇒ $Udtr! / (Udtr-ikkeHetal)!$

nPr(*Liste1*, *Liste2*) ⇒ *liste*

Returnerer en liste med permutationer baseret på de sammenhørende elementpar i de to lister. Argumenterne skal være lister af samme størrelse.

$nPr(z, 3)$	$z \cdot (z-2) \cdot (z-1)$
-------------	-----------------------------

<i>Ans</i> $z=5$	60
--------------------	----

$nPr(z, -3)$	$\frac{1}{(z+1) \cdot (z+2) \cdot (z+3)}$
--------------	---

$nPr(z, c)$	$\frac{z!}{(z-c)!}$
-------------	---------------------

<i>Ans</i> · $nPr(z-c, -c)$	1
-----------------------------	---

$nPr(\{5, 4, 3\}, \{2, 4, 2\})$	$\{20, 24, 6\}$
---------------------------------	-----------------

nPr()Katalog > **nPr**(*Matrix1*, *Matrix2*) $\Rightarrow$ *matrix*

$$\text{nPr}\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right) = \begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}$$

Returnerer en matrix med permutationer baseret på de sammenhørende elementpar i de to matricer. Argumenterne skal være matricer af samme størrelse.

npv()Katalog > **npv**(*RenteSats*, *CFO*, *CFListe* [, *CFFrekv*])

$$\begin{aligned} \text{list1} &:= \{6000, -8000, 2000, -3000\} \\ &\quad \{6000, -8000, 2000, -3000\} \\ \text{list2} &:= \{2, 2, 2, 1\} \quad \{2, 2, 2, 1\} \\ \text{npv}(10, 5000, \text{list1}, \text{list2}) &= 4769.91 \end{aligned}$$

Finansfunktion, der beregner nettonutidsværdien. Summen af de aktuelle værdier for indkommende og udgående pengestrømme. Et positivt resultat for npv indikerer en profitabel investering.

RenteSats er renten, som pengestrømmen skal reduceres med over en periode (pengenes pris).

CFO er startpengestrømmen på tidspunkt 0. Den skal være et reelt tal.

CFListe er en liste over pengestrømsbeløb efter startpengestrømmen *CFO*.

CFFrekv er en liste, hvor hvert element angiver hyppighedsfrekvensen for et grupperet (fortløbende) pengestrømsbeløb, som er det tilsvarende element i *CFListe*. Standardværdien er 1. Hvis du indtaster værdier, skal de være positive heltal < 10.000.

nSolve()Katalog > **nSolve**(*Ligning*, *Var*[=*Gæt*]) $\Rightarrow$ *tal* eller *fejlstreng*

$$\begin{aligned} \text{nSolve}(x^2 + 5 \cdot x - 25 = 9, x) &= 3.84429 \\ \text{nSolve}(x^2 = 4, x = -1) &= -2. \\ \text{nSolve}(x^2 = 4, x = 1) &= 2. \end{aligned}$$

nSolve(*Ligning*, *Var* [*=Gæt*], *nedreGrænse*) $\Rightarrow$ *tal* eller *fejlstreng*

nSolve(*Ligning*, *Var*
[=*Guess*], *nedreGrænse*, *øvreGrænse*)
 $\Rightarrow$ *tal* eller *fejlstreng*

nSolve(*Ligning*, *Var*[=*Guess*]) |
nedreGrænse $\leq$ *Var* $\leq$ *øvreGrænse* $\Rightarrow$ *tal*
eller *fejlstreng*

Søger iterativt efter en approksimeret
reel numerisk løsning af *Ligning* for
dens ene variabel. Angiv variabelen som:

variabel

– eller –

variabel = *reelt tal*

For eksempel er x gyldig, og det er x=3
også.

nSolve() er ofte meget hurtigere end
solve() eller **zeros()**, især hvis “|”
operatoren anvendes til at begrænse
søgningen til et lille interval, der
indeholder nøjagtig én simpel løsning.

nSolve() forsøger at bestemme enten et
punkt, hvor residualen er nul, eller to
forholdsvis tætte punkter, hvor
residualerne har modsatte fortegn, og
residualen ikke er for stor. Hvis dette
ikke kan opnås med et beskedent antal
datapunkter, returneres strengen “Ingen
løsning blev fundet.”

Bemærk: Se også **cSolve()**, **cZeros()**, **solve**
() og **zeros()**.

Bemærk: Hvis der er flere løsninger, kan du
anvende et gæt til at finde en partikulær
løsning.

$\text{nSolve}(x^2 + 5 \cdot x - 25 = 9, x) x < 0$	-8.84429
$\text{nSolve}\left(\frac{(1+r)^{24}-1}{r} = 26, r\right) r > 0 \text{ and } r < 0.25$	0.006886
$\text{nSolve}(x^2 = 1, x)$	"No solution found"

O

OneVar

OneVar [*1*], *X*[, [*Hyppighed*]
[, [*Kategori*, *Medtag*]]]

OneVar [*n*], *X*1, *X*2[, [*X*3[, ..., [*X*20]]]]

Beregner statistik med en variabel på op til
20 lister. En sammenfatning af resultaterne
lagres i variabelen *stat.results*. (side 187.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

X argumenterne er datalister.

Frekv er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende X værdi. Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

Kategori er en liste med numeriske kategorikoder for tilsvarende X værdier.

Medtag er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Et tomt (ugyldigt) element i en af listerne X , *Freq* eller *Category* resulterer i at tilsvarende element i alle disse lister bliver ugyldigt. Et tomt element i en af listerne $X1$ til $X20$ resulterer i at tilsvarende element i alle disse lister bliver ugyldigt. Yderligere oplysninger om tomme elementer findes på side 266

Output-variabel	Beskrivelse
stat. $\bar{x}$	Gennemsnit af x -værdier
stat. Σx	Summen af x -værdier
stat. Σx^2	Summen af x^2 værdier
stat.sx	Standardafvigelse for målingen for x
stat. x	Populations standardafvigelse for x
stat.n	Antal datapunkter
stat.MinX	Minimum af x -værdier
stat.Q ₁ X	1. kvartil af x
stat.MedianX	Median af x .
stat.Q ₃ X	3. kvartil af x .
stat.MaxX	Maksimum af x -værdier.
stat.SSX	Summen af kvadraterne på afvigelser fra middelværdien for x .

BoolskUdtryk1 **or** *BoolskUdtryk2*
returnerer *boolsk udtryk*

$x \geq 3$ or $x \geq 4$	$x \geq 3$
--------------------------	------------

BoolskListe1 **or** *BoolskListe2* returnerer
Boolsk liste

Define $g(x) = \text{Func}$	Done
If $x \leq 0$ or $x \geq 5$	
Goto <i>end</i>	
Return $x \cdot 3$	
Lbl <i>end</i>	
EndFunc	

BoolskMatrix1 **or** *BoolskMatrix2*
returnerer *Boolsk matrix*

Returnerer true eller false eller en forenklet form af den oprindelige indtastning.

$g(3)$	9
$g(0)$	A function did not return a value

Returnerer true, hvis enten et eller begge udtryk kan reduceres til true. Returnerer kun false, hvis begge udtryk evalueres til false.

Bemærk: Se *xor*.

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

Heltal1 **or** *Heltal2* $\Rightarrow$ *heltal*

Sammenligner to reelle heltal bit for bit med en or-operation. Internt konverteres begge heltal til 64-bit binære tal med fortegn. Når de tilsvarede bits sammenlignes, er resultatet 1, hvis en af bittene er 1. Resultatet er kun 0, hvis begge bits er 0. Den returnerede værdi repræsenterer bit-resultaterne og vises i overensstemmelse med den valgte talsystemstilstand.

I hexadecimal tilstand:

0h7AC36 or 0h3D5F	0h7BD7F
-------------------	---------

Vigtigt: Tallet nul, ikke bogstavet O.

I binær tilstand:

0b100101 or 0b100	0b100101
-------------------	----------

Bemærk: En binær indtastning kan have op til 64 cifre (præfikset 0b ikke medregnet). En hexadecimal indtastning kan have op til 16 cifre.

Du kan indtaste heltallene i ethvert talsystem. Til binære eller hexadecimal indtastninger skal du som præfiks benytte henholdsvis 0b eller 0h. Uden præfiks behandles heltallene som decimaltal (10 talssystem).

Hvis du indtaster et decimalt heltal, der er for stort til en 64-bit binær form med fortegn, anvendes en symmetrisk modulo-operation til at bringe værdien ind i det korrekte område. Yderligere oplysninger findes under ►Base2, side 18.

Bemærk: Se xor.

ord()

Katalog > 

ord(*Streng*)⇒*heltal*

ord("hello")	104
--------------	-----

ord(*Liste1*)⇒*liste*

char(104)	"h"
-----------	-----

ord(char(24))	24
---------------	----

Returnerer den numeriske kode til første tegn i tegnstrengen *Streng*, eller en liste med de første tegn i hvert listeelement.

ord({"alpha", "beta"})	{ 97, 98 }
------------------------	------------

P

P►Rx()

Katalog > 

P►Rx(*rUdtr*, *θUdtr*)⇒*udtryk*

I vinkeltilstanden Radian:

P►Rx(*rListe*, *θListe*)⇒*liste*

$P \blacktriangleright Rx(r, \theta)$	$\cos(\theta) \cdot r$
---------------------------------------	------------------------

P►Rx(*rMatrix*, *θMatrix*)⇒*matrix*

$P \blacktriangleright Rx(4, 60^\circ)$	2
---	---

Returnerer den ækvivalente x-koordinat til parret (*r*, *θ*).

$P \blacktriangleright Rx\left(\{-3, 10, 1.3\}, \left\{\frac{\pi}{3}, \frac{-\pi}{4}, 0\right\}\right)$	$\left\{\frac{-3}{2}, 5, \sqrt{2}, 1.3\right\}$
---	---

Bemærk: Argumentet *θ* tolkes i grader, nygrader eller radianer afhængigt af den aktuelle vinkeltilstand. Hvis argumentet er et udtryk, kan du anvende °, ^G eller ^r til midlertidigt at tilsidesætte den indstillede vinkeltilstand.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **P@>Rx (...)**.

P►Ry()

Katalog > 

P►Ry(*rUdtr*, *θUdtr*)⇒*udtryk*

I vinkeltilstanden Radian:

P►Ry(*rListe*, *θListe*)⇒*liste*

P►Ry(*rMatrix*, *θMatrix*)⇒*matrix*

Returnerer den ækvivalente y-koordinat til parret (*r*, *θ*).

Bemærk: Argumentet *θ* tolkes i grader, nygrader eller radianer afhængigt af den aktuelle vinkeltilstand. Hvis argumentet er et udtryk, kan du anvende °, [°] eller ^r til midlertidigt at tilsidesætte den indstillede vinkeltilstand.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **P@>Ry (...)**.

P►Ry (<i>r</i> , <i>θ</i>)	$\sin(\theta) \cdot r$
P►Ry (4,60°)	$2 \cdot \sqrt{3}$
P►Ry ($\left\{-3, 10, 1.3\right\}, \left\{\frac{\pi}{3}, \frac{-\pi}{4}, 0\right\}$)	$\left\{\frac{-3 \cdot \sqrt{3}}{2}, -5 \cdot \sqrt{2}, 0\right\}$

PassErr

PassErr

Se et eksempel på **PassErr** i Eksempel2 under **Try**-kommandoen, side 202.

Videresender en fejl til næste niveau.

Hvis systemvariabel *errCode* er nul, gør **PassErr** ingenting.

Else betingelsen i **Try...Else...EndTry**-blokken bør anvende **ClrErr** eller **PassErr**. Brug **ClrErr**, hvis fejlen skal behandles eller ignoreres. Brug **PassErr**, hvis det ikke er kendt, hvad der skal gøres ved fejlen, for at sende den til den næste fejlhåndtering. Hvis der ikke er flere ventende **Try...Else...EndTry**-fejlhåndteringer, vises fejldialogboksen som normalt.

Bemærk: Se også **ClrErr**, side 27, og **Try**, side 202.

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

piecewise()

[Katalog >](#) 

piecewise(*Udtr1* [, *Betingelse1* [, *Udtr2* [, *Betingelse2* [, ...]]]])

Returnerer definitioner for en stykkevis funktion i form af en liste. Du kan også oprette stykkevise definitioner ved hjælp af en skabelon.

Bemærk: Se også **Stykkevis skabelon**, side 3.

Define $p(x) = \begin{cases} x, & x > 0 \\ \text{undef}, & x \leq 0 \end{cases}$	Done
$p(1)$	1
$p(-1)$	undef

poissCdf()

[Katalog >](#) 

poissCdf(λ , *nedreGrænse*, *øvreGrænse*) $\Rightarrow$ *tal*
hvis *nedreGrænse* og *øvreGrænse* er tal,
liste hvis *nedreGrænse* og *øvreGrænse* er lister

poissCdf(λ , *øvreGrænse*) (for $P(0 \leq X \leq \text{øvreGrænse}) \Rightarrow \text{tal}$ hvis *øvreGrænse* er et tal, *liste* hvis *øvreGrænse* er en liste

Beregner den kumulerede sandsynlighed for den diskrete Poisson-distribution med en angivet middelværdi λ .

For $P(X \leq \text{øvreGrænse})$, sæt *nedreGrænse*=0

poissPdf()

[Katalog >](#) 

poissPdf(λ , *XVærdi*) $\Rightarrow$ *tal* hvis *XVærdi* er et tal, *liste* hvis *XVærdi* er en liste

Beregner en sandsynlighed for den diskrete Poisson-distribution med den angivne middelværdi λ .

►Polar

[Katalog >](#) 

Vektor ►**Polar**

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>**Polar**.

Viser *vektor* i polær form [$r \angle \theta$].

Vektoren skal være af dimensionen 2 og kan være en række eller kolonne.

$\begin{bmatrix} 1 & 3 \end{bmatrix}$ ►Polar	$\begin{bmatrix} 3.16228 & \angle 1.24905 \end{bmatrix}$
$\begin{bmatrix} x & y \end{bmatrix}$ ►Polar	$\begin{bmatrix} \sqrt{x^2 + y^2} & \angle \frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1}\left(\frac{x}{y}\right) \end{bmatrix}$

Bemærk: ►Polar er en displayformat-kommando, ikke en konverteringsfunktion. Du kan kun anvende den i slutningen af en indtastningslinje, og den opdaterer ikke *ans*.

Bemærk: Se også ►Rect, side 154.

kompleksVærdi ►Polar

Viser *kompleksVærdi* i polær form.

- Vinkeltilstanden Grader returnerer $(r \angle \theta)$.
- Vinkeltilstanden Radian returnerer $re^{i\theta}$.

kompleksVærdi kan have enhver kompleks form. Men en $re^{i\theta}$ -indtastning udløser en fejl i vinkeltilstanden Grader.

Bemærk: Du skal anvende parenteser til en $(r \angle \theta)$ polær indtastning.

I vinkeltilstanden Radian:

$$\begin{array}{l} (3+4 \cdot i) \text{ ►Polar} \\ e^{i \cdot \left(\frac{\pi}{2} - \tan^{-1} \left(\frac{3}{4} \right) \right) \cdot .5} \\ \left(4 \angle \frac{\pi}{3} \right) \text{ ►Polar} \\ e^{\frac{i \cdot \pi}{3} \cdot .4} \end{array}$$

I vinkeltilstanden Nygrader:

$$(4 \cdot i) \text{ ►Polar} \quad (4 \angle 100.)$$

I vinkeltilstanden Grader:

$$(3+4 \cdot i) \text{ ►Polar} \quad \left(5 \angle 90 - \tan^{-1} \left(\frac{3}{4} \right) \right)$$

polyCoeffs()

polyCoeffs(Poly [,Var])⇒*liste*

Returnerer en liste over koefficienterne i polynomiet *Poly* vedrørende variabelen *Var*.

Poly skal være et polynomielt udtryk i *Var*. Vi anbefaler, at du ikke udelader *Var*, med mindre *Poly* er et udtryk i en enkelt variabel.

$$\text{polyCoeffs}(4 \cdot x^2 - 3 \cdot x + 2, x) \quad \{4, -3, 2\}$$

$$\text{polyCoeffs}((x-1)^2 \cdot (x+2)^3) \quad \{1, 4, 1, -10, -4, 8\}$$

Udvider polynomiet og vælger *x* for den udeladte *Var*.

$\text{polyCoeffs}\left((x+y+z)^2, x\right)$	$\{1, 2 \cdot (y+z), (y+z)^2\}$
$\text{polyCoeffs}\left((x+y+z)^2, y\right)$	$\{1, 2 \cdot (x+z), (x+z)^2\}$
$\text{polyCoeffs}\left((x+y+z)^2, z\right)$	$\{1, 2 \cdot (x+y), (x+y)^2\}$

polyDegree()

polyDegree(*Poly* [, *Var*]) $\Rightarrow$ værdi

Returnerer graden af det polynomielle udtryk *Poly* med hensyn til variabelen *Var*. Hvis du udelader *Var*, vælger

polyDegree()-funktionen en standardværdi fra variablene i polynomiet *Poly*.

Poly skal være et polynomielt udtryk i *Var*. Vi anbefaler, at du ikke udelader *Var*, med mindre *Poly* er et udtryk i en enkelt variabel.

$\text{polyDegree}(5)$	0
$\text{polyDegree}(\ln(2) + \pi, x)$	0

Konstante polynomier

$\text{polyDegree}(4 \cdot x^2 - 3 \cdot x + 2, x)$	2
$\text{polyDegree}((x-1)^2 \cdot (x+2)^3)$	5

$\text{polyDegree}((x+y^2+z^3)^2, x)$	2
$\text{polyDegree}((x+y^2+z^3)^2, y)$	4
$\text{polyDegree}((x-1)^{10000}, x)$	10000

Graden kan også udledes, selvom koefficienterne ikke kan. Det skyldes, at graden kan udtrækkes uden at ekspandere polynomiet.

polyEval()

polyEval(*Liste1*, *Udtr1*) $\Rightarrow$ udtryk

polyEval(*Liste1*, *Liste2*) $\Rightarrow$ udtryk

$\text{polyEval}(\{a, b, c\}, x)$	$a \cdot x^2 + b \cdot x + c$
$\text{polyEval}(\{1, 2, 3, 4\}, 2)$	26
$\text{polyEval}(\{1, 2, 3, 4\}, \{2, -7\})$	$\{26, -262\}$

polyEval()Katalog > 

Fortolker første argument som koefficienten til et polynomium i faldende grad og returnerer polynomiet beregnet for værdien af det andet argument.

polyGcd()Katalog > 

polyGcd(*Udtryk1*, *Udtryk2*) $\Rightarrow$ *udtryk*

Returnerer den største fælles divisor af de to argumenter.

Udtryk1 og *Udtryk2* skal være polynomielle udtryk.

Liste-, matrix-, og Boolske argumenter er ikke tilladt.

$\text{polyGcd}(100, 30)$	10
$\text{polyGcd}(x^2 - 1, x - 1)$	$x - 1$
$\text{polyGcd}(x^3 - 6 \cdot x^2 + 11 \cdot x - 6, x^2 - 6 \cdot x + 8)$	$x - 2$

polyQuotient()Katalog > 

polyQuotient(*Poly1*, *Poly2* [, *Var*]) $\Rightarrow$ *udtryk*

Returnerer kvotienten af polynomiet *Poly1* divideret med polynomiet *Poly2* med hensyn til den angivne variabel *Var*.

Poly1 og *Poly2* skal være polynomielle udtryk i *Var*. Vi anbefaler, at du ikke udelader *Var*, med mindre *Poly1* og *Poly2* er udtryk i den samme variabel.

$\text{polyQuotient}(x - 1, x - 3)$	1
$\text{polyQuotient}(x - 1, x^2 - 1)$	0
$\text{polyQuotient}(x^2 - 1, x - 1)$	$x + 1$
$\text{polyQuotient}(x^3 - 6 \cdot x^2 + 11 \cdot x - 6, x^2 - 6 \cdot x + 8)$	x
$\text{polyQuotient}((x - y) \cdot (y - z), x + y + z, x)$	$y - z$
$\text{polyQuotient}((x - y) \cdot (y - z), x + y + z, y)$	$2 \cdot x - y + 2 \cdot z$
$\text{polyQuotient}((x - y) \cdot (y - z), x + y + z, z)$	$-(x - y)$

polyRemainder()Katalog > 

polyRemainder(*Poly1*, *Poly2* [, *Var*]) $\Rightarrow$ *udtryk*

$\text{polyRemainder}(x - 1, x - 3)$	2
$\text{polyRemainder}(x - 1, x^2 - 1)$	$x - 1$
$\text{polyRemainder}(x^2 - 1, x - 1)$	0

polyRemainder()

Katalog > 

Returnerer resten af polynomiet *Poly1* divideret med polynomiet *Poly2* med hensyn til den angivne variabel *Var*.

Poly1 og *Poly2* skal være polynomielle udtryk i *Var*. Vi anbefaler, at du ikke udelader *Var*, med mindre *Poly1* og *Poly2* er udtryk i den samme enkelte variabel.

$$\begin{array}{l} \text{polyRemainder}((x-y) \cdot (y-z), x+y+z, x) \\ \quad \quad \quad -(y-z) \cdot (2 \cdot y+z) \\ \hline \text{polyRemainder}((x-y) \cdot (y-z), x+y+z, y) \\ \quad \quad \quad -2 \cdot x^2 - 5 \cdot x \cdot z - 2 \cdot z^2 \\ \hline \text{polyRemainder}((x-y) \cdot (y-z), x+y+z, z) \\ \quad \quad \quad (x-y) \cdot (x+2 \cdot y) \end{array}$$

polyRoots()

Katalog > 

polyRoots(*Poly*, *Var*) $\Rightarrow$

polyRoots(*ListeAfKoeff*) $\Rightarrow$

Den første syntaks, **polyRoots(*Poly*, *Var*)**, returnerer en liste af reelle rødder af polynomiet *Poly* med hensyn til variabelen *Var*. Hvis der ikke findes en reel rod, returneres en tom liste: { }.

Poly skal være et polynomium i en variabel.

Den anden syntaks, **polyRoots(*ListeAfKoeff*)**, returnerer en liste med reelle rødder for koefficienterne i *ListeAfKoeff*.

Bemærk: Se også **cPolyRoots()**, side 38.

$$\begin{array}{l} \text{polyRoots}(y^3+1, y) \quad \quad \quad \{-1\} \\ \hline \text{cPolyRoots}(y^3+1, y) \\ \quad \quad \quad \left\{ -1, \frac{1}{2} - \frac{\sqrt{3}}{2}i, \frac{1}{2} + \frac{\sqrt{3}}{2}i \right\} \\ \hline \text{polyRoots}(x^2+2 \cdot x+1, x) \quad \quad \quad \{-1, -1\} \\ \hline \text{polyRoots}(\{1, 2, 1\}) \quad \quad \quad \{-1, -1\} \end{array}$$

PowerReg

Katalog > 

PowerReg *X*, *Y* [, *Frekv*] [, *Kategori*, *Medtag*]

Beregner den lineære regression $y = (a \cdot (x)^b)$ på liste *X* og *Y* med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

X og *Y* er lister med uafhængige og afhængige variable.

Frekv er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hvert tilsvarende *X* og *Y* datapunkt. Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

Kategori er en liste, der indeholder kategorikoder for *X* og *Y* data..

Medtag er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot (x)^b$
stat.a, stat.b	Regressionskoefficienter
stat.r ²	Koefficient af en lineær forklaringsgrad til transformerede data
stat.r	Korrelationskoefficient til transformerede data ($\ln(x)$, $\ln(y)$)
stat.Resid	Residualer forbundet med eksponentielmodellen
stat.ResidTrans	Residualer associeret med lineær tilpasning af transformerede data
stat.XReg	Liste af datapunkter i den modificerede <i>X-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

Prgm*Blok***EndPrgm**

Beregn GCD og vis mellemresultater.

Skabelon til oprettelse af et brugerdefineret program. Skal anvendes sammen med **Define**, **Define LibPub**, eller **Define LibPriv**-kommandoen.

Blok kan være en enkelt sætning, en række sætninger adskilt med kolon eller en række sætninger på separate linjer.

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

```
Define proggcd(a,b)=Prgm
  Local d
  While b≠0
    d:=mod(a,b)
    a:=b
    b:=d
  Disp a," ",b
  EndWhile
  Disp "GCD=",a
  EndPrgm
```

Done

proggcd(4560,450)

450 60

60 30

30 0

GCD=30

Done

prodSeq()

Se $\Pi()$, side 237.

Product (PI)

Se $\Pi()$, side 237.

product()

product(Liste[, Start[, slut]]) $\Rightarrow$ *udtryk*

Returnerer produktet af elementerne indeholdt i *Liste*. *Start* og *Slut* er valgfri. De angiver en serie af elementer.

product(Matrix1[, Start[, slut]]) $\Rightarrow$ *matrix*

Returnerer en rækkevektor med produkterne af elementerne i kolonnerne i *Matrix1*. *Start* og *slut* er valgfri. De angiver en serie af rækker.

product({1,2,3,4}) 24

product({2,x,y}) $2 \cdot x \cdot y$

product({4,5,8,9},2,3) 40

product($\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$) $\begin{bmatrix} 28 & 80 & 162 \end{bmatrix}$

product($\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$,1,2) $\begin{bmatrix} 4 & 10 & 18 \end{bmatrix}$

Tomme (ugyldige) elementer ignoreres.
Yderligere oplysninger om tomme elementer findes på side 266.

propFrac()

propFrac(Udtr1[, Var]) ⇒ *udtryk*

propFrac(rationalt_tal) returnerer *rationalt_tal* som summen af et heltal og en brøk med samme fortegn og en større nævner end tæller.

propFrac(rationalt_Udtryk, Var) returnerer summen af ægte brøker og et polynomium med hensyn til *Var*. Graden af *Var* i nævneren overstiger graden af *Var* i tælleren i hver enkelt ægte brøk. Ens potenser af *Var* samles. Leddene og deres faktorer sorteres med *Var* som hovedvariabel.

Hvis *Var* udelades, foretages en udvikling i ægte brøker med hensyn til den hyppigst forekommende variabel. Koefficienterne af polynomiumdelen gøres derefter ægte med hensyn til deres hyppigst forekommende variabel osv.

For rationale udtryk er **propFrac()** et hurtigere men mindre vidtgående alternativ til **expand()**.

$$\text{propFrac}\left(\frac{4}{3}\right) \quad 1 + \frac{1}{3}$$

$$\text{propFrac}\left(\frac{-4}{3}\right) \quad -1 - \frac{1}{3}$$

$$\text{propFrac}\left(\frac{x^2+x+1}{x+1} + \frac{y^2+y+1}{y+1}, x\right)$$

$$\frac{1}{x+1} + x + \frac{y^2+y+1}{y+1}$$

$$\text{propFrac}(\text{Ans}) \quad \frac{1}{x+1} + x + \frac{1}{y+1} + y$$

Q

QR

QR Matrix, qmatNavn, rmatNavn[, Tol]

Beregner Householder QR faktoropløsningen af en reel eller kompleks matrix. De resulterende Q og R-matricer lagres i de angivne *matNavne*. Q matrix er unitær. R matrix er øvre-triangular.

Tallet med flydende komma (9). i m1 får resultaterne beregnet med flydende komma.

Ethvert matricelement kan valgfrit behandles som nul, hvis dets absolutte værdi er mindre end *Tol*. Denne tolerance anvendes kun, hvis matricen har elementer med flydende decimaler og ikke indeholder symbolske variable, der ikke er tildelt en værdi. Ellers ignoreres *Tol*.

- Hvis du anvender   eller indstiller **Auto eller tilnærmet** - tilstanden til Approks, foretages beregningerne med aritmetik med flydende komma.
- Hvis *Tol* udelades eller ikke anvendes, beregnes standardtolerancen som:
 $5E-14 \cdot \max(\dim(Matrix)) \cdot \text{rowNorm}(Matrix)$

QR faktoropløsningen beregnes numerisk med Householder-transformationer. Den symbolske løsning beregnes med Gram-Schmidt. Kolonnerne i *qmatNavn* er ortonormale vektorer, der udspænder rummet defineret ved *matrix*.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
QR <i>m1,qm,rm</i> Done	
<i>qm</i>	$\begin{bmatrix} 0.123091 & 0.904534 & 0.408248 \\ 0.492366 & 0.301511 & -0.816497 \\ 0.86164 & -0.301511 & 0.408248 \end{bmatrix}$
<i>rm</i>	$\begin{bmatrix} 8.12404 & 9.60114 & 11.0782 \\ 0. & 0.904534 & 1.80907 \\ 0. & 0. & 0. \end{bmatrix}$

$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} m & n \\ o & p \end{bmatrix}$
QR <i>m1,qm,rm</i> Done	
<i>qm</i>	$\begin{bmatrix} \frac{m}{\sqrt{m^2+o^2}} & \frac{-\text{sign}(m \cdot p - n \cdot o) \cdot o}{\sqrt{m^2+o^2}} \\ \frac{o}{\sqrt{m^2+o^2}} & \frac{m \cdot \text{sign}(m \cdot p - n \cdot o)}{\sqrt{m^2+o^2}} \end{bmatrix}$
<i>rm</i>	$\begin{bmatrix} \sqrt{m^2+o^2} & \frac{m \cdot n + o \cdot p}{\sqrt{m^2+o^2}} \\ 0 & \frac{m \cdot p - n \cdot o}{\sqrt{m^2+o^2}} \end{bmatrix}$

QuadReg

QuadReg *X, Y [, Frekv] [, Kategori, Medtag]*

Beregner andengrads polynomiel regression $y = a \cdot x^2 + b \cdot x + c$ på listerne *X* og *Y* med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

X og Y er lister med uafhængige og afhængige variable.

Frekv er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver frekvensen af hændelse for hver tilsvarende X og Y datapunkt. Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

Kategori er en liste, der indeholder kategorikoder for X og Y data..

Medtag er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot x^2 + b \cdot x + c$
stat.a, stat.b, stat.c	Regressionskoefficienter
stat.r ²	Forklaringsgraden
stat.Resid	Residualer fra regressionen
stat.XReg	Liste af datapunkter i den modificerede X -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede Y -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

QuartReg $X, Y [, Frekv] [, Kategori, Medtag]$

Beregner den polynomielle tredjegradsregression

$y = a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$ på listerne X og Y med frekvens *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

X og Y er lister med uafhængige og afhængige variable.

Frekv er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver frekvensen af hændelse for hver tilsvarende X og Y datapunkt. Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

Kategori er en liste, der indeholder kategorikoder for X og Y data..

Medtag er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$
stat.a, stat.b, stat.c, stat.d, stat.e	Regressionskoefficienter
stat.r ²	Forklaringsgraden
stat.Resid	Residualer fra regressionen
stat.XReg	Liste af datapunkter i den modificerede <i>X</i> -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategori</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y</i> -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategori</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

R ► P0()

Katalog >

R ► P0 (xExpr, yExpr) ⇒ *udtryk*R ► P0 (xList, yList) ⇒ *liste*R ► P0 (xMatrix, yMatrix) ⇒ *matrix*

Returnerer den ækvivalente θ -koordinat for (x,y) argumentparret.

Bemærk: Resultatet returneres som en vinkel i grader, nygrader eller radian afhængig af den aktuelle vinkeltilstand.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **R@>Ptheta (...)**.

I vinkeltilstanden Grader:

$$R \blacktriangleright P0(x,y) \quad 90 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

I vinkeltilstanden Nygrader:

$$R \blacktriangleright P0(x,y) \quad 100 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

I vinkeltilstanden Radian:

$$R \blacktriangleright P0(3,2) \quad \tan^{-1}\left(\frac{2}{3}\right)$$

$$R \blacktriangleright P0\left([3 \ -4 \ 2], \left[0 \ \frac{\pi}{4} \ 1.5\right]\right)$$

$$\left[0 \ \tan^{-1}\left(\frac{16}{\pi}\right) + \frac{\pi}{2} \ 0.643501\right]$$

R ► Pr()

Katalog >

R ► Pr (xExpr, yExpr) ⇒ *udtryk*R ► Pr (xList, yList) ⇒ *liste*R ► Pr (xMatrix, yMatrix) ⇒ *matrix*

Returnerer den ækvivalente r -koordinat for (x,y) argumentparret.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **R@>Pr (...)**.

I vinkeltilstanden Radian:

$$R \blacktriangleright Pr(3,2) \quad \sqrt{13}$$

$$R \blacktriangleright Pr(x,y) \quad \sqrt{x^2 + y^2}$$

$$R \blacktriangleright Pr\left([3 \ -4 \ 2], \left[0 \ \frac{\pi}{4} \ 1.5\right]\right)$$

$$\left[3 \ \frac{\sqrt{\pi^2 + 256}}{4} \ 2.5\right]$$

► Rad

Katalog >

Expr1 ► Rad ⇒ *udtryk*

Konverterer argumentet til vinkelmåling i radian.

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive **@>Rad**.

I vinkeltilstanden grader:

$$(1.5) \blacktriangleright \text{Rad} \quad (0.02618)^r$$

I vinkeltilstanden Nygrader:

$$(1.5) \blacktriangleright \text{Rad} \quad (0.023562)^r$$

rand()

Katalog > 

rand() $\Rightarrow$ *udtryk*

rand(#Trials) $\Rightarrow$ *liste*

rand() returnerer en vilkårlig værdi mellem 0 og 1.

rand(#Trials) returnerer en liste med #Trials vilkårlig værdier mellem 0 og 1.

Angiv seed-værdien for et vilkårligt tal.

RandSeed 1147	Done
rand(2)	{ 0.158206, 0.717917 }

randBin()

Katalog > 

randBin(*n*, *p*) $\Rightarrow$ *udtryk*

randBin(*n*, *p*, #Trials) $\Rightarrow$ *liste*

randBin(*n*, *p*) returnerer et vilkårlig reelt tal fra en given binomialfordeling.

randBin(*n*, *p*, #Trials) returnerer en liste med #Trials vilkårlige reelle tal fra en given binomialfordeling.

randBin(80,0.5)	42
randBin(80,0.5,3)	{ 41, 32, 39 }

randInt()

Katalog > 

randInt

(*lowBound*, *upBound*) $\Rightarrow$ *udtryk*

randInt

(*lowBound*, *upBound*, #Trials) $\Rightarrow$ *liste*

randInt

(*lowBound*, *upBound*) returnerer et

vilkårligt heltal i det område, der angives af heltalsgrænserne *lowBound* og *upBound*.

randInt

(*lowBound*, *upBound*, #Trials) returnerer en liste med #Trials vilkårlige heltal i det angivne område.

randInt(3,10)	5
randInt(3,10,4)	{ 9, 7, 5, 8 }

randMat()Katalog > **randMat**(*numRows*, *numColumns*) ⇒ *matrice*

Returnerer en matrix med heltal mellem -9 og 9 af den angivne dimension.

Begge argumenter skal kunne reduceres til heltal.

RandSeed 1147	Done
randMat (3,3)	$\begin{bmatrix} 8 & -3 & 6 \\ -2 & 3 & -6 \\ 0 & 4 & -6 \end{bmatrix}$

Bemærk: Værdierne i denne matrix ændres, hver gang du trykker **enter**.

randNorm()Katalog > **randNorm**(μ , σ) ⇒ *udtryk***randNorm**(μ , σ , *#Trials*) ⇒ *liste*

randNorm(μ , σ) returnerer et decimalt tal fra den angivne normalfordeling. Det kan være ethvert reelt tal, men vil være kraftigt koncentreret i intervallet [$\mu-3\cdot\sigma$, $\mu+3\cdot\sigma$].

randNorm(μ , σ , *#Trials*) returnerer en liste med *#Trials* decimaltal fra den angivne normalfordeling.

RandSeed 1147	Done
randNorm (0,1)	0.492541
randNorm (3,4.5)	-3.54356

randPoly()Katalog > **randPoly**(*Var*, *Order*) ⇒ *udtryk*

Returnerer et polynomium i *Var* af den angivne *Order*. Koefficienterne er vilkårlige heltal i området -9 til 9. Koefficienten af højeste grad vil ikke være nul.

Order skal være 0–99.

RandSeed 1147	Done
randPoly (<i>x</i> ,5)	$-2\cdot x^5 + 3\cdot x^4 - 6\cdot x^3 + 4\cdot x - 6$

randSamp()Katalog > **randSamp**(*List*, *#Trials*[, *noRepl*]) ⇒ *liste*

Returnerer en liste med en vilkårlig stikprøve af *#Trials* målinger fra *List* med mulighed for tilbagelægning (*noRepl*=0) eller ingen tilbagelægning (*noRepl*=1). Standardindstillingen er med tilbagelægning.

Define <i>list3</i> = { 1,2,3,4,5 }	Done
Define <i>list4</i> = randSamp (<i>list3</i> ,6)	Done
<i>list4</i>	{ 2,3,4,3,1,2 }

RandSeed *Number*

Hvis *Number* = 0, indstilles startværdien til fabriksindstillingerne for vilkårlig-talgeneratoren. Hvis *Number* ≠ 0, anvendes det til at generere to startværdier, der lagres i systemvariablerne seed1 og seed2.

RandSeed 1147

Done

rand()

0.158206

real()**real**(*Expr1*) ⇒ *udtryk*

Returnerer den reelle del af argumentet.

Bemærk: Alle udefinerede variable behandles som reelle variable. Se også **imag()**, page 95.

real(*List1*) ⇒ *liste*

Returnerer realdelen af alle elementer.

real(*Matrix1*) ⇒ *matrix*

Returnerer realdelen af alle elementer.

 $\text{real}(2+3 \cdot i)$ 2 $\text{real}(z)$ z $\text{real}(x+i \cdot y)$ x $\text{real}(\{a+i \cdot b, 3, i\})$ $\{a, 3, 0\}$

$$\text{real}\left(\begin{bmatrix} a+i \cdot b & 3 \\ c & i \end{bmatrix}\right) \quad \begin{bmatrix} a & 3 \\ c & 0 \end{bmatrix}$$
► **Rect***Vektor* ► **Rect**

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>**Rect**.

Viser *Vektor* i rektangulær form [x, y, z]. Vektoren skal være af dimensionen 2 eller 3 og kan være en række eller kolonne.

Bemærk: ► **Rect** er en displayformatkommando, ikke en konverteringsfunktion. Du kan kun anvende den i slutningen af en indtastningslinje, og den opdaterer ikke *ans*.

Bemærk: Se også ► **Polær ligning**, side 140.

complexValue ► **Rect**

$$\left(3 \angle \frac{\pi}{4} \angle \frac{\pi}{6}\right) \text{►Rect}$$

$$\begin{bmatrix} 3 \cdot \sqrt{2} & 3 \cdot \sqrt{2} & 3 \cdot \sqrt{3} \\ 4 & 4 & 2 \end{bmatrix}$$

$$\begin{bmatrix} a \angle b \angle c \\ [a \cdot \cos(b) \cdot \sin(c) \quad a \cdot \sin(b) \cdot \sin(c) \quad a \cdot \cos(c)] \end{bmatrix}$$

I vinkeltilstanden Radian:

► Rect

Katalog >

Viser *complexValue* i rektangulær form $a+bi$. *complexValue* kan have enhver kompleks form. Men en $re^{i\theta}$ -indtastning udløser en fejl i vinkeltilstanden Grader.

Bemærk: Du skal anvende parenteser til en $(r \angle \theta)$ polær indtastning.

$$\left(4 \cdot e^{\frac{\pi}{3}}\right) \rightarrow \text{Rect} \quad 4 \cdot e^{\frac{\pi}{3}}$$

$$\left(\left(4 \angle \frac{\pi}{3}\right)\right) \rightarrow \text{Rect} \quad 2+2 \cdot \sqrt{3} \cdot i$$

I vinkeltilstanden Nygrader:

$$\left((1 \angle 100)\right) \rightarrow \text{Rect} \quad i$$

I vinkeltilstanden grader:

$$\left((4 \angle 60)\right) \rightarrow \text{Rect} \quad 2+2 \cdot \sqrt{3} \cdot i$$

Bemærk: Du kan skrive $\angle$ ved at vælge det i symbollisten i kataloget.

ref()

Katalog >

$\text{ref}(\text{MatrixI}, \text{Tol}) \Rightarrow \text{matrix}$

Returnerer række-echelonformen af *MatrixI*.

Ethvert matricelement kan valgfrit behandles som nul, hvis dets absolutte værdi er mindre end *Tol*. Denne tolerance anvendes kun, hvis matricen har elementer med flydende decimaler og ikke indeholder symbolske variable, der ikke har fået tildelt en værdi. Ellers ignoreres *Tol*.

- Hvis du anvender eller indstiller **Auto- eller Approximate**-tilstanden til Approximate, foretages beregningerne med flydende komma.
- Hvis *Tol* udelades eller ikke anvendes, beregnes standardtolerancen som:
 $5E-14 \cdot \max(\dim(\text{MatrixI})) \cdot \text{rowNorm}(\text{MatrixI})$

Undgå udefinerede elementer i *MatrixI*. De kan føre til uventede resultater.

$$\text{ref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right) \quad \begin{bmatrix} 1 & -\frac{2}{5} & -\frac{4}{5} & \frac{4}{5} \\ 0 & 1 & \frac{4}{7} & \frac{11}{7} \\ 0 & 0 & 1 & -\frac{62}{71} \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow mI \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\text{ref}(mI) \quad \begin{bmatrix} 1 & \frac{d}{c} \\ 0 & 1 \end{bmatrix}$$

Hvis for eksempel a er udefineret i følgende udtryk, vil en advarsel blive vist, og resultatet vises som:

$$\text{ref} \left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \quad \begin{bmatrix} 1 & \frac{1}{a} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Advarslen vises, fordi det generaliserede element $1/a$ ikke ville være gyldig for $a=0$.

Du kan undgå dette ved i forvejen at gemme en værdi til a eller ved at bruge betingelsesoperatoren ("|") til at substituere en værdi som vist i følgende eksempel.

$$\text{ref} \left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mid a=0 \right) \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Bemærk: Se også `rref()`, page 165.

RefreshProbeVars

RefreshProbeVars

Giver dig adgang til sensordata fra alle tilkoblede sensorprober i dit TI-Basic-program.

StatusVar-værdi

Status

`statusVar`
=0 Normal (fortsæt med programmet)
Vernier DataQuest™-applikation er i dataindsamlingsstilstand.

`statusVar`
=1 **Bemærk:** Vernier DataQuest™-applikation skal være i målerstilstand for at denne kommando virker. 

Eksempel

```
Define temp()=
Prgm
© Tjek, om systemet er klar
RefreshProbeVars-status
If status=0 Then
Disp "klar"
For n,1,50
RefreshProbeVars-status
temperatur:=meter.temperatur
Disp "Temperatur: ",temperatur
If temperature>30 Then
```


StatusVar-værdi	Status	
<i>statusVar</i> =2	Vernier DataQuest™-applikation blev ikke startet.	Disp "For varmt"
<i>statusVar</i> =3	Vernier DataQuest™-applikationen blev startet, men du har ikke tilkoblet nogen prober.	EndIf
		© Vent i 1 sekund mellem stikprøver
		Wait 1
		EndFor
		Else
		Disp "Ikke klar. Prøv igen senere"
		EndIf
		EndPrgm

Bemærk: Dette kan også bruge med TI-Innovator™-hubben.

remain()

remain(*Expr1*, *Expr2*) ⇒ *udtryk*
remain(*List1*, *List2*) ⇒ *liste*
remain(*Matrix1*, *Matrix2*) ⇒ *matrix*

Returnerer resten af det første argument med hensyn til det andet argument som defineret af identiteterne:

remain(*x*,0) *x*
 remain(*x*,*y*) $x - y \cdot \text{iPart}(x/y)$

Bemærk, at som følge heraf, **remain**($-x, y$) = **remain**(*x*,*y*). Resultatet er enten nul, eller det har samme fortegn som det første argument.

Bemærk: Se også **mod()**, side 123.

remain(7,0)	7
remain(7,3)	1
remain(-7,3)	-1
remain(7,-3)	1
remain(-7,-3)	-1
remain({12,-14,16},{9,7,-5})	{3,0,1}

remain($\begin{bmatrix} 9 & -7 \\ 6 & 4 \end{bmatrix}, \begin{bmatrix} 4 & 3 \\ 4 & -3 \end{bmatrix})$	$\begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}$
---	---

Request

Request *promptString*, *var*[, *DispFlag* [, *statusVar*]]

Request *promptString*, *func*(*arg1*, ...*argn*) [, *DispFlag* [, *statusVar*]]

Definer et program:

```
Define request_demo()=Prgm
  Request "Radius: ",r
  Disp "Areal = ",pi*r^2
EndPrgm
```

Programmeringskommando: Standser programmet midlertidigt og viser en dialogboks med meddelelsen *promptString* og et indtastningsfelt til brugerens svar.

Når brugeren skriver et svar og klikker på **OK**, tildeles indtastningsfeltets indhold til variabelen *var*.

Hvis brugeren klikker på **Annuller**, fortsætter programmet uden at acceptere noget input. Programmet bruger den tidligere værdi for *var* hvis *var* var defineret i forvejen.

Det valgfrie argument *DispFlag* kan være et hvilket som helst udtryk.

- Hvis *DispFlag* er udelades eller evalueres til **1**, vil klarmeddelelsen og brugerens svar blive vist i beregnerens historik.
- Hvis *DispFlag* evalueres til **0**, vil klarmeddelelsen og svaret ikke blive vist historikken.

Det valgfrie argument *statusVar* gør det muligt for programmet at bestemme, hvordan brugeren forlod dialogboksen. Bemærk, at *statusVar* kræver argumentet *DispFlag*.

- Hvis brugeren klikkede **OK** eller trykkede på **Enter** eller **Ctrl+Enter**, indstilles variabelen *statusVar* til en værdi på **1**.
- I modsat fald indstilles variabelen *statusVar* til en værdi på **0**.

Med argumentet *func()* kan et program lagre brugerens svar som en funktionsdefinition. Denne syntaks virker, ligesom hvis brugeren udførte kommandoen:

```
Define func(arg1, ...argn) = brugers svar
```

Start programmet, og indtast et svar:

```
request_demo()
```



Resultat efter valg af **OK**:

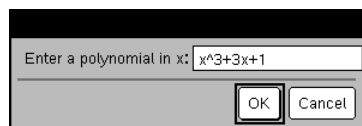
```
Radius: 6/2
Areal = 28,2743
```

Definer et program:

```
Define polynomial()=Prgm
  Request "Indtast et polynomium i
  x:",p(x)
  Disp "Reelle rødder er:",polyRoots
  (p(x),x)
EndPrgm
```

Start programmet, og indtast et svar:

```
polynomial()
```



Resultat efter indtastning af x^3+3x+1 og valg af **OK**:

```
Reelle rødder er: {-0.322185}
```

Så kan programmet bruge den definerede funktion *func()*.
promptString bør vejlede brugeren til at indtaste en passende *brugers svar*, der fuldender funktionsdefinitionen.

Bemærk: Du kan bruge kommandoen Request i et brugerdefineret program, men ikke i en funktion.

Sådan stoppes et program, der indeholder kommandoen **Request** i en uendelig løkke:

- **Håndholdt:** Hold tasten  nede, mens du gentagne gange trykker på .
- **Windows®:** Hold tasten **F12** nede, mens du gentagne gange trykker på **Enter**.
- **Macintosh®:** Hold tasten **F5** nede, mens du gentagne gange trykker på **Enter**.
- **iPad®:** App'en viser en meddelelse. Du kan fortsat vente eller annullere.

Bemærk: Se også **RequestStr**, page 159.

RequestStr

RequestStr *promptString*, *var*[, *DispFlag*]

Programmeringskommando: Fungerer identisk med den første syntaks i kommandoen **Request**, bortset fra at brugerens svar altid fortolkes som en streng. I modsætning hertil fortolker kommandoen **Request** svarene som et udtryk, medmindre brugeren omslutter det med citationstegn ("").

Bemærk: Du kan bruge kommandoen **RequestStr** i et brugerdefineret program, men ikke i en funktion.

Sådan standses et program, der indeholder en **RequestStr** kommando i en uendelig løkke:

Definer et program:

```
Define requestStr_demo()=Prgm
    RequestStr "Dit navn:",name,0
    Disp "Svaret har ",dim(name),"
    tegn."
EndPrgm
```

Start programmet, og indtast et svar:

requestStr_demo()



- **Håndholdt:** Hold tasten  nede, mens du gentagne gange trykker på .
- **Windows®:** Hold tasten **F12** nede, mens du gentagne gange trykker på **Enter**.
- **Macintosh®:** Hold tasten **F5** nede, mens du gentagne gange trykker på **Enter**.
- **iPad®:** App'en viser en meddelelse. Du kan fortsat vente eller annullere.

Resultat efter valg af **OK** (Bemærk, at argumentet *DispFlag* fra **0** udelader klarmeddelelsen og svaret fra historikken):

```
requestStr_demo()
```

Svaret har fem tegn.

Bemærk: Se også **Request**, page 157.

Return

Return [*Expr*]

Returnerer *Expr* som resultatet af funktionen. Bruges i en **Func...EndFunc**-blok.

Bemærk: Brug **Return** uden et argument med en **Prgm...EndPrgm**-blok for at lukke et program.

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

```
Define factorial (nn)=
Func
Local answer,counter
1 → answer
For counter,1,nn
answer·counter → answer
EndFor
Return answer
EndFunc
```

factorial (3)

6

right()

right(*List1* [, *Num*]) ⇒ *liste*

Returnerer *Num*-elementer længst til højre i *List1*.

Hvis du udelader *Num*, returneres hele *List1*.

right(*sourceString* [, *Num*]) ⇒ *string*

Returnerer *Num*-tegn længst til højre i tegnstrengen *sourceString*.

Hvis du udelader *Num*, returneres hele *sourceString*.

right({ 1,3,-2,4 },3)

{ 3,-2,4 }

right("Hello",2)

"lo"

right(Comparison) ⇒ *udtryk*right($x < 3$) 3

Returnerer højre side af en ligning eller ulighed.

rk23 ()

rk23(Expr, Var, depVar, {Var0, VarMax}, depVar0, VarStep [, diftol])
⇒ *matrix*

rk23(SystemOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep[, diftol])
⇒ *matrix*

rk23(ListOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep[, diftol])
⇒ *matrix*

Anvender Runge-Kutta-metoden til at løse systemet

$$\frac{d \text{ depVar}}{d \text{ Var}} = \text{Expr}(\text{Var}, \text{depVar})$$

med $\text{depVar}(\text{Var0}) = \text{depVar0}$ i intervallet $[\text{Var0}, \text{VarMax}]$. Returnerer en matrix, hvor første række definerer Var -outputværdierne som defineret af VarStep . Den anden række definerer værdien af den første løsningskomponent i de tilsvarende Var -værdier, osv.

Expr er højresiden, som definerer den ordinære differentialligning (ODE - ordinary differential equation).

SystemOfExpr er et system af højresider, der definerer ODE'erne (svarende til rækkefølgen af afhængige variable i *ListOfDepVars*).

ListOfExpr er en liste af højresider, der definerer systemet af ODE'er (svarende til rækkefølgen af afhængige variable i *ListOfDepVars*).

Var er den uafhængige variabel.

Differentialligning:

$$y' = 0.001 \cdot y \cdot (100 - y) \text{ og } y(0) = 10$$

$$\text{rk23}\left(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1\right)$$

0.	1.	2.	3.	4.
10.	10.9367	11.9493	13.042	14.2

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

Samme ligning med *diftol* sat til 1.E-6

$$\text{rk23}\left(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1, 1.E-6\right)$$

0.	1.	2.	3.	4.
10.	10.9367	11.9495	13.0423	14.2189

Sammenlign resultatet ovenfor med den eksakte CAS løsning opnået ved brug af *deSolve()* og *seqGen()*:

$$\text{deSolve}(y' = 0.001 \cdot y \cdot (100 - y) \text{ and } y(0) = 10, t, y)$$

$$y = \frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9.}$$

$$\text{seqGen}\left(\frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9.}, t, y, \{0, 100\}\right)$$

{ 10., 10.9367, 11.9494, 13.0423, 14.2189, 15.49 }

System af ligninger:

$$\begin{cases} y1' = -y1 + 0.1 \cdot y1 \cdot y2 \\ y2' = 3 \cdot y2 - y1 \cdot y2 \end{cases}$$

med $y1(0) = 2$ og $y2(0) = 5$

ListOfDepVars er en liste med afhængige variabler.

$\{Var0, VarMax\}$ er en liste med to elementer, der informerer funktionen om at integrere fra *Var0* til *VarMax*.

ListOfDepVars0 er en liste af oprindelige værdier for afhængige variabler.

Hvis *VarStep* evalueres til et tal forskelligt fra nul: $\text{sign}(VarStep) = \text{sign}(VarMax - Var0)$ og løsninger returneres ved $Var0 + i \cdot VarStep$ for alle $i=0,1,2,\dots$ således at $Var0 + i \cdot VarStep$ er i $[var0, VarMax]$ (der vil muligvis ikke være en løsningsværdi ved *VarMax*).

Hvis *VarStep* evalueres til nul, returneres løsningerne i "Runge-Kutta" *Var*-værdier.

difftol er fejltolerancen (som standard 0.001).

$$\text{rk23} \left(\begin{matrix} -y1+0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{matrix}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1 \right)$$

0.	1.	2.	3.	4.
2.	1.94103	4.78694	3.25253	1.82848
5.	16.8311	12.3133	3.51112	6.27245

root()

$\text{root}(Expr) \Rightarrow \text{rod}$

$\text{root}(Expr1, Expr2) \Rightarrow \text{rod}$

$\text{root}(Expr)$ returnerer kvadratroden af *Expr*.

$\text{root}(Expr1, Expr2)$ returnerer *Expr2*-roden af *Expr1*. *Expr1* kan være en reel eller kompleks konstant med flydende decimaler, et heltal eller en kompleks rational konstant eller et generelt symbolsk udtryk.

Bemærk: Se også **N-te rod-skabelon**, side 1.

$\sqrt[3]{8}$	2
$\sqrt[3]{3}$	$\frac{1}{3^3}$
$\sqrt[3]{3.}$	1.44225

rotate()

$\text{rotate}(IntegerI[, \#ofRotations]) \Rightarrow \text{heltal}$

I binær tilstand:

rotate()Katalog > 

Hvis *#ofRotations* er positivt, kører rotationen mod venstre. Hvis *#ofRotations* er negativt, kører rotationen mod højre. Standardindstillingen -1 (rotér en bit til højre).

round()Katalog > 

round(*Expr1*[, *digits*]) $\Rightarrow$ *udtryk*

$\text{round}(1.234567, 3)$	1.235
-----------------------------	-------

Returnerer argumentet afrundet til det angivne antal cifre efter decimalpunktet.

digits skal være et heltal i området 0–12. Hvis *digits* ikke er inkluderet, returneres argumentet afrundet til 12 væsentlige cifre.

Bemærk: Vis cifre-tilstanden kan påvirke den måde, dette vises på.

round(*List1*[, *digits*]) $\Rightarrow$ *liste*

$\text{round}(\{\pi, \sqrt{2}, \ln(2)\}, 4)$	$\{3.1416, 1.4142, 0.6931\}$
--	------------------------------

Returnerer en liste med elementerne afrundet til det angivne antal cifre.

round(*Matrix1*[, *digits*]) $\Rightarrow$ *matrix*

$\text{round}\left(\begin{bmatrix} \ln(5) & \ln(3) \\ \pi & e^1 \end{bmatrix}, 1\right)$	$\begin{bmatrix} 1.6 & 1.1 \\ 3.1 & 2.7 \end{bmatrix}$
--	--

Returnerer en matrix med elementerne afrundet til det angivne antal cifre.

rowAdd()Katalog > 

rowAdd(*Matrix1*, *rIndex1*, *rIndex2*) $\Rightarrow$ *matrix*

$\text{rowAdd}\left(\begin{bmatrix} 3 & 4 \\ -3 & -2 \end{bmatrix}, 1, 2\right)$	$\begin{bmatrix} 3 & 4 \\ 0 & 2 \end{bmatrix}$
--	--

Returnerer en kopi af *Matrix1* med rækken *rIndex2* udskiftet med summen af rækker *rIndex1* og *rIndex2*.

$\text{rowAdd}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}, 1, 2\right)$	$\begin{bmatrix} a & b \\ a+c & b+d \end{bmatrix}$
--	--

rowDim()Katalog > 

rowDim(*Matrix*) $\Rightarrow$ *udtryk*

Returnerer antallet af rækker i *Matrix*.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$
--	---

Bemærk: Se også **colDim()**, side 28.

$\text{rowDim}(m1)$	3
---------------------	---

rowNorm()

Katalog >

rowNorm(*Matrix*) $\Rightarrow$ *udtryk*

Returnerer maksimum for summerne af de absolutte værdier for elementerne i *Matrix*-rækkerne.

Bemærk: Alle matricens elementer skal kunne reduceres til tal. Se også **colNorm()**, side 28.

$$\text{rowNorm}\left(\begin{bmatrix} -5 & 6 & -7 \\ 3 & 4 & 9 \\ 9 & -9 & -7 \end{bmatrix}\right) = 25$$

rowSwap()

Katalog >

rowSwap(*Matrix1*, *rIndex1*, *rIndex2*)
 $\Rightarrow$ *matrix*

Returnerer *Matrix1* med rækkerne *rIndex1* og *rIndex2* byttet.

$$\begin{array}{ccc} \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} & \rightarrow \text{mat} & \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \\ \text{rowSwap}(\text{mat}, 1, 3) & & \begin{bmatrix} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{bmatrix} \end{array}$$

rref()

Katalog >

rref(*Matrix1* [, *Tol*]) $\Rightarrow$ *matrix*

Returnerer den reducerede række-echelonform af *Matrix1*.

$$\begin{array}{ccc} \text{rref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right) & & \begin{bmatrix} 1 & 0 & 0 & \frac{66}{71} \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix} \end{array}$$

Ethvert element af matricen kan valgfrit behandles som nul, hvis dets absolutte værdi er mindre end *Tol*. Denne tolerance anvendes kun, hvis matricen har elementer med flydende decimaler og ikke indeholder symbolske variable, der ikke er tildelt en værdi. Ellers ignoreres *Tol*.

$$\text{rref}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Hvis du anvender eller indstiller **Auto-** eller **Approximate** - tilstanden til Approximate, foretages beregningerne med flydende komma.
- Hvis *Tol* udelades eller ikke anvendes, beregnes standardtolerancen som:
 $5E-14 \cdot \max(\dim(\text{Matrix1})) \cdot \text{rowNorm}(\text{Matrix1})$

Bemærk: Se også **ref()**, page 155.

sec() **-tast****sec**(*Udtr1*) $\Rightarrow$ *udtryk*

I vinkeltilstanden Grader:

sec(*Liste1*) $\Rightarrow$ *liste*

Returnerer sekans til *Udtr1* eller
returnerer en liste med sekansen til alle
elementer i *Liste1*.

Bemærk: Argumentet fortolkes som en
vinkel målt i grader, nygrader eller
radianer afhængigt af den aktuelt
indstillede vinkeltilstand. Du kan bruge
°, G eller R til midlertidigt at ignorere
vinkeltilstanden.

$$\begin{array}{l} \text{sec}(45) \quad \sqrt{2} \\ \text{sec}(\{1, 2, 3, 4\}) \quad \left\{ \frac{1}{\cos(1)}, 1.00081, \frac{1}{\cos(4)} \right\} \end{array}$$

sec⁻¹() **-tast****sec⁻¹**(*Udtr1*) $\Rightarrow$ *udtryk*

I vinkeltilstanden Grader:

sec⁻¹(*Liste1*) $\Rightarrow$ *liste*

Returnerer den vinkel, hvis sekans er
Udtr1, eller returnerer en liste med de
inverse sekanser til de enkelte elementer
i *Liste1*.

Bemærk: Resultatet returneres som en
vinkel i grader eller radianer afhængigt
af den aktuelle vinkeltilstand.

Bemærk: Du kan indsætte denne
funktion fra computerens tastatur ved at
skrive **arcsec (...)**.

$$\text{sec}^{-1}(1) \quad 0$$

I vinkeltilstanden Nygrader:

$$\text{sec}^{-1}(\sqrt{2}) \quad 50$$

I vinkeltilstanden Radian:

$$\text{sec}^{-1}(\{1, 2, 5\}) \quad \left\{ 0, \frac{\pi}{3}, \text{sec}^{-1}\left(\frac{1}{5}\right) \right\}$$

sech()**Katalog >** **sech**(*Udtr1*) $\Rightarrow$ *udtryk***sech**(*Liste1*) $\Rightarrow$ *liste*

Returnerer den hyperbolske sekans til
Udtr1 eller returnerer en liste med den
hyperbolske sekans til elementerne i
Liste1.

$$\begin{array}{l} \text{sech}(3) \quad \frac{1}{\cosh(3)} \\ \text{sech}(\{1, 2, 3, 4\}) \quad \left\{ \frac{1}{\cosh(1)}, 0.198522, \frac{1}{\cosh(4)} \right\} \end{array}$$

sech⁻¹(Udtr1) ⇒ *udtryk*

sech⁻¹(Liste1) ⇒ *liste*

Returnerer den inverse hyperbolske sekans til *Udtr1* eller returnerer en liste med den inverse hyperbolske sekans til de enkelte elementer i *Liste1*.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arcsech** (...).

I vinkeltilstanden radian og tilstanden rektangulært kompleks:

sech ⁻¹ (1)	0
sech ⁻¹ ($\{1, -2, 2, 1\}$)	$\left\{0, \frac{2 \cdot \pi}{3} \cdot i, 8 \cdot 10^{-15} + 1.07448 \cdot i\right\}$

Send

Hub-menu

Send *exprOrString1* [, *exprOrString2*] ...

Programmeringskommando: Send én eller flere TI-Innovator™ Hub kommandoer til en tilsluttet hub.

exprOrString skal være en gyldig TI-Innovator™ Hub kommando. Normalt vil *exprOrString* indeholde en "SET ..." kommando for at kontrollere en enhed eller en "READ ..." kommando for at anmode om data.

Argumenterne sendes til hubben efter hinanden.

Bemærk: Du kan bruge kommandoen **Send** i et brugerdefineret program, men ikke i en funktion.

Bemærk: Se også **Get** (side 83), **GetStr** (side 90), og **eval()** (side 66).

Eksempel: Tænd det blå element i den indbyggede RGB LED i 0,5 sekunder.

Send "SET COLOR.BLUE ON TIME .5"	Done
----------------------------------	------

Eksempel: Anmod om den nuværende værdi for hubbens indbyggede lysniveausensor. En **Get**-kommando henter værdien og tildeler den til den variable *lightval*.

Send "READ BRIGHTNESS"	Done
Get <i>lightval</i>	Done
<i>lightval</i>	0.347922

Eksempel: Send en beregnet frekvens til hubbens indbyggede højttaler. Brug den særlige variable *iostr.SendAns* til at vise hub-kommandoen med udtrykket evalueret.

<i>n</i> :=50	50
<i>m</i> :=4	4
Send "SET SOUND eval(<i>m</i> · <i>n</i>)"	Done
<i>iostr.SendAns</i>	"SET SOUND 200"

seq()Katalog > **seq(Udtr, Var, Lav, Høj[, Trin])**⇒liste

Øger *Var* fra *Low* til *High* i trin på *Step*, beregner *Expr* og returnerer resultaterne som en liste. Det oprindelige indhold af *Var* er der stadigvæk, når **seq()** er gennemført.

Standardværdien for *trin* = 1.

$$\begin{array}{l} \text{seq}\left(n^2, n, 1, 6\right) \quad \{1, 4, 9, 16, 25, 36\} \\ \text{seq}\left(\frac{1}{n}, n, 1, 10, 2\right) \quad \left\{1, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \frac{1}{9}\right\} \\ \text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right) \quad \frac{1968329}{1270080} \end{array}$$

Bemærk: Sådanne gennemtvungne et tilnærmet resultat,

Håndholdt: Tryk på  .

Windows®: Tryk **Ctrl+Enter**.

Macintosh®: Tryk på **⌘+Enter**.

iPad®: Hold **ENTER** nede, og vælg .

$$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right) \quad 1.54977$$

seqGen()Katalog > 

seqGen(Expr, Var, depVar, {Var0, VarMax}[, ListOfInitTerms [, VarStep [, CeilingValue]]]) ⇒liste

Genererer en liste med led for sekvensen $depVar(Var) = Expr$ som følger: Øger den uafhængige variabel *Var* fra *Var0* til *VarMax* med *VarStep*, beregner *depVar* (*Var*) for tilsvarende værdier af *Var* vha. udtrykket *Expr* og *ListOfInitTerms* og returnerer resultaterne som en liste.

seqGen(ListOrSystemOfExpr, Var, ListOfDepVars, {Var0, VarMax}[, MatrixOfInitTerms [, VarStep [, CeilingValue]]]) ⇒matrix

Genererer de første fem led i sekvensen $u(n) = u(n-1)^2/2$ med $u(1)=2$ og $VarStep=1$.

$$\begin{array}{l} \text{seqGen}\left(\frac{u(n-1)^2}{n}, n, u, \{1, 5\}, \{2\}\right) \\ \{2, 2, \frac{4}{3}, \frac{4}{9}, \frac{16}{405}\} \end{array}$$

Eksempel, hvor $Var0=2$:

$$\begin{array}{l} \text{seqGen}\left(\frac{u(n-1)+1}{n}, n, u, \{2, 5\}, \{3\}\right) \\ \left\{3, \frac{4}{3}, \frac{7}{12}, \frac{19}{60}\right\} \end{array}$$

Eksempel, hvor den startværdien er symbolsk:

$$\begin{array}{l} \text{seqGen}\left(u(n-1)+2, n, u, \{1, 5\}, \{a\}\right) \\ \{a, a+2, a+4, a+6, a+8\} \end{array}$$

System med to sekvenser:

Genererer en matrix af led for et system (eller en liste) af sekvenser

ListOfDepVars

(*Var*)=*ListOrSystemOfExpr* som følger:

Øger den uafhængige variabel *Var* fra *Var0* til *VarMax* med *VarStep*, beregner

ListOfDepVars(Var) for tilsvarende

værdier af *Var* vha. udtrykket

ListOrSystemOfExpr og

MatrixOfInitTerms, og returnerer resultaterne som en matrix.

Det oprindelige indhold af *Var* er uændret efter **seqGen()** er gennemført .

Standardværdien for *VarStep* = 1.

$$\text{seqGen}\left(\left\{\frac{1}{n}, \frac{u_1(n-1)}{2} + u_1(n-1)\right\}, n, \{u_1, u_2\}, \{1, 5\}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$$

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{2} & \frac{19}{2} \end{bmatrix}$$

Bemærk: Void () i den oprindelige ledmatrix ovenfor bruges for at angive, at det oprindelige led for $u_1(n)$ er beregnet vha. den eksplicitte sekvensformel $u_1(n)=1/n$.

seqn()

seqn(*Expr*(*u*, *n* [, *ListOfInitTerms* [, *nMax* [, *CeilingValue*]]]) $\Rightarrow$ liste

Genererer en liste med led for sekvensen $u(n)=\text{Expr}(u, n)$ som følger: Øger *n* fra 1 til *nMax* med 1, beregner $u(n)$ for de tilsvarende værdier af *n* vha. udtrykket $\text{Expr}(u, n)$ og *ListOfInitTerms*, og returnerer resultaterne som en liste.

seqn(*Expr*(*n* [, *nMax* [, *CeilingValue*]])) $\Rightarrow$ liste

Genererer en liste med led for sekvens $u(n)=\text{Expr}(n)$ som følger: Øger *n* fra 1 til *nMax* med 1, beregner $u(n)$ for de tilsvarende værdier af *n* vha. udtrykket $\text{Expr}(n)$ og returnerer resultaterne som en liste.

Hvis *nMax* mangler, sættes *nMax* til 2500

Hvis *nMax*=0, sættes *nMax* til 2500

Bemærk: **seqn()** kalder **seqGen()** med *n0*=1 og *nstep* =1

Genererer de første fem led i sekvensen $u(n) = u(n-1)/2$ med $u(1)=2$.

$$\text{seqn}\left(\frac{u(n-1)}{n}, \{2\}, 6\right)$$

$$\left\{2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360}\right\}$$

$$\text{seqn}\left(\frac{1}{n^2}, 6\right)$$

$$\left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}\right\}$$

series(*Udtr1*, *Var*, *Orden* [, *Punkt*]) $\Rightarrow$ udtryk

series(*Udtr1*, *Var*, *Orden* [, *Punkt*]) | *Var*>*Punkt* $\Rightarrow$ udtryk

series(*Udtr1*, *Var*, *Orden* [, *Punkt*]) | *Var*<*Punkt* $\Rightarrow$ udtryk

Returnerer en generel afkortet repræsentation af en potensrække af *Udtr1* udviklet omkring *Punkt* til graden *Orden*. *Ordenen* kan være et rationelt tal. De resulterende potenser af (*Var* – *Punkt*) kan indeholde negative og/eller brøkeksponenter. Koefficienterne af disse potenser kan indeholde logaritmer af (*Var* – *Punkt*) og andre funktioner af *Var*, der er domineret af alle potenser af (*Var* – *Punkt*), der har det samme eksponentfortegn.

Punkt er som standard 0. *Punkt* kan være ∞ eller $-\infty$, i hvilke tilfælde udviklingen er gennem graden *Orden* i $1/(Var - Punkt)$.

series(...) returnerer “**series(...)**” hvis der ikke kan bestemmes en sådan repræsentation, som for eksempel for væsentlige singulariteter som $\sin(1/z)$ ved $z=0$, $e^{-1/z}$ ved $z=0$, eller e^z ved $z = \infty$ eller $-\infty$.

$$\text{series}\left(\frac{1-\cos(x-1)}{(x-1)^2}, x, 4, 1\right) \quad \frac{1}{2} - \frac{(x-1)^2}{24} + \frac{(x-1)^4}{720}$$

$$\text{series}\left(\frac{-1}{e^{z-}}, z, -1\right) \quad z - 1$$

$$\text{series}\left(\left(1 + \frac{1}{n}\right)^n, n, 2, \infty\right) \quad e - \frac{e}{2 \cdot n} + \frac{11 \cdot e}{24 \cdot n^2}$$

$$\text{series}\left(\tan^{-1}\left(\frac{1}{x}\right), x, 5\right), x > 0 \quad \frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5}$$

$$\text{series}\left(\int \frac{\sin(x)}{x} dx, x, 6\right) \quad x - \frac{x^3}{18} + \frac{x^5}{600}$$

$$\text{series}\left(\int_0^x \sin(x \cdot \sin(t)) dt, x, 7\right) \quad \frac{x^3}{2} - \frac{x^5}{24} - \frac{29 \cdot x^7}{720}$$

$$\text{series}\left(\left(1 + e^x\right)^2, x, 2, 1\right) \quad (e+1)^2 + 2 \cdot e \cdot (e+1) \cdot (x-1) + e \cdot (2 \cdot e+1) \cdot (x-1)^2$$

Hvis rækken eller en af dens afledte afbrydes af et spring ved *Punkt*, indeholder resultatet sandsynligvis deludtryk på formen $\text{sign}(\dots)$ eller $\text{abs}(\dots)$ for en reel udviklingsvariabel eller $(-1)^{\text{floor}(\dots \text{angle}(\dots) \dots)}$ for en kompleks udviklingsvariabel, der slutter med “ ”. Hvis du kun vil bruge rækken til værdier på den ene side af *Punkt*, skal du tilføje den korrekte blandt funktionerne “| $\text{Var} > \text{Punkt}$ ”, “| $\text{Var} < \text{Punkt}$ ”, “| $\text{Var} \geq \text{Punkt}$ ”, eller “ $\text{Var} \leq \text{Punkt}$ ” for at opnå et enklere resultat.

series() kan give symbolske tilnærmelser til ubestemte og bestemte integraler, for hvilke, der ellers ikke kan opnås symbolske løsninger.

series() fordeler sig over lister og matricer med 1.-argument.

series() er en generaliseret version af **taylor()**.

Som vist med det sidste eksempel til højre kan visningsrutinerne flytte rundt på leddene i det resultat $\text{taylor}(\dots)$ giver,, så det dominerende led ikke er det længst til venstre.

Bemærk: Se også **dominantTerm()**, side 60.

setMode()

setMode(tilstandNavnHeltal, indstilHeltal) $\Rightarrow$ heltal **setMode(liste) $\Rightarrow$ heltalsliste**

Kun gyldig i en funktion eller et program.

Viser en tilnærmet værdi af π med standardindstillingen for Viste cifre, og viser derefter π med en indstilling på Fix2. Sørg for, at standardindstillingen gendannes efter programmet er eksekveret.

setMode(*tilstandNavnHeltal*, *indstilHeltal*) indstiller midlertidigt tilstanden *tilstandNavnHeltal* til den nye indstilling *indstilHeltal* og returnerer et heltal, der svarer til denne tilstands oprindelige indstilling. Ændringen er begrænset til varigheden af eksekveringen af programmet/funktionen.

tilstandNavnHeltal angiver hvilken tilstand, du vil indstille. Det skal være et af tilstandsheltallene fra nedenstående tabel.

indstilHeltal angiver den nye indstilling for tilstanden. Det skal være et af indstillingsheltallene for den tilstand, du indstiller.

setMode(*liste*) lader dig ændre flere indstillinger. *liste* indeholder et par af tilstandsheltal og indstillingsheltal.

setMode(*liste*) returnerer en tilsvarende liste, hvis heltalspar repræsenterer de oprindelige tilstande og indstillinger.

Hvis du har gemt alle tilstandsindstillinger med **getMode(0)** → *var*, kan du anvende **setMode(var)** til at gendanne disse indstillinger, indtil funktionen eller programmet afsluttes. Se **getMode()**, side 89.

Bemærk: De aktuelle tilstandsindstillinger videregives til kaldte under rutiner. Hvis en under rutine ændrer en tilstandsindstilling, mistes tilstandsændringen, når kontrollen returnerer til den kaldende rutine.

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

Define <i>prog1()</i> =Prgm	Done
Disp approx(π)	
setMode(1,16)	
Disp approx(π)	
EndPrgm	
<i>prog1()</i>	
	3.14159
	3.14
	Done

Tilstands- navn	Tilstands- heltal	Indstillingsheltal
Viste cifre	1	1=Float, 2=Float1, 3=Float2, 4=Float3, 5=Float4, 6=Float5, 7=Float6, 8=Float7, 9=Float8, 10=Float9, 11=Float10, 12=Float11, 13=Float12, 14=Fix0, 15=Fix1, 16=Fix2, 17=Fix3, 18=Fix4, 19=Fix5, 20=Fix6, 21=Fix7, 22=Fix8, 23=Fix9, 24=Fix10, 25=Fix11, 26=Fix12
Vinkel	2	1=Radian, 2=Grader, 3=Gradian
Eksponentielt format	3	1=Normal, 2=Videnskabelig, 3=Teknisk
Reel eller kompleks	4	1=Reel, 2=Rektangulær, 3=Polær
Auto eller tilnærmet	5	1=Auto, 2=Tilnærmet, 3=Eksakt
Vektorformat	6	1=Rektangulær, 2=Cylindrisk, 3=Sfærisk
Talsystem	7	1=Decimal, 2=Hex, 3=Binær
Enhedssystem	8	1=SI, 2=Eng/US

shift()

Katalog > 

shift(Heltal1[,antalFlyt])⇒heltal

Flytter bittene i et binært heltal. Du kan indtaste *Heltal1* i ethvert talsystem. Det konverteres automatisk til en 64-bit binær form med fortegn. Hvis *Heltal1* er for stort til denne form, bringer en symmetrisk modulooperation værdien ind i det korrekte område. Yderligere oplysninger findes under **Base2**, side 18.

Hvis *antalFlytninger* er positivt, kører flytningen mod venstre. Hvis *antalFlytninger* er negativt, kører flytningen mod højre. Standardindstillingen er -1 (flytter en bit til højre).

I en flytning til højre droppes bittene længst mod højre, og 0 eller 1 indsættes for at matche bittene længst til venstre. I en flytning til venstre droppes bittene længst mod venstre, og 0 indsættes som bittene længst til højre.

I binær tilstand:

shift(0b1111010110000110101)	0b111101011000011010
shift(256,1)	0b1000000000

I hexadecimal tilstand:

shift(0h78E)	0h3C7
shift(0h78E, -2)	0h1E3
shift(0h78E, 2)	0h1E38

Vigtigt: Til binære eller hexadecimal indtastninger skal du som præfiks altid benytte henholdsvis 0b eller 0h (nul, ikke bogstavet O).

For eksempel i en højreflytning:

Hver bit flytter til højre.

0b0000000000000111101011000011010

Indsætter 0, hvis bitten længst til venstre er 0,

eller 1, hvis bitten længst til venstre er 1.

giver:

00b0000000000000011110101100001101

Resultatet vises i den valgte tilstand for talsystem. Der vises ikke foranstillede nuller.

shift(Liste1 [,antalFlytninger])⇒liste

Returnerer en kopi af *Liste1* flyttet til højre eller venstre med *antalflytninger* elementer. Ændrer ikke *Liste1*.

Hvis *antalFlytninger* er positivt, kører flytningen mod venstre. Hvis *antalFlytninger* er negativt, kører flytningen mod højre. Standardindstillingen er -1 (flyt en bit til højre).

Elementer indført i starten eller slutningen af *liste* af flytningen, indstilles til symbol "undef".

shift(Streng1 [,antalFlytninger])⇒streng

Returnerer en kopi af *Streng1* flyttet til højre eller venstre med *antalflytninger* tegn. Ændrer ikke *Streng1*.

Hvis *antalFlytninger* er positivt, kører flytningen mod venstre. Hvis *antalFlytninger* er negativt, kører flytningen mod højre. Standardindstillingen er -1 (flyt et tegn til højre).

Tegn indført i starten eller slutningen af *streng* af flytningen, indstilles til et mellemrum.

I decimal tilstand:

shift({1,2,3,4})	{undef,1,2,3}
shift({1,2,3,4},-2)	{undef,undef,1,2}
shift({1,2,3,4},2)	{3,4,undef,undef}

shift("abcd")	" abc"
shift("abcd",-2)	" ab"
shift("abcd",1)	"bcd "

sign()Katalog > **sign(Udtr1)**⇒*udtryk*

$\text{sign}(-3.2)$	-1.
---------------------	-----

sign(Liste1)⇒*liste*

$\text{sign}(\{2, 3, 4, -5\})$	$\{1, 1, 1, -1\}$
--------------------------------	-------------------

sign(Matrix1)⇒*matrix*

$\text{sign}(1+ x)$	1
----------------------	---

For reelle og komplekse *Udtr1*,
returneres *Udtr1/abs(Udtr1)*, når
Udtr1≠0.

Hvis kompleks formattilstand er real:

$\text{sign}\begin{pmatrix} -3 & 0 & 3 \end{pmatrix}$	$\begin{bmatrix} -1 & \pm 1 & 1 \end{bmatrix}$
---	--

Returnerer 1, hvis *Udtr1* er positivt.Returnerer -1, hvis *Udtr1* er negativt.

sign(0) returnerer ±1, hvis det komplekse
format er reel. Ellers returnerer det sig
selv.

sign(0) repræsenterer enhedscirklen i det
komplekse domæne.

For en liste eller matrix returneres
fortegnene af elementerne.

simult()Katalog > **simult(koeffMatrix, konstVektor[, Tol])**⇒*matrix*

Løs for x og y:

$$x + 2y = 1$$

$$3x + 4y = -1$$

Returnerer en kolonnevektor, der
indeholder løsningerne til et system af
lineære ligninger.

$\text{simult}\left(\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}\right)$	$\begin{bmatrix} -3 \\ 2 \end{bmatrix}$
---	---

Bemærk: Se også **linSolve()**, side 108.

Løsningen er x=-3 og y=2.

koeffMatrix skal være en kvadratisk
matrix, der indeholder koefficienterne til
ligningerne.

Løs:

$$ax + by = 1$$

$$cx + dy = 2$$

konstVektor skal have samme antal
rækker (samme dimension) som
koeffMatrix og indeholde konstanterne.

Ethvert matricelement kan valgfrit
behandles som nul, hvis den absolutte
værdi er mindre end *Tol*. Denne
tolerance anvendes kun, hvis matricen
har elementer med flydende decimaler
og ikke indeholder symbolske variable,
der ikke er tildelt en værdi. Ellers
ignoreres *Tol*.

$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \rightarrow \text{matx1}$	$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$
$\text{simult}\left(\text{matx1}, \begin{pmatrix} 1 \\ 2 \end{pmatrix}\right)$	$\begin{bmatrix} -(2 \cdot b - d) \\ a \cdot d - b \cdot c \\ 2 \cdot a - c \\ a \cdot d - b \cdot c \end{bmatrix}$

- Hvis du sætter tilstanden **Auto eller tilnærmet** til Approks, foretages beregningerne med aritmetik med flydende komma.
- Hvis *tol* udelades eller ikke anvendes, beregnes standardtolerancen som:
 $5E-14 \cdot \max(\dim(koeffMatrix)) \cdot \text{rækkeNorm}(koeffMatrix)$

simult(koeffMatrix, konstMatrix[, tol]) ⇒ matrix

Løser flere systemer af lineære ligninger, hvor hvert system har de samme ligningskoefficienter men forskellige konstanter.

Hver kolonne i *konstMatrix* skal indeholde konstanterne for et ligningssystem. Hver kolonne i den resulterende indeholder løsningen for det tilsvarende system.

Løs:

$$x + 2y = 1$$

$$3x + 4y = -1$$

$$x + 2y = 2$$

$$3x + 4y = -3$$

$$\text{simult}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ -1 & -3 \end{bmatrix}\right) \quad \begin{bmatrix} -3 & -7 \\ 2 & \frac{9}{2} \end{bmatrix}$$

For det første system er $x=-3$ og $y=2$. For det andet system er $x=-7$ og $y=9/2$.

Udtr ►sin

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>**sin**.

Repræsenterer *Udtr* i termer af sinus. Det er en konverteringsoperator for visning. Denne operator kan kun anvendes ved slutningen af indtastningslinjen.

►sin reducerer alle potenser af

$$\cos(\dots) \text{ modulo } 1 - \sin(\dots)^2$$

således, at alle tilbageværende potenser af $\sin(\dots)$ har eksponenter i området (0, 2). Således vil resultatet være uden $\cos(\dots)$ hvis, og kun hvis $\cos(\dots)$ optræder i det givne udtryk, kun med lige potenser.

$$\frac{(\cos(x))^2 \text{ ►sin}}{1 - (\sin(x))^2}$$

Bemærk: Konverteringsoperatører understøttes ikke i vinkeltilstand, Grader eller Nygrader. Før brug skal man sikre at vinkeltilstanden er indstillet til radianer, og at *Udtr* ikke indeholder eksplicit reference til grader eller nygrader.

sin()

 -tast

sin(*Udtr1*) ⇒ *udtryk*

I vinkeltilstanden Grader:

sin(*Liste1*) ⇒ *liste*

$$\sin\left(\frac{\pi}{4}\right) = \frac{\sqrt{2}}{2}$$

sin(*Udtr1*) returnerer sinus af argumentet som et udtryk.

$$\sin(45^\circ) = \frac{\sqrt{2}}{2}$$

sin(*Liste1*) returnerer en liste med sinus til alle elementer i *Liste1*.

$$\sin(\{0,60,90\}) = \left\{0, \frac{\sqrt{3}}{2}, 1\right\}$$

Bemærk: Argumentet fortolkes som en vinkel i enten grader eller radianer afhængigt af den aktuelle vinkeltilstand. Du kan bruge °, °G, eller ° til midlertidigt at tilsidesætte den indstillede vinkeltilstand.

I vinkeltilstanden Nygrader:

$$\sin(50^\circ) = \frac{\sqrt{2}}{2}$$

I vinkeltilstanden Radian:

$$\sin\left(\frac{\pi}{4}\right) = \frac{\sqrt{2}}{2}$$

$$\sin(45^\circ) = \frac{\sqrt{2}}{2}$$

sin(*kvadratMatrix1*) ⇒ *kvadratMatrix*

I vinkeltilstanden Radian:

Returnerer matrixsinus af *kvadratMatrix1*. Dette er ikke det samme som at beregne sinus for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

$$\sin\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$$

KvadratMatrix1 skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

$\sin^{-1}()$



$\sin^{-1}(Udtr1) \Rightarrow udtryk$

I vinkeltilstanden Grader:

$\sin^{-1}(Liste1) \Rightarrow liste$

$\sin^{-1}(1)$ 90

$\sin^{-1}(Udtr1)$ returnerer den vinkel, hvis sinus er $Udtr1$ som et udtryk.

I vinkeltilstanden Nygrader:

$\sin^{-1}(Liste1)$ returnerer en liste med de inverse sinusværdier for hvert element af $Liste1$.

$\sin^{-1}(1)$ 100

Bemærk: Resultatet returneres som en vinkel i grader eller radianer afhængigt af den aktuelle vinkeltilstand.

I vinkeltilstanden Radian:

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arcsin (...)**.

$\sin^{-1}\{0,0,2,0,5\}$ $\{0,0,201358,0,523599\}$

$\sin^{-1}(kvadratMatrix1) \Rightarrow kvadratMatrix$

I vinkeltilstanden radian og tilstanden rektangulært komplekst format:

Returnerer den matrixinverse sinus af $kvadratMatrix1$. Dette er ikke det samme som at beregne den inverse sinus for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

$\sin^{-1}\left(\begin{bmatrix} 1 & 5 \\ 4 & 2 \end{bmatrix}\right)$
 $\begin{bmatrix} -0.174533-0.12198 \cdot i & 1.74533-2.35591 \cdot i \\ 1.39626-1.88473 \cdot i & 0.174533-0.593162 \cdot i \end{bmatrix}$

$KvadratMatrix1$ skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

$\sinh()$

Katalog >

$\sinh(Udtr1) \Rightarrow udtryk$

$\sinh(1.2)$ 1.50946

$\sinh(Liste1) \Rightarrow liste$

$\sinh\{0,1,2,3\}$ $\{0,1.50946,10.0179\}$

$\sinh(Udtr1)$ returnerer den hyperbolske sinus af argumentet som et udtryk.

$\sinh(Liste1)$ returnerer en liste af de hyperbolske sinuser af hvert element af $Liste1$.

I vinkeltilstanden Radian:

$\sinh(kvadratMatrix1) \Rightarrow kvadratMatrix$

Returnerer den matrixhyperbolske sinus af $kvadratMatrix1$. Dette er ikke det samme som at beregne den hyperbolske sinus for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

sinh()

Katalog > 

KvadratMatrix1 skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

$$\sinh \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 360.954 & 305.708 & 239.604 \\ 352.912 & 233.495 & 193.564 \\ 298.632 & 154.599 & 140.251 \end{bmatrix}$$

sinh⁻¹()

Katalog > 

sinh⁻¹(Udtr1)⇒udtryk

$$\sinh^{-1}(0) \quad 0$$

sinh⁻¹(Liste1)⇒liste

$$\sinh^{-1}(\{0, 2, 1, 3\}) \quad \{0, 1.48748, \sinh^{-1}(3)\}$$

sinh⁻¹(Udtr1) returnerer den inverse hyperbolske sinus af argumentet som et udtryk.

sinh⁻¹(Liste1) returnerer den inverse hyperbolske sinus til hvert element i *Liste1*.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arcsinh(...)**.

sinh⁻¹(kvadratMatrix1)⇒kvadratMatrix

I vinkeltilstanden Radian:

Returnerer den matrixinvers hyperbolske sinus af *kvadratMatrix1*. Dette er ikke det samme som at beregne den inverse hyperbolske sinus for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

$$\sinh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 0.041751 & 2.15557 & 1.1582 \\ 1.46382 & 0.926568 & 0.112557 \\ 2.75079 & -1.5283 & 0.57268 \end{bmatrix}$$

KvadratMatrix1 skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

SinReg

Katalog > 

SinReg *X*, *Y* [, [*Iterationer*],[*Periode*] [, [*Kategori*, *Medtag*]]

Beregner sinusregressionen på listerne *X* og *Y*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

X og Y er lister med uafhængige og afhængige variable.

Iterationer er en valgfri værdi, som angiver det maksimale antal gange (1 til 16) en løsning vil forsøges. Hvis udeladt, anvendes 8. Typisk resulterer større værdier i større nøjagtighed men længere eksekveringstider og omvendt.

Periode angiver en estimeret periode. Hvis den udelades, skal forskellen mellem værdierne i X være lige store og i sekventiel orden. Hvis du angiver *Periode*, kan forskellen mellem x -værdierne være forskellig.

Kategori er en liste, der indeholder kategorikoder for X og Y data..

Medtag er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Output af **SinReg** er altid i radianer, uanset vinkelindstillingen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot \sin(bx+c)+d$
stat.a, stat.b, stat.c, stat.d	Regressionskoefficienter
stat.Resid	Residualer fra regressionen
stat.XReg	Liste af datapunkter i den modificerede X -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede Y -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

solvee(*Ligning, Var*) $\Rightarrow$ Boolesk udtryk

solve(*Ligning, Var=Gæt*) $\Rightarrow$ Boolesk udtryk

solve(*Ulighed, Var*) $\Rightarrow$ Boolesk udtryk

Returnerer reelle løsninger til en ligning eller ulighed for *Var*. Målet er at vise alle løsninger. Der kan dog være ligninger eller uligheder, for hvilke, antallet af løsninger er uendeligt.

Løsninger kan være ikke-reelle for visse kombinationer af værdier for udefinerede variable.

For Auto-indstillingen af

Auto eller tilnærmet-tilstanden er målet at producere eksakte løsninger, når de er koncise, og suppleret af iterative søgninger med approksimerende aritmetik, når nøjagtige løsninger er upraktiske

På grund standardforkortelse med den største fælles divisor af tæller og nævner i brøker, kan løsninger kun være ensidige løsninger fra en eller to sider.

For uligheder af typerne $\geq$, $\leq$, $<$, eller $>$, er eksplicitte løsninger usandsynlige, medmindre uligheden er lineær og kun indeholder *Var*.

For Eksakt-indstillingen af

Auto eller tilnærmet-tilstanden returneres dele, der ikke kan løses om en implicit ligning eller ulighed.

Brug operatoren ("|") til at afgrænse løsningsintervallet og/eller andre variable, der optræder i ligningen eller uligheden. Når du finder en løsning i et interval, kan du anvende ulighedsoperatorene til at udelukke dette interval fra efterfølgende søgninger.

$$\text{solve}(a \cdot x^2 + b \cdot x + c = 0, x)$$

$$x = \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a} \text{ or } x = \frac{-\left(\sqrt{b^2 - 4 \cdot a \cdot c} + b\right)}{2 \cdot a}$$

$$\text{Ans}|a=1 \text{ and } b=1 \text{ and } c=1$$

$$x = \frac{-1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ or } x = \frac{-1}{2} - \frac{\sqrt{3}}{2} \cdot i$$

$$\text{solve}((x-a) \cdot e^x = x \cdot (x-a), x)$$

$$x=a \text{ or } x=-0.567143$$

$$(x+1) \cdot \frac{x-1}{x-1} + x-3 \quad 2 \cdot x-2$$

$$\text{solve}(5 \cdot x-2 \geq 2 \cdot x, x) \quad x \geq \frac{2}{3}$$

$$\text{exact}(\text{solve}((x-a) \cdot e^x = x \cdot (x-a), x))$$

$$e^x + x = 0 \text{ or } x = a$$

I vinkeltilstanden Radian:

$$\text{solve}\left(\tan(x) = \frac{1}{x}, x\right) | x > 0 \text{ and } x < 1$$

$$x = 0.860334$$

False returneres, når der ikke bliver fundet nogen reelle løsninger. True returneres, hvis **solve()** kan bestemme, at enhver endelig reel værdi af *var* opfylder ligningen eller uligheden.

Da **solve()** altid returnerer et Boolsk resultat, kan du anvende "and", "or," og "not" til at kombinere resultaterne fra **solve()** med hinanden eller med andre Boolske udtryk.

Løsningerne kan indeholde en unik, ny udefineret variabel af formen *nj*, hvor *j* er et heltal i intervallet 1–255. Sådanne variable angiver et arbitrært heltal.

I Reel tilstand beregnes kun den reelle gren af brøkpotenser med ulige nævnere. Ved flertydige udtryk, som fx brøkpotenser, logaritmer og inverse trigonometriske funktioner, beregnes kun hovedgrenen. Derfor giver **solve()** kun løsninger, der svarer til den ene reelle gren eller hovedgrenen.

Bemærk: Se også **cSolve()**, **cZeros()**, **nSolve()** og **zeros()**.

solve(*Lign1* and *Lign2* [and...],
VarEllerGæt1, *VarEllerGæt2* [, ...
]) ⇒ *Boolesk* - udtryk

solve(*SystemAfLign*, *VarEllerGæt1*,
VarEllerGæt2 [, ...]) ⇒ *Boolesk* - udtryk

solve({*Lign1*, *Lign2* [...]}
{*VarEllerGæt1*, *VarEllerGæt2* [, ...
]}) ⇒ *Boolesk* - udtryk

Returnerer mulige reelle løsninger til de sammenhørende algebraiske ligninger, hvor hvert *VarEllerGæt* angiver en variabel, du vil løse for.

$\text{solve}(x=x+1,x)$	false
$\text{solve}(x=x,x)$	true

$2 \cdot x - 1 \leq 1$ and $\text{solve}(x^2 \neq 9, x)$	$x \neq -3$ and $x \leq 1$
--	----------------------------

I vinkeltilstanden Radian:

$\text{solve}(\sin(x)=0,x)$	$x=n1 \cdot \pi$
-----------------------------	------------------

$\text{solve}\left(x^{\frac{1}{3}}=1,x\right)$	$x=-1$
$\text{solve}(\sqrt{x}=-2,x)$	false
$\text{solve}(-\sqrt{x}=-2,x)$	$x=4$

$\text{solve}\left(\sqrt{y}=x^2-2 \text{ and } x+2 \cdot y=1, \{x,y\}\right)$
$x=-\frac{3}{2} \text{ and } y=\frac{1}{4} \text{ or } x=1 \text{ and } y=-1$

Du kan adskille ligningerne med **and**-operatoren, eller du kan indtaste et *SystemAlign* med en skabelon fra Katalog. Antallet af *VarEllerGæt*-argumenter skal svare til antallet af ligninger. Du kan også vælge at angive et initielt gæt til en variabel.

Hvert *VarEllerGæt* skal have formen:

variabel

- eller -

variabel = reel eller ikke-reelt tal

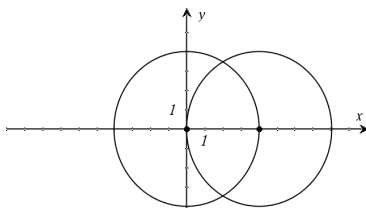
For eksempel er x gyldig, og det er $x=3$ også.

Hvis alle ligningerne er polynomier, og hvis du IKKE angiver nogle initielle gæt, benytter **solve()** den leksikale Gröbner/Buchberger eliminationsmetode som forsøg på at bestemme alle reelle løsninger.

Antag for eksempel, at du har en cirkel med radius r og centrum i origo og en anden cirkel med radius r og centrum, hvor den første cirkel skærer den positive x -akse. Anvend **solve()** til at finde skæringspunkterne.

Som illustreret ved r i eksemplet til højre kan sammenhørende polynomielle ligninger have ekstra variable, der ikke har nogen værdier, men repræsenterer givne numeriske værdier, der kan erstattes senere.

Du kan også (eller i stedet) medtage løsningsvariable, der ikke optræder i ligningerne. For eksempel kan du medtage z som en løsningsvariabel for at udvide det foregående eksempel til to parallelle skærende cylindre med radius r .



$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ or } x=\frac{r}{2} \text{ and } y=-\frac{\sqrt{3}\cdot r}{2}$$

$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y,z\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ and } z=c1 \text{ or } x=\frac{r}{2} \text{ and } y=-\frac{\sqrt{3}\cdot r}{2} \text{ and } z=c1$$

Du kan se hele resultatet ved at trykke på **▲** og derefter bruge **◀** og **▶** til at bevæge markøren.

Cylinderløsningerne illustrerer, hvordan familier af løsninger kan indeholde arbitrære konstanter af formen κ , hvor κ er et heltalssuffiks fra 1 til og med 255

For polynomielle systemer afhænger beregningstiden eller hukommelsesforbrug stærkt af den rækkefølge løsningsvariablene angives i. Hvis det initielle valg kræver for meget hukommelse eller tålmodighed, skal du prøve at omarrangere variablene i ligningerne og/eller `varEllerGæt` listen.

Hvis du ikke medtager nogen gæt, og hvis en ligning er ikke-polynomiell i en variabel, men alle ligninger er lineære i løsningsvariablene, anvender **solve()** en Gauss-eliminering i et forsøg på at bestemme alle reelle løsninger.

Hvis et system hverken er polynomielt i alle variable eller lineært i sine løsningsvariable, bestemmer **solve()** højst en løsning med en iterativ approksimationsmetode. Dette gøres ved at lade antallet af løsningsvariable være lig med antallet af ligninger og reducere alle andre variable i ligningerne til tal.

Hver løsningsvariabel starter ved sin gættede værdi, hvis der findes en; Ellers starter den ved 0.0.

Anvend gæt til at søge flere løsninger, én ad gangen. For at opnå konvergens skal et gæt være meget tæt på en løsning.

$$\text{solve}\left(x + e^z \cdot y = 1 \text{ and } x - y = \sin(z), \{x, y\}\right)$$

$$x = \frac{e^z \cdot \sin(z) + 1}{e^z + 1} \text{ and } y = \frac{-\sin(z) - 1}{e^z + 1}$$

$$\text{solve}\left(e^z \cdot y = 1 \text{ and } y = \sin(z), \{y, z\}\right)$$

$$y = 2.812\text{E-}10 \text{ and } z = 21.9911 \text{ or } y = 0.001871$$

Du kan se hele resultatet ved at trykke på ► og derefter bruge ◀ og ▶ til at bevæge markøren.

$$\text{solve}\left(e^z \cdot y = 1 \text{ and } y = \sin(z), \{y, z = 2 \cdot \pi\}\right)$$

$$y = 0.001871 \text{ and } z = 6.28131$$

SortA

SortA *Liste1* [, *Liste2*] [, *Liste3*] ...

SortA *Vektor1* [, *Vektor2*] [, *Vektor3*] ...

Sorterer elementerne i første argument i stigende rækkefølge.

$\{2, 1, 4, 3\} \rightarrow \text{list1}$	$\{2, 1, 4, 3\}$
SortA list1	Done
list1	$\{1, 2, 3, 4\}$
$\{4, 3, 2, 1\} \rightarrow \text{list2}$	$\{4, 3, 2, 1\}$
SortA list2, list1	Done
list2	$\{1, 2, 3, 4\}$
list1	$\{4, 3, 2, 1\}$

Hvis du medtager yderligere argumenter, sorteres elementerne i hvert argument således, at deres nye positioner passer til de nye positioner for elementerne i det første argument.

Alle argumenter skal være navne på lister eller vektorer. Alle argumenterne skal have ens dimensioner.

tomme (ugyldige) elementer i det første argument flyttes til bunden. Yderligere oplysninger om tomme elementer findes på side 266.

SortD

SortD *Liste1* [, *Liste2*] [, *Liste3*] ...

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
---------------------------------	---------------

SortD *Vektor1* [, *Vektor2*] [, *Vektor3*] ...

$\{1,2,3,4\} \rightarrow list2$	$\{1,2,3,4\}$
---------------------------------	---------------

Identisk med **SortA**, med den undtagelse, at **SortD** sorterer elementerne i faldende rækkefølge.

SortD <i>list1</i> , <i>list2</i>	Done
-----------------------------------	------

<i>list1</i>	$\{4,3,2,1\}$
--------------	---------------

<i>list2</i>	$\{3,4,1,2\}$
--------------	---------------

tomme (ugyldige) elementer i det første argument flyttes til bunden. Yderligere oplysninger om tomme elementer findes på side 266.

Sphere

Vektor ▶ **Sphere**

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>**Sphere**.

Viser række- eller kolonnevektoren i sfærisk form [$p \angle \theta \angle \phi$].

Vektor skal have dimensionen 3 og kan være enten en række- eller kolonnevektor.

Bemærk: Sådan gennemtvinges et tilnærmet resultat,

Håndholdt: Tryk på  .

Windows®: Tryk **Ctrl+Enter**.

Macintosh®: Tryk på  + **Enter**.

iPad®: Hold **ENTER** nede, og vælg .

$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \text{▶ Sphere}$ $\begin{bmatrix} 3.74166 & \angle 1.10715 & \angle 0.640522 \end{bmatrix}$
--

Bemærk: ►Sphere er en displayformat-kommando, ikke en konverteringsfunktion. Du kan kun anvende den i slutningen af en indtastningslinje.

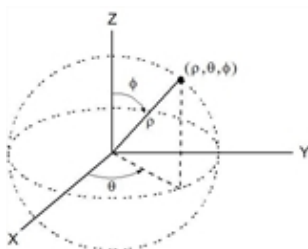
$$\left(\begin{array}{c} 2 \quad \angle \frac{\pi}{4} \quad 3 \end{array} \right) \text{►Sphere}$$

$$\left[3.60555 \quad \angle 0.785398 \quad \angle 0.588003 \right]$$

Tryk 

$$\left(\begin{array}{c} 2 \quad \angle \frac{\pi}{4} \quad 3 \end{array} \right) \text{►Sphere}$$

$$\left[\sqrt{13} \quad \angle \frac{\pi}{4} \quad \angle \sin^{-1} \left(\frac{2 \cdot \sqrt{13}}{13} \right) \right]$$



sqrt()

sqrt(*Udtryk*) ⇒ *udtryk*

$$\sqrt{4} \quad 2$$

sqrt(*Liste1*) ⇒ *liste*

$$\sqrt{\{9, a, 4\}} \quad \{3, \sqrt{a}, 2\}$$

Returnerer kvadratroden af argumentet.

For en liste returneres kvadratrødderne af alle elementer i *Liste1*.

Bemærk: Se også **Kvadratrodsskabelon**, side 1.

stat.results

Viser resultater fra en statistikberegning.

Resultaterne vises som en mængde af navn-værdi-par. De viste specifikke navne afhænger af den senest beregnede statistiske funktion eller kommando.

Du kan kopiere et navn eller en værdi og sætte den ind andre steder.

Bemærk: Undgå at definere variable, der anvender samme navne som dem, der anvendes til statistisk analyse. I visse tilfælde kan der opstå en fejl. Variabelnavne, der anvendes til statistisk analyse, vises i nedenstående tabel.

$xlist := \{1,2,3,4,5\}$	$\{1,2,3,4,5\}$
$ylist := \{4,8,11,14,17\}$	$\{4,8,11,14,17\}$

LinRegMx $xlist, ylist, 1$: stat.results

"Title"	"Linear Regression (mx+b)"
"RegEqn"	"m*x+b"
"m"	3.2
"b"	1.2
"r ² "	0.996109
"r"	0.998053
"Resid"	"{...}"

stat.values	"Linear Regression (mx+b)"
	"m*x+b"
	3.2
	1.2
	0.996109
	0.998053
	"{-0.4,0.4,0.2,0,-0.2}"

stat.a	stat.dfDenom	stat.MedianY	stat.Q3X	stat.SSBLOCK
stat.AdjR ²	stat.dfBlock	stat.MEPred	stat.Q3Y	stat.SSCol
stat.b	stat.dfCol	stat.MinX	stat.r	stat.SSX
stat.b0	stat.dfError	stat.MinY	stat.r ²	stat.SSY
stat.b1	stat.dfInteract	stat.MS	stat.RegEqn	stat.SSError
stat.b2	stat.dfReg	stat.MSBlock	stat.Resid	stat.SSInteract
stat.b3	stat.dfNumer	stat.MSCol	stat.ResidTrans	stat.SSReg
stat.b4	stat.dfRow	stat.MSError	stat.ox	stat.SSRow
stat.b5	stat.DW	stat.MSInteract	stat.oy	stat.tList
stat.b6	stat.e	stat.MSReg	stat.ox1	stat.UpperPred
stat.b7	stat.ExpMatrix	stat.MSRow	stat.ox2	stat.UpperVal
stat.b8	stat.F	stat.n	stat.Σx	stat.ȳ
stat.b9	stat.FBlock	stat.ŷ	stat.Σx ²	stat.ȳ1
stat.b10	stat.Fcol	stat.ŷ1	stat.Σxy	stat.ȳ2
stat.bList	stat.FInteract	stat.ŷ2	stat.Σy	stat.ȳDiff
stat.χ ²	stat.FreqReg	stat.ŷDiff	stat.Σy ²	stat.ȳList
stat.c	stat.Frow	stat.PList	stat.s	stat.XReg

stat.CLower	stat.Leverage	stat.PVal	stat.SE	stat.XVal
stat.CLowerList	stat.LowerPred	stat.PValBlock	stat.SEList	stat.XValList
stat.CompList	stat.LowerVal	stat.PValCol	stat.SEPred	stat. $\bar{y}$
stat.CompMatrix	stat.m	stat.PValInteract	stat.sResid	stat. $\hat{y}$
stat.CookDist	stat.MaxX	stat.PValRow	stat.SEslope	stat. $\hat{y}$ List
stat.CUpper	stat.MaxY	stat.Q1X	stat.sp	stat.YReg
stat.CUpperList	stat.ME	stat.Q1Y	stat.SS	
stat.d	stat.MedianX			

Bemærk: Hver gang en liste- & regneark-funktion beregner statistiske resultater, kopierer den "stat."gruppevariable til en "stat#."gruppe, hvor # er et tal der automatisk sammenlignes. Dette gør, at man kan bevare tidligere resultater, mens man udfører flere beregninger.

stat.values

Katalog > 

stat.values

Se eksemplet med `stat.results`.

Viser en matrix med de beregnede værdier for den senest beregnede statistiske funktion eller kommando.

I modsætning til `stat.results` udelader `stat.values` de navne, der er knyttet til værdierne.

Du kan kopiere en værdi og sætte den ind andre steder.

stDevPop()

Katalog > 

`stDevPop(Liste[, Hyppighedsliste])` ⇒ *udtryk*

Returnerer population standardafvigelsen af elementerne i *Liste*.

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

I vinkeltilstanden Radian og tilstanden Auto:

$\text{stDevPop}\{\{a, b, c\}\}$	
$\frac{\sqrt{2 \cdot (a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2)}}{3}$	
$\text{stDevPop}\{\{1, 2, 5, -6, 3, -2\}\}$	$\frac{\sqrt{465}}{6}$
$\text{stDevPop}\{\{1.3, 2.5, -6.4\}, \{3, 2, 5\}\}$	4.11107

Bemærk: *Liste* skal have mindst to elementer. Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 266

stDevPop(*Matrix1* [, *Hyppighedsmatrix*]) $\Rightarrow$ *matrix*

Returnerer en rækkevektor af populationsstandardafvigelser for kolonnerne i *Matrix1*.

Hvert *Hyppighedsmatrix*-element tæller antallet af konsekutive forekomster af det tilsvarende element i *Matrix1*.

Bemærk: *Matrix1* skal have mindst to rækker. Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 266.

$$\begin{array}{l} \text{stDevPop} \left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix} \right) \left[\frac{4\sqrt{6}}{3} \quad \frac{\sqrt{78}}{3} \quad \frac{2\sqrt{6}}{3} \right] \\ \text{stDevPop} \left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix} \right) \\ \left[2.52608 \quad 5.21506 \right] \end{array}$$

stDevSamp()

stDevSamp(*Liste* [, *Hyppighedsliste*]) $\Rightarrow$ *udtryk*

Returnerer stikprøvestandardafvigelsen af elementerne i *liste*.

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

Bemærk: *Liste* skal have mindst to elementer. Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 266

stDevSamp(*Matrix1* [, *Hyppighedsmatrix*]) $\Rightarrow$ *matrix*

Returnerer en rækkevektor af standardafvigelser for målingerne i kolonnerne i *Matrix1*.

Hvert *Hyppighedsmatrix*-element tæller antallet af konsekutive forekomster af det tilsvarende element i *Matrix1*.

$$\begin{array}{l} \text{stDevSamp}(\{a, b, c\}) \\ \frac{\sqrt{3 \cdot (a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2)}}{3} \\ \text{stDevSamp}(\{1, 2, 5, -6, 3, -2\}) \quad \frac{\sqrt{62}}{2} \\ \text{stDevSamp}(\{1.3, 2.5, -6.4\}, \{3, 2, 5\}) \\ 4.33345 \end{array}$$

$$\begin{array}{l} \text{stDevSamp} \left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix} \right) \left[4 \quad \sqrt{13} \quad 2 \right] \\ \text{stDevSamp} \left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix} \right) \\ \left[2.7005 \quad 5.44695 \right] \end{array}$$

Bemærk: *Matrix1* skal have mindst to rækker. Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 266.

Stop

Katalog >

Stop

Programmeringskommando: Afslutter programmet.

Stop er ikke tilladt i funktioner.

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

<i>i</i> :=0	0
Define <i>prog1</i> ()=Prgm	Done
For <i>i</i> ,1,10,1	
If <i>i</i> =5	
Stop	
EndFor	
EndPrgm	
<i>prog1</i> ()	Done
<i>i</i>	5

Store

Se → (store), side 248.

string()

Katalog >

string(*Udtr*)⇒*streng*

Reducerer *Udtr* og returnerer resultatet som en tegnstreng.

string(1.2345)	"1.2345"
string(1+2)	"3"
string(cos(x)+√3)	"cos(x)+√(3)"

subMat()

Katalog >

subMat(*Matrix1* [, *Startrække*] [, *Startkolonne*] [, *Slutrække*] [, *Slutkolonne*]) ⇒*matrix*

Returnerer den angivne delmatrix af *Matrix1*.

Standardindstillinger: *Startrække*=1, *Startkolonne*=1, *Slutrække*=sidste række, *Slutkolonne*=sidste kolonne.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
subMat(<i>m1</i> ,2,1,3,2)	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
subMat(<i>m1</i> ,2,2)	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

sum()

Katalog > **sum(Liste[, Start[, Slut]])** ⇒ *udtryk*Returnerer summen af elementer i *Liste*.*Start* og *Slut* er valgfri. De angiver en serie af elementer.

$\text{sum}\left(\{1,2,3,4,5\}\right)$	15
--	----

$\text{sum}\left(\{a,2\cdot a,3\cdot a\}\right)$	$6\cdot a$
--	------------

$\text{sum}\left(\text{seq}(n,n,1,10)\right)$	55
---	----

$\text{sum}\left(\{1,3,5,7,9\},3\right)$	21
--	----

Alle ugyldige argumenter giver et ugyldigt resultat. Tomme (ugyldige) elementer i *List* ignoreres. Yderligere oplysninger om tomme elementer findes på side 266.

sum(MatrixI[, Start[, Slut]]) ⇒ *matrix*Returnerer en rækkevektor med summerne af elementerne i kolonnerne i *MatrixI*.*Start* og *Slut* er valgfri. De angiver en serie af rækker.

$\text{sum}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}\right)$	$\begin{bmatrix} 5 & 7 & 9 \end{bmatrix}$
---	---

$\text{sum}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right)$	$\begin{bmatrix} 12 & 15 & 18 \end{bmatrix}$
--	--

$\text{sum}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},2,3\right)$	$\begin{bmatrix} 11 & 13 & 15 \end{bmatrix}$
--	--

Alle ugyldige argumenter giver et ugyldigt resultat. Tomme (ugyldige) elementer i *MatrixI* ignoreres. Yderligere oplysninger om tomme elementer findes på side 266.

sumIf()

Katalog > **sumIf(Liste,Kriterie[, SumListe])** ⇒ *værdi*

Returnerer den akkumulerede sum af alle elementerne i *Liste*, der opfylder det angivne *Kriterie*. Du kan også vælge at angive en alternativ liste, *sumListe*, for at levere de elementer, der skal akkumuleres.

Liste kan være et udtryk, en liste eller en matrix. *SumListe* skal, hvis den er angivet, have samme dimensioner som *Liste*.

Kriterie kan være:

$\text{sumIf}\left(\{1,2,e,3,\pi,4,5,6\},2.5<?<4.5\right)$	$e+\pi+7$
--	-----------

$\text{sumIf}\left(\{1,2,3,4\},2<?<5,\{10,20,30,40\}\right)$	70
--	----

- En værdi, et udtryk eller en streng. For eksempel akkumulerer **34** kun de elementer i *Liste*, der reduceres til værdien 34.
- Et Boolsk udtryk, der indeholder symbolet **?** som pladsholder for hvert element. For eksempel akkumulerer **?<10** kun de elementer i *Liste*, der er mindre end 10.

Når et element i *Liste* opfylder *Kriterie*, føjes elementet til den akkumulerede sum. Hvis du inkluderer *sumListe*, tilføjes det tilsvarende element fra *sumListe* til summen i stedet.

I applikationen Lister og regneark kan du anvende et celleområde i stedet for *Liste* og *sumListe*.

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 266.

Bemærk: Se også **countlf()**, side 37.

system(*Udtr1* [, *Udtr2* [, *Udtr3* [, ...]]])

$$\text{solve}\left(\begin{cases} x+y=0 \\ x-y=8 \end{cases}, x, y\right)$$

$$x=4 \text{ and } y=-4$$

system(*Ligning1* [, *Ligning2* [, *Ligning3* [, ...]]])

Returnerer et system af ligninger formateret som en liste. Du kan også oprette et system ved hjælp af en skabelon.

Bemærk: Se også **System af Ligninger**, side 3.

T (transponere)

Katalog >  $MatrixI \mathbf{T} \Rightarrow matrix$

Returnerer den kompleks konjugerede transponerede $MatrixI$.

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @t.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}^T$	$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$
$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^T$	$\begin{bmatrix} a & c \\ b & d \end{bmatrix}$
$\begin{bmatrix} 1+i & 2+i \\ 3+i & 4+i \end{bmatrix}^T$	$\begin{bmatrix} 1-i & 3-i \\ 2-i & 4-i \end{bmatrix}$

tan()

 -tast $\tan(UdtrI) \Rightarrow udtryk$

I vinkeltilstanden Grader:

 $\tan(ListeI) \Rightarrow liste$

$\tan(UdtrI)$ returnerer tangens af argumentet som et udtryk.

$\tan(ListeI)$ returnerer en liste med tangens til alle elementer i $ListeI$.

Bemærk: Argumentet fortolkes som en vinkel i enten grader eller radianer afhængigt af den aktuelle vinkeltilstand. Du kan bruge °, g, eller r til midlertidigt at tilsidesætte den indstillede vinkeltilstand.

I vinkeltilstanden Nygrader:

$\tan\left(\frac{\pi}{4}\right)$	1
$\tan(45)$	1
$\tan(\{0,60,90\})$	$\{0,\sqrt{3},\text{undef}\}$

I vinkeltilstanden Radian:

$\tan\left(\frac{\pi}{4}\right)$	1
$\tan(45^\circ)$	1
$\tan\left(\left\{\pi, \frac{\pi}{3}, \pi, \frac{\pi}{4}\right\}\right)$	$\{0,\sqrt{3},0,1\}$

 $\tan(kvadratMatrixI) \Rightarrow kvadratMatrix$

I vinkeltilstanden Radian:

Returnerer matrixtangens til $kvadratMatrixI$. Dette er ikke det samme som at beregne tangens for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

tan()



KvadratMatrix1 skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

$$\tan \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} -28.2912 & 26.0887 & 11.1142 \\ 12.1171 & -7.83536 & -5.48138 \\ 36.8181 & -32.8063 & -10.4594 \end{bmatrix}$$

$\tan^{-1}()$



$\tan^{-1}(Udtr1) \Rightarrow$ udtryk

I vinkeltilstanden Grader:

$\tan^{-1}(Liste1) \Rightarrow$ list

$$\tan^{-1}(1) \quad 45$$

$\tan^{-1}(Udtr1)$ returnerer vinklen, hvis tangens er *Udtr1*, som et udtryk.

I vinkeltilstanden Nygrader:

$\tan^{-1}(Liste1)$ returnerer en liste med den inverse tangens til hvert element af *Liste1*.

$$\tan^{-1}(1) \quad 50$$

Bemærk: Resultatet returneres som en vinkel i grader eller radianer afhængigt af den aktuelle vinkeltilstand.

I vinkeltilstanden Radian:

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arctan (...)**.

$$\tan^{-1}(\{0,0,2,0,5\}) \quad \{0,0,197396,0,463648\}$$

$\tan^{-1}(kvadratMatrix1) \Rightarrow kvadratMatrix$

I vinkeltilstanden Radian:

Returnerer den matrixinverse tangens af *kvadratMatrix1*. Dette er ikke det samme som at beregne den inverse tangens for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

$$\tan^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} -0.083658 & 1.26629 & 0.62263 \\ 0.748539 & 0.630015 & -0.070012 \\ 1.68608 & -1.18244 & 0.455126 \end{bmatrix}$$

KvadratMatrix1 skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

tangentLine()Katalog > **tangentLine(Udtr1,Var,Punkt)⇒udtryk**

$\text{tangentLine}(x^2, x, 1)$	$2 \cdot x - 1$
---------------------------------	-----------------

tangentLine(Udtr1,Var,Punkt)⇒udtryk

$\text{tangentLine}((x-3)^2-4, x=3)$	-4
--------------------------------------	----

Returnerer tangentlinjen til kurven repræsenteret ved *Udtr1* i punktet angivet i *Var=Punkt*.

$\text{tangentLine}\left(x \frac{1}{3}, x=0\right)$	$x=0$
---	-------

Vær sikker på, at den uafhængige variabel ikke er defineret. For eksempel, Hvis $f1(x):=5$ og $x:=3$, så returnerer **tangentLine(f1(x),x,2)** "false."

$\text{tangentLine}(\sqrt{x^2-4}, x=2)$	undef
---	-------

$x:=3; \text{tangentLine}(x^2, x, 1)$	5
---------------------------------------	---

tanh()Katalog > **tanh(Udtr1)⇒udtryk**

$\tanh(1.2)$	0.833655
--------------	----------

tanh(Liste1)⇒liste

$\tanh(\{0, 1\})$	$\{0, \tanh(1)\}$
-------------------	-------------------

tanh(Udtr1) returnerer den hyperbolske tangens til argumentet som et udtryk.

tanh(Liste1) returnerer en liste med den hyperbolske tangens til hvert enkelt element i *Liste1*.

tanh(kvadratMatrix1)⇒kvadratMatrix

Returnerer den matrixhyperbolske tangens af *kvadratMatrix1*. Dette er ikke det samme som at beregne den hyperbolske tangens for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

I vinkeltilstanden Radian:

$\tanh\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$	$\begin{bmatrix} -0.097966 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$
--	---

KvadratMatrix1 skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

tanh⁻¹()Katalog > **tanh⁻¹(Udtr1)⇒udtryk**

I rektangulært komplekst format:

$\tanh^{-1}(0)$	0
-----------------	---

tanh⁻¹(Liste1)⇒liste

$\tanh^{-1}(\{1, 2, 1, 3\})$	$\left\{ \text{undef}, 0.518046 - 1.5708 \cdot i, \frac{\ln(2)}{2} - \frac{\pi}{2} \cdot i \right\}$
------------------------------	--

tanh⁻¹(Udtr1) returnerer den inverse hyperbolske tangens af argumentet som et udtryk.

tanh⁻¹()

Katalog >

tanh⁻¹(Liste1) returnerer en liste med den inverse hyperbolske tangens til alle elementer i *Liste1*.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arctanh(...)**.

tanh⁻¹(kvadratMatrix1)⇒kvadratMatrix

Returnerer den matrixinverse hyperbolske tangens af *kvadratMatrix1*. Dette er ikke det samme som at beregne den inverse hyperbolske tangens for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

KvadratMatrix1 skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

Du kan se hele resultatet ved at trykke på **▲** og derefter bruge **◀** og **▶** til at bevæge markøren.

I vinkeltilstanden Radian og rektangulært komplekst format:

$$\tanh^{-1}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$$
$$\begin{bmatrix} -0.099353+0.164058\cdot i & 0.267834-1.4908 \\ -0.087596-0.725533\cdot i & 0.479679-0.94730 \\ 0.511463-2.08316\cdot i & -0.878563+1.7901 \end{bmatrix}$$

Du kan se hele resultatet ved at trykke på **▲** og derefter bruge **◀** og **▶** til at bevæge markøren.

taylor()

Katalog >

taylor(Udtr1, Var, Orden[, Punkt])⇒udtryk

Returnerer det ønskede Taylor-polynomium. Polynomiet medtager led, som er forskellige fra nul, og hvis grad er et helt tal fra nul til *Orden* i (*Var* minus *Punkt*). **taylor()** returnerer sig selv, hvis der ikke er nogen afkortet potensserie af denne orden, eller hvis den ville kræve negative eksponenter eller brøkeksponenter. Anvend substitutions- og/eller midlertidig multiplikation med en potens af (*Var* minus *Punkt*) for at bestemme mere generelle potensserier.

Punkt er som standard i nul og er ekspansionspunktet.

$$\begin{array}{l} \text{taylor}(e^{\sqrt{x}}, x, 2) \qquad \text{taylor}(e^{\sqrt{x}}, x, 2, 0) \\ \text{taylor}(e^t, t, 4) | t = \sqrt{x} \\ \frac{x^2}{24} + \frac{x^2}{6} + \frac{x}{2} + \sqrt{x} + 1 \\ \text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3\right) \qquad \text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3, 0\right) \\ \text{expand}\left(\frac{\text{taylor}\left(\frac{x}{x \cdot (x-1)}, x, 4\right)}{x}, x\right) \\ -x^3 - x^2 - x - \frac{1}{x} - 1 \end{array}$$

tCdf()

Katalog >

tCdf(nedreGrænse, øvreGrænse, fg)⇒tal
hvis *nedreGrænse* og *øvreGrænse* er tal,
liste hvis *nedreGrænse* og *øvreGrænse* er

lister

Beregner Student-*t* sandsynlighedsfordelingen mellem *nedreGrænse* og *øvreGrænse* for de angivne frihedsgrader *fg*.

For $P(X \leq \text{øvreGrænse})$, sæt *nedreGrænse* = $-\infty$.

tCollect()

tCollect(*Udtr1*) $\Rightarrow$ *udtryk*

Returnerer et udtryk hvor produkter og heltalspotenser af sinus- og cosinus konverteres til en linearkombination af sinus- og cosinus af flere vinkler, vinkelsummer og vinkeldifferenser. Transformation konverterer trigonometriske polynomier til en linearkombination af deres harmoniske svingninger.

Undertiden kan **tCollect()** give det ønskede resultat, når den trigonometriske standardreduktion ikke gør det. **tCollect()** er tilbøjelig til at reversere transformationer udført af **tExpand()**. Nogen gange forenkler det et udtryk at anvende **tExpand()** på et resultat fra **tCollect()**, eller omvendt, i to separate trin.

$$\begin{array}{l} \text{tCollect}(\cos(\alpha)^2) \quad \frac{\cos(2 \cdot \alpha) + 1}{2} \\ \text{tCollect}(\sin(\alpha) \cdot \cos(\beta)) \quad \frac{\sin(\alpha - \beta) + \sin(\alpha + \beta)}{2} \end{array}$$

tExpand

tExpand(*Udtr1*) $\Rightarrow$ *udtryk*

Returnerer et udtryk hvori sinus- og cosinus af multiple vinkler, vinkelsummer og vinkeldifferenser ekspanderes. På grund af identiteten $(\sin(x))^2 + (\cos(x))^2 = 1$ er der mange mulige ækvivalente resultater. Derfor kan et resultat adskille sig fra et resultat vist i andre publikationer.

$$\begin{array}{l} \text{tExpand}(\sin(3 \cdot \phi)) \quad 4 \cdot \sin(\phi) \cdot \cos(\phi)^2 - \sin(\phi) \\ \text{tExpand}(\cos(\alpha - \beta)) \quad \cos(\alpha) \cdot \cos(\beta) + \sin(\alpha) \cdot \sin(\beta) \end{array}$$

Undertiden kan **tExpand()** give det ønskede resultat, når den trigonometriske standardreduktion ikke gør det. **tExpand()** er tilbøjelig til at reversere transformationer udført af **tCollect()**. Nogen gange forenkler det et udtryk at anvende **tCollect()** på et resultat fra **tExpand()**, eller omvendt, i to separate trin.

Bemærk: Skalering i gradtilstand med $\pi/180$ forstyrrer funktionen i **tExpand()**, der genkender ekspanderbare former. For at få de bedste resultater skal **tExpand()** anvendes i Radian-tilstand.

Text

TextpromptStreng[, DispFlag]

Programmeringskommando: Standser programmet midlertidigt og viser tegnstrengen *promptStreng* i en dialogboks.

Når en bruger vælger **OK**, fortsætter programudførelsen.

Det valgfrie argument *flag* kan være et hvilket som helst udtryk.

- Hvis *DispFlag* udelades eller beregnes til **1**, føjes tekstmeddelelsen til Regner-historikken.
- Hvis *DispFlag* udelades eller beregnes til **0**, føjes tekstmeddelelsen ikke til historikken.

Hvis programmet skal bruge et skriftligt svar fra brugeren, henvises til **Request**, side 159 eller **RequestStr**, side 159.

Bemærk: Du kan bruge denne kommando i et brugerdefineret program men ikke i en funktion.

Definer et program, der standser midlertidigt for at vise hvert af fem tilfældige tal i en dialogboks.

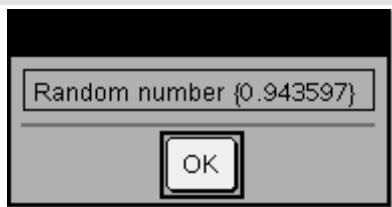
I Prgm...EndPrgm-skabelonen udfyldes hver linje ved at trykke på  i stedet for på **enter**. På computerens tastatur skal du holde **Alt** nede og trykke på **Enter**.

```
Define text_demo()=Prgm
  For i,1,5
    strinfo:="Random number " & string
      (rand(i))
    Text strinfo
  EndFor
EndPrgm
```

Kør programmet:

```
text_demo()
```

Eksempel på en dialogboks:

**tInterval**

tInterval *Liste[,Frekv[,CNiveau]]*

(Datalisteinput)

tInterval $\bar{x}, s_x, n[, CNiveau]$

(Sammenfatning, stat input)

Beregner et *t* konfidensinterval En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval for et ukendt populationsgennemsnit
stat. $\bar{x}$	Middelværdi af stikprøven fra den uniforme fordeling
stat.ME	Fejlmargen
stat.fg	Frihedsgrader
stat. σ_x	Stikprøve standardafvigelse
stat.n	Længde på datasekvensen med målingsgennemsnit

tInterval_2Samp

tInterval_2Samp *Liste1, Liste2
[,Hyppighed1[,Hyppighed2[,CNiveau]*

[,Puljet]]]

(Datalisteinput)

tInterval_2Samp $\bar{x}1, sx1, n1, \bar{x}2, sx2, n2$
[,CNiveau[,Puljet]]

(Sammenfatning, stat input)

Beregner et t konfidensinterval med to målinger. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 187.)

Puljet=1 puljer varianser. *Puljet=0* puljer ikke varianser.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelingssandsynlighed
stat. $\bar{x}1$ - $\bar{x}2$	Middelværdi stikprøver i i datasekvenserne fra den vilkårlige normalfordeling
stat.ME	Fejlmargen
stat.df	Frihedsgrader
stat. $\bar{x}1$, stat. $\bar{x}2$	Middelværdi stikprøver i i datasekvenserne fra den vilkårlige normalfordeling
stat. $\sigma x1$, stat. $\sigma x2$	Stikprøve standardafvigelse for <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Antal målinger i datasekvenserne
stat.sp	Den puljede standardafvigelse. Beregnes, når <i>Puljet</i> = 1.

tmpCnv()

tmpCnv(*Udtr*, °*tempEnhed*, °*tempEnhed2*) ⇒ *udtryk* °*tempEnhed2*

Konverterer en temperaturværdi med *Udtr* mellem enhederne. De gyldige enheder for temperatur er:

_°C Celsius

_°F Fahrenheit

tmpCnv(100·_°C,_°F)	212·_°F
tmpCnv(32·_°F,_°C)	0·_°C
tmpCnv(0·_°C,_°K)	273.15·_°K
tmpCnv(0·_°F,_°R)	459.67·_°R

Bemærk: Med Katalog kan du vælge temperaturenheder.

_°K Kelvin

_°R Rankine

Skriv ° ved at vælge den blandt Katalogsymboler.

Skriv _ ved at trykke på  .

For eksempel konverteres 100_°C til 212_°F.

Et temperaturområde konverteres i stedet med ΔtmpCnv().

ΔtmpCnv()

ΔtmpCnv(Udtryk_°tempEnhed, _°tempEnhed2) ⇒ udtryk _°tempEnhed2

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **deltaTmpCnv (...)**.

Konverterer et temperaturområde (forskellen mellem to temperaturværdier) angivet med *Udtryk* fra en enhed til en anden. De gyldige enheder for temperatur er:

_°C Celsius

_°F Fahrenheit

_°K Kelvin

_°R Rankine

Du kan skrive ° ved at vælge det på symbolpaletten eller skrive @d.

Skriv _ ved at trykke på  .

1_°C og 1_°K har samme størrelse som 1_°F og 1_°R. 1_°C er dog 9/5 større end 1_°F.

For eksempel er et 100_°C område (fra 0_°C til 100_°C) ækvivalent med et område 180_°F-område.

Skriv Δ ved at vælge den blandt Katalogsymboler.

ΔtmpCnv(100_°C,_°F)	180_°F
ΔtmpCnv(180_°F,_°C)	100_°C
ΔtmpCnv(100_°C,_°K)	100_°K
ΔtmpCnv(100_°F,_°R)	100_°R
ΔtmpCnv(1_°C,_°F)	1.8_°F

Bemærk: Med Katalog kan du vælge temperaturenheder.

Δ tmpCnv()

Katalog > 

For at konvertere en bestemt temperaturværdi i stedet for et område skal du anvende **tmpCnv()**.

tPdf()

Katalog > 

tPdf(*XVærdi*,*fg*) $\Rightarrow$ *tal* hvis *XVærdi* er et tal,
liste hvis *XVærdi* er en liste

Beregner tæthedsfunktionen (pdf) for Student-*t* fordelingen ved en angivet *x* værdi med angivne frihedsgrader *fg*.

trace()

Katalog > 

trace(*kvadratMatrix*) $\Rightarrow$ *udtryk*

Returner sporet (sum af alle elementer på hoveddiagonalen) af *kvadratMatrix*.

$\text{trace}\left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}\right)$	15
$\text{trace}\left(\begin{pmatrix} a & 0 \\ 1 & a \end{pmatrix}\right)$	$2 \cdot a$

Try

Katalog > 

Try
blok1
Else
blok2
EndTry

Eksekverer *blok1*, med mindre der opstår en fejl. Programmets eksekvering fortsætter til *blok2*, hvis der opstår en fejl i *blok1*. Systemvariablen *errCode* indeholder fejlkoden, der gør det muligt for programmet at udføre fejlretning. En liste med fejlkoder findes i "Fejlkoder og fejlmeddelelser," side 276.

blok1 og *blok2* kan enten være en enkelt sætning eller en række sætninger adskilt med kolon.

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

Eksempel 2

```
Define prog1()  
  Try  
    z:=z+1  
    Disp "z incremented."  
  Else  
    Disp "Sorry, z undefined."  
  EndTry  
EndPrgm  
  
z:=1:prog1()  
z incremented.  
  
DelVar z:prog1()  
Sorry, z undefined.  
  
Done
```

Define eigenvals(a,b)=Prgm

Du kan se kommandoerne **Try**, **ClrErr** og **PassErr** i funktion ved at indtaste programmet `eigenvals()` vist til højre. Kør programmet ved at eksekvere hvert af de følgende udtryk.

$$\text{eigenvals}\left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix}\right)$$

$$\text{eigenvals}\left(\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$$

© Program `eigenvals(A,B)` viser
eigenverdier for A·B

Try

Disp "A= ",a

Disp "B= ",b

Disp " "

Disp "Eigenverdier for A·B
er:",eigVl(a*b)

Else

If errCode=230 Then

Disp "Error: Produktet af A·B skal
være en kvadratisk matrix"

ClrErr

Else

PassErr

EndIf

EndTry

EndPrgm

Bemærk: Se også **ClrErr**, side 27, og **PassErr**, side 139.

tTest

tTest $\mu_0, \text{Liste}[, \text{Hyppighed}[, \text{Hypot}]]$

(Datalisteinput)

tTest $\mu_0, \bar{x}, s_x, n, [\text{Hypot}]$

(Sammenfatning, stat input)

Udfører en hypotesetest for et enkelt
ubekendt populationsgennemsnit μ , når
populationens standardafvigelse er
ubekendt. En sammenfatning af resultaterne
lagres i variablen *stat.results*. (side 187.)

Test $H_0: \mu = \mu_0$, mod en af de følgende:

Til $H_a: \mu < \mu_0$, sæt *Hypot*<0

til $H_a: \mu \neq \mu_0$ (standard), sæt $Hypot=0$

til $H_a: \mu > \mu_0$, sæt $Hypot>0$

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.t	$(\bar{x} - \mu_0) / (\text{stdafvigelse} / \sqrt{n})$
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.fg	Frihedsgrader
stat. $\bar{x}$	Middelværdi af stikprøver af datasekvensen i <i>Liste</i>
stat.sx	Standardmåleafvigelse for datasekvensen
stat.n	Stikprøvens størrelse

tTest_2Samp

tTest_2Samp *Liste1, Liste2[, Hyppighed1
[, Hyppighed2[, Hypot[, Puljet]]]]*

(Datalisteinput)

tTest_2Samp $\bar{x}1, sx1, n1, \bar{x}2, sx2, n2[, Hypot
[, Puljet]]$

(Sammenfatning, stat input)

Beregner en to-prøvers t test. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 187.)

Test $H_0: \mu_1 = \mu_2$, mod en af de følgende:

Til $H_a: \mu_1 < \mu_2$, sæt $Hypot<0$

til $H_a: \mu_1 \neq \mu_2$ (standard), sæt $Hypot=0$

til $H_a: \mu_1 > \mu_2$, sæt $Hypot>0$

Puljet=1 puljer varianser

Puljet=0 puljer ikke varianser

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.t	Standardnormalværdi beregnet som differens af gennemsnit
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.df	Frihedsgrader for t-statistik
stat. $\bar{x}1$, stat. $\bar{x}2$	Middelværdi af stikprøver for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.sx1, stat.sx2	Stikprøve standardafvigelse for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Størrelse på stikprøverne
stat.sp	Den puljede standardafvigelse. Beregnes, når <i>Puljet=1</i> .

tvmFV()

tvmFV(*N,I,PV,Pmt,[PpY],[CpY],[PmtAt]*) $\Rightarrow$ værdi

tvmFV(120,5,0,-500,12,12) 77641.1

Finansfunktion, der beregner penges fremtidige værdi.

Bemærk: Argumenter, der bruges i TVM-funktionerne er beskrevet i tabellen over TVM-argumenter, side 206. Se også **amortTbl()**, side 8.

tvmI()

tvmI(*N,PV,Pmt,FV,[PpY],[CpY],[PmtAt]*) $\Rightarrow$ værdi

tvmI(240,100000,-1000,0,12,12) 10.5241

Finansfunktion, der beregner den årlige rente.

Bemærk: Argumenter, der bruges i TVM-funktionerne er beskrevet i tabellen over TVM-argumenter, side 206. Se også **amortTbl()**, side 8.

tvmN()

tvmN(*I,PV,Pmt,FV,[PpY],[CpY],[PmtAt]*) $\Rightarrow$ værdi

tvmN(5,0,-500,77641,12,12) 120.

Finansfunktion, der beregner antallet af betalingsperioder.

Bemærk: Argumenter, der bruges i TVM-funktionerne er beskrevet i tabellen over TVM-argumenter, side 206. Se også **amortTbl()**, side 8.

tvmPmt()

tvmPmt(*N,I,PV,FV,[PpY],[CpY],
[PmtAt]*)⇒*værdi*

tvmPmt(60,4,30000,0,12,12) -552.496

Finansfunktion, der beregner beløbet for hver betaling.

Bemærk: Argumenter, der bruges i TVM-funktionerne er beskrevet i tabellen over TVM-argumenter, side 206. Se også **amortTbl()**, side 8.

tvmPV()

tvmPV(*N,I,Pmt,FV,[PpY],[CpY],
[PmtAt]*)⇒*værdi*

tvmPV(48,4,-500,30000,12,12) -3426.7

Finansfunktion, der beregner den aktuelle værdi.

Bemærk: Argumenter, der bruges i TVM-funktionerne er beskrevet i tabellen over TVM-argumenter, side 206. Se også **amortTbl()**, side 8.

TVM-argument*	Beskrivelse	Datatype
N	Antal betalingsperioder	reelt tal
I	Årlig rentesats	reelt tal
PV	Nutidsværdi	reelt tal
Pmt	Betalingsbeløb	reelt tal
FV	Fremtidsværdi	reelt tal
PpY	Betalinger pr år, standardværdi=1	heltal > 0
CpY	Rentetilskrivninger per år, standardværdi=1	heltal > 0

TVM-argument*	Beskrivelse	Datatype
PmtAt	Betaling, der forfalder ved starten af hver periode, standardværdi=slut	heltal (0=slut, 1=start)

* Disse argumentnavne for tidsdiskonterede pengeværdier svarer til til TVM-variabelnavnene (som f.eks. **tvm.pv** og **tvm.pmt**), der anvendes af *Calculator* applikationens FinansRegner. Finansfunktioner gemmer dog ikke deres argumentværdier eller resultater i TVM-variablene.

TwoVar

Katalog > 

TwoVar *X*, *Y*, [*Frekv*] [, *Kategori*, *Medtag*]

Beregner statistik med to variable. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 187.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

X og *Y* er lister med uafhængige og afhængige variable.

Frekv er en valgfri liste med Frekvensværdier. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende *X* og *Y* datapunkt. Standardværdien er 1. Alle elementer skal være heltal ≥ 0 .

Kategori er en liste, der indeholder kategorikoder for *X* og *Y* data..

Medtag er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Et tomt (ugyldigt) element i en af listerne *X*, *Freq* eller *Category* resulterer i at det tilsvarende element i alle disse lister bliver ugyldigt. Et tomt element i en af listerne *X1* til *X20* resulterer i at det tilsvarende element i alle disse lister bliver ugyldigt. Yderligere oplysninger om tomme elementer findes på side 266.

Output-variabel	Beskrivelse
stat. $\bar{x}$	Gennemsnit af x-værdier

Output-variabel	Beskrivelse
stat. x	Summen af x-værdier
stat. x2	Summen af x2-værdier
stat.sx	Standardafvigelse for målingen for x
stat. x	Populations standardafvigelse for x
stat.n	Antal datapunkter
stat. $\bar{y}$	Gennemsnit af y-værdier
stat. y	Summen af y-værdier
stat. y ²	Summen af y2-værdier
stat.sy	Standardafvigelse fra målingen for y
stat. y	Standardafvigelse fra populationen for y
stat. xy	Summen af x · y værdier
stat.r	Korrelationskoefficient
stat.MinX	Minimum af x-værdier
stat.Q ₁ X	1. kvartil af x
stat.MedianX	Median af x
stat.Q ₃ X	3. kvartil af x
stat.MaxX	Maksimum af x-værdier
stat.MinY	Minimum af y-værdier
stat.Q ₁ Y	1. kvartil af y
stat.MedY	Median af y
stat.Q ₃ Y	3. kvartil af y
stat.MaxY	Maksimum af y-værdier
stat. (x-) ²	Summen af kvadraterne på afvigelser fra middelværdien for x
stat. (y-) ²	Summen af kvadraterne på afvigelser fra middelværdien for y

unitV()Katalog > **unitV(Vektor1)**⇒vektor

Returnerer enten en række- eller en kolonneenhedsvektor afhængigt af formen af *Vektor1*.

Vektor1 skal være enten en enkeltrækkematrix eller en enkeltkolonnematrix.

$$\begin{aligned} &\text{unitV}\left(\begin{bmatrix} a & b & c \end{bmatrix}\right) \\ &\left[\frac{a}{\sqrt{a^2+b^2+c^2}} \quad \frac{b}{\sqrt{a^2+b^2+c^2}} \quad \frac{c}{\sqrt{a^2+b^2+c^2}} \right] \\ &\text{unitV}\left(\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}\right) \quad \left[\frac{\sqrt{6}}{6} \quad \frac{\sqrt{6}}{3} \quad \frac{\sqrt{6}}{6} \right] \\ &\text{unitV}\left(\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\right) \quad \left[\frac{\sqrt{14}}{14} \quad \frac{14}{\sqrt{14}} \quad \frac{7}{3\sqrt{14}} \quad \frac{14}{14} \right] \end{aligned}$$

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

unLockCatalog > **unLockVar1** [, *Var2*] [, *Var3*] ...**unLockVar.**

Oplåser de angivne variable eller variabelgrupper. Låste variable kan ikke redigeres eller slettes.

Se **Lock**, side 112 og **getLockInfo()**, side 88.

<i>a</i> :=65	65
Lock <i>a</i>	Done
getLockInfo(<i>a</i>)	1
<i>a</i> :=75	"Error: Variable is locked."
DelVar <i>a</i>	"Error: Variable is locked."
Unlock <i>a</i>	Done
<i>a</i> :=75	75
DelVar <i>a</i>	Done

varPop()Katalog > **varPop(Liste[, hyppighedsliste])**⇒udtryk

Returnerer populationsvariansen af *Liste*.

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

varPop({5,10,15,20,25,30})	875
	12
Ans*1.	72.9167

Bemærk: *Liste* skal indeholde mindst to elementer.

Hvis et element i en af listerne er tom (ugyldig), ignoreres dette element, og det tilsvarende element i den anden liste ignoreres også. Yderligere oplysninger om tomme elementer findes på side 266.

varSamp()

varSamp(*Liste*[,
Hyppighedsliste])⇒*udtryk*

Returnerer stikprøvevariansen for *Liste*.

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

Bemærk: *Liste* skal indeholde mindst to elementer.

Hvis et element i en af listerne er tom (ugyldig), ignoreres dette element, og det tilsvarende element i den anden liste ignoreres også. Yderligere oplysninger om tomme elementer findes på side 266.

varSamp(*Matrix1*[,
Hyppighedsmatrix])⇒*matrix*

Returnerer en rækkevektor med stikprøvevariansen for hver kolonne i *Matrix1*.

Hvert *Hyppighedsmatrix*-element tæller antallet af konsekutive forekomster af det tilsvarende element i *Matrix1*.

Bemærk: *Matrix1* skal indeholde mindst to rækker.

Hvis et element i en af matricerne er tom (ugyldig), ignoreres dette element, og det tilsvarende element i den anden matrix ignoreres også. Yderligere oplysninger om tomme elementer findes på side 266.

$\text{varSamp}(\{a, b, c\})$	
	$\frac{a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2}{3}$
$\text{varSamp}(\{1, 2, 5, -6, 3, -2\})$	$\frac{31}{2}$
$\text{varSamp}(\{1, 3, 5\}, \{4, 6, 2\})$	$\frac{68}{33}$

$\text{varSamp}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{bmatrix}\right)$	$[4.75 \quad 1.03 \quad 4]$
$\text{varSamp}\left(\begin{bmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{bmatrix}, \begin{bmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{bmatrix}\right)$	$[3.91731 \quad 2.08411]$

Wait

katalog >

Wait *tidISekunder*

Afbryder afvikling i en periode på *tidISekunder* sekunder.

Wait er især nyttig i et program, der behøver en kort forsinkelse for at lade ønskede data blive tilgængelige.

Argumentet *tidISekunder* skal være et udtryk, der reduceres til en decimal værdi inden for området 0 til og med 100. Kommandoen runder denne værdi op til det nærmeste 0,1 sekund.

For at annullere en **Wait** der er i gang,

- **Håndholdt:** Hold tasten nede, mens du gentagne gange trykker på .
- **Windows®:** Hold tasten **F12** nede, mens du gentagne gange trykker på **Enter**.
- **Macintosh®:** Hold tasten **F5** nede, mens du gentagne gange trykker på **Enter**.
- **iPad®:** App'en viser en meddelelse. Du kan fortsat vente eller annullere.

Bemærk: kan du bruge kommandoen **Wait** i et brugerdefineret program, men ikke i en funktion.

For at vente 4 sekunder:

Wait 4

For at vente 1/2 sekund:

Wait 0.5

For at vente 1,3 sekund ved hjælp af variablen *seccount*:

seccount:=1.3

Wait seccount

Dette eksempel tænder en grøn LED i 0,5 sekunder, og slukker den så.

Send "SET GREEN 1 ON"

Wait 0.5

Send "SET GREEN 1 OFF"

warnCodes ()

Katalog >

warnCodes(*Expr1*, *StatusVar*)⇒*udtryk*

Beregner udtrykket *Expr1*, returnerer resultater og gemmer koderne fra eventuelle fejl, der er opstået, i listevariablen *StatusVar*. Hvis der ikke genereres fejl, tildeler denne funktion en tom liste til *StatusVar*.

Expr1 kan være et vilkårligt, gyldigt matematisk TI-Nspire™- eller TI-Nspire™ CAS-udtryk. Du kan ikke bruge en kommando eller tildeling som *Expr1*.

The image shows a TI-Nspire calculator screen. At the top, the command `warnCodes(solve(sin(10*x)=x^2/x),warn)` is entered. Below it, the results are displayed: `x=-0.84232 or x=-0.706817 or x=-0.2852`. At the bottom, the variable `warn` is shown with its value: `{10007,10009}`.

Du kan se hele resultatet ved at trykke på og derefter bruge og til at bevæge markøren.

StatusVar skal være et gyldigt variabelnavn.

Se side 284 for at få en oversigt over fejlkoder og tilhørende meddelelser.

when()

when(*Betingelse*, *sandtResultat* [, *falskResultat*][, *ubekendtResultat*])
 $\Rightarrow$ *udtryk*

Returnerer *sandtResultat*, *falskResultat* eller *ubekendtResultat* afhængigt af, om *betingelse* er true, false eller ubekendt. Returnerer inputtet, hvis der er for få argumenter til at angive det korrekte resultat.

Udelad både *falskResultat* og *ubekendtResultat* for at lave et udtryk, der kun er defineret i området, hvor *Betingelse* er true.

Anvend en **undef** *falskResultat* til at definere et udtryk, der kun tegner grafen til et interval.

when() er nyttig til definition af rekursive funktioner.

$\text{when}(x < 0, x + 3)$	$x = 5$	undef
-----------------------------	---------	-------

$\text{when}(n > 0, n \cdot \text{factorial}(n-1), 1)$	$\rightarrow \text{factorial}(n)$	Done
$\text{factorial}(3)$		6
3!		6

While *Betingelse*
Blok
EndWhile

Udfører sætningerne i *Blok*, så længe *Betingelse* er true.

Blok kan enten være en enkelt sætning eller en serie sætninger adskilt med kolon.

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

Define $sum_of_recip(n)$ =Func	
Local $i,tempsum$	
$1 \rightarrow i$	
$0 \rightarrow tempsum$	
While $i \leq n$	
$tempsum + \frac{1}{i} \rightarrow tempsum$	
$i + 1 \rightarrow i$	
EndWhile	
Return $tempsum$	
EndFunc	
	Done
$sum_of_recip(3)$	$\frac{11}{6}$

X

BoolskUdtryk1 **xor** *BoolskUdtryk2*
 returnerer *Boolsk udtryk*

BoolskListe1 **xor** *BoolskListe2* returnerer
Boolsk liste

BoolskMatrix1 **xor** *BoolskMatrix2*
 returnerer *Boolsk matrix*

Returnerer true, hvis *Boolsk Udtr1* er true, og *Boolsk Udtr2* er false eller omvendt.

Hvis du indtaster et decimalt heltal, der er for stort til en 64-bit binær form med fortegn, anvendes en symmetrisk modulo-operation til at bringe værdien ind i det korrekte område. Yderligere oplysninger findes under ►**Base2**, side 18.

Bemærk: Se or, xxxx.

Heltal1 **xor** *Heltal2* $\Rightarrow$ *heltal*

true xor true	false
$5 > 3$ xor $3 > 5$	true

I hexadecimal tilstand:

Vigtigt: Tallet nul, ikke bogstavet O.

0h7AC36 xor 0h3D5F	0h79169
--------------------	---------

I binær tilstand:

Sammenligner to reelle heltal bit for bit med en **xor**-operation. Internt konverteres begge heltal til 64-bit binære tal med fortegn. Når de tilsvarende bits sammenlignes, er resultatet 1, hvis en af bittene (men ikke begge to) er 1. Resultatet er 0, hvis begge bits er 0 eller begge bits er 1. Den returnerede værdi repræsenterer bitresultaterne og vises i overensstemmelse med den valgte talsystemstilstand.

Du kan indtaste heltallene i ethvert talsystem. Til binære eller hexadecimal indtastninger skal du som præfiks benytte henholdsvis 0b eller 0h. Uden præfiks behandles heltallene som decimaltal (10 talsystem).

Hvis du indtaster et decimalt heltal, der er for stort til en 64-bit binær form med fortegn, anvendes en symmetrisk modulooperation til at bringe værdien ind i det korrekte område.

Bemærk: Se **or**, **xxxx**.

0b100101 xor 0b100

0b100001

Bemærk: En binær indtastning kan have op til 64 cifre (præfikset 0b ikke medregnet). En hexadecimal indtastning kan have op til 16 cifre.

Z

zeros()

zeros(*Udtr*, *Var*) ⇒ *liste*

zeros(*Udtr*, *Var*=*Gæt*) ⇒ *liste*

Returnerer en liste med reelle læninger for *Var* som gør *Udtr*=0. **zeros()** gør dette ved beregning **exp**list(**solve**(*Udtr*=0, *Var*), *Var*).

Til visse formål er resultatformatet for **zeros()** mere hensigtsmæssigt end det for **solve()**. Men resultatformatet af **zeros()** kan ikke udtrykke implicitte løsninger, løsninger, der kræver uligheder, eller løsninger, der ikke omfatter *Var*.

Bemærk: Se også **cSolve()**, **cZeros()** og **solve()**.

$$\text{zeros}\left(a \cdot x^2 + b \cdot x + c, x\right) = \left\{ \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a}, -\frac{\left(\sqrt{b^2 - 4 \cdot a \cdot c} + b\right)}{2 \cdot a} \right\}$$

$$a \cdot x^2 + b \cdot x + c | x = \text{Ans}[2] \quad 0$$

$$\text{exact}\left(\text{zeros}\left(a \cdot \left(e^x + x\right) \cdot \left(\text{sign}(x) - 1\right), x\right)\right) \quad \{\cdot\}$$

$$\text{exact}\left(\text{solve}\left(a \cdot \left(e^x + x\right) \cdot \left(\text{sign}(x) - 1\right) = 0, x\right)\right)$$

$$e^x + x = 0 \text{ or } x > 0 \text{ or } a = 0$$

zeros({*Udtr1*, *Udtr2*}, {*VarEllerGæt1*, *VarEllerGæt2* [, ...]}) $\Rightarrow$ *matrix*

Returnerer de mulige reelle nulpunkter for de sammenhørende algebraiske udtryk, hvor hvert *VarEllerGæt* angiver en ubekendt, du søger værdien for.

Du kan også vælge at angive et initielt gæt til en variabel. Hvert *VarEllerGæt* skal have formen:

variabel

– eller –

variabel = reelt eller ikke-reelt tal

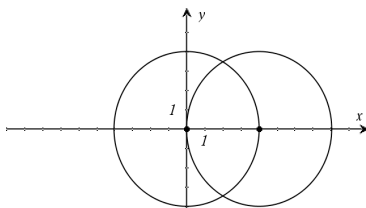
For eksempel er *x* gyldig, og det er *x=3* også.

Hvis alle ligningerne er polynomier, og hvis du IKKE angiver nogle initielle gæt, benytter **zeros()** den leksikale Gröbner/Buchberger eliminationsmetode som forsøg på at bestemme alle komplekse nulpunkter.

Antag for eksempel, at du har en cirkel med radius *r* og centrum i origo og en anden cirkel med radius *r* og centrum, hvor den første cirkel skærer den positive *x*-akse. Anvend **zeros()** til at finde skæringspunkterne.

Som illustreret ved *r* i eksemplet til højre kan sammenhørende polynomielle udtryk have ekstra variable, der ikke har nogen værdier, men repræsenterer givne numeriske værdier, der kan erstattes senere.

Hver række i den resulterende matrix repræsenterer et alternativt nulunkt med komponenterne arrangeret på samme måde som *varEllerGæt*-listen. Du kan udtrække en række ved at indeksere matricen efter [*række*].



$$\text{zeros}\left(\left\{x^2+y^2-r^2, (x-r)^2+y^2-r^2\right\}, \left\{x, y\right\}\right) = \begin{bmatrix} \frac{r}{2} & \frac{-\sqrt{3}\cdot r}{2} \\ \frac{r}{2} & \frac{\sqrt{3}\cdot r}{2} \end{bmatrix}$$

Udtræk række 2:

$$\text{Ans}[2] = \begin{bmatrix} \frac{r}{2} & \frac{\sqrt{3}\cdot r}{2} \end{bmatrix}$$

Du kan også (eller i stedet) medtage ubekendte, der ikke optræder i udtrykkene. For eksempel kan du medtage z som en ubekendt for at udvide det foregående eksempel til to parallelle skærende cylindre med radius r . Cylindrenes nul illustrerer, hvordan familier af nulpunkter kan indeholde arbitrære konstanter af formen $@k$, hvor k er et heltalssuffiks fra 1 til og med 255.

For polynomielle systemer afhænger beregningstiden eller hukommelsesforbruget stærkt af den rækkefølge, de ubekendte angives i. Hvis det initiale valg kræver for meget hukommelse eller tålmodighed, skal du prøve at omarrangere variablene i udtrykkene og/eller *varEllerGæt* listen.

Hvis du ikke medtager nogen gæt, og hvis en ligning er ikke-polynomiell i en variabel, men alle udtryk er lineære i alle ubekendte, anvender **zeros()** en Gauss-eliminering i et forsøg på at bestemme alle nulpunkter.

Hvis et system hverken er polynomielt i alle variable eller lineært i sine ubekendte, bestemmer **zeros()** højst et nul med en iterativ approksimationsmetode. Dette gøres ved at lade antallet af ubekendte være lig med antallet af udtryk og reducere alle andre variable i udtrykkene til tal.

Hver ubekendte starter ved sin gættede værdi, hvis der findes en. Ellers starter den ved 0.0.

Anvend gæt til at søge flere nulpunkter, ét ad gangen. For at opnå konvergens skal et gæt være meget tæt på et nulpunkt.

$$\text{zeros}\left(\left\{x^2+y^2-r^2, (x-r)^2+y^2-r^2\right\}, \{x, y, z\}\right)$$

$$\begin{bmatrix} \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} & c1 \\ \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} & c1 \end{bmatrix}$$

$$\text{zeros}\left(\left\{x+e^z \cdot y-1, x-y-\sin(z)\right\}, \{x, y\}\right)$$

$$\begin{bmatrix} \frac{e^z \cdot \sin(z)+1}{e^z+1} & \frac{-(\sin(z)-1)}{e^z+1} \end{bmatrix}$$

$$\text{zeros}\left(\left\{e^z \cdot y-1, y-\sin(z)\right\}, \{y, z\}\right)$$

0.041458	3.18306
0.001871	6.28131
4.76E-11	1796.99
2.E-13	254.469

$$\text{zeros}\left(\left\{e^z \cdot y-1, y-\sin(z)\right\}, \{y, z=2 \cdot \pi\}\right)$$

0.001871	6.28131
----------	---------

zInterval

zInterval $\sigma, \text{Liste}, \text{Hyppighed}, \text{CNiveau}]$

(DataListeinput)

zInterval $\sigma, \bar{x}, n$ [,CNiveau]

(Sammenfatning, stat input)

Beregner et z konfidensinterval. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 187.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval for en ukendt populationsmiddelværdi
stat. $\bar{x}$	Middelværdi af stikprøven fra den uniforme fordeling
stat.ME	Fejlmargen
stat.sx	Stikprøve standardafvigelse
stat.n	Længde af datasekvens med stikprøvemiddelværdi
stat. σ	Kendt populationsstandardafvigelse for datasekvensen <i>Liste</i>

zInterval_1Prop**zInterval_1Prop** x, n [,CNiveau]

Beregner én-proportion z konfidensinterval. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 187.)

x er et ikke-negativt heltal.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelingssandsynlighed
stat. $\hat{p}$	Den beregnede brøkdel af succeser
stat.ME	Fejlmargen
stat.n	Antal stikprøver i datasekvens

zInterval_2Prop $x1, n1, x2, n2$ [,CNiveau]

Beregner et to-proportion z konfidensinterval. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 187.)

$x1$ og $x2$ er ikke-negative heltal.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelingsandsynlighed
stat. $\hat{p}$ Diff	Den beregnede differens mellem proportioner
stat.ME	Fejlmargen
stat. $\hat{p}1$	Første stikprøves proportionsestimat
stat. $\hat{p}2$	Anden stikprøves proportionsestimat
stat.n1	Stikprøvestørrelsen i datasekvens 1
stat.n2	stikprøvestørrelsen i datasekvens 2

zInterval_2Samp**zInterval_2Samp** σ_1, σ_2 ,Liste1, Liste2
[,Hyppighed1[,Hyppighed2[,CNiveau]]]

(Datalisteinput)

zInterval_2Samp $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2$
[,CNiveau]

(Sammenfatning, stat input)

Beregner et z konfidensinterval med to målinger. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 187.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelings sandsynlighed
stat. $\bar{x}_1$ - $\bar{x}_2$	Middelværdi stikprøver i i datasekvenserne fra den vilkårlige normalfordeling
stat.ME	Fejlmargen
stat. $\bar{x}_1$, stat. $\bar{x}_2$	Middelværdi stikprøver i datasekvenserne fra den vilkårlige normalfordeling
stat. σx_1 , stat. σx_2	Stikprøve standardafvigelse for <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Antal stikprøver i datasekvenserne
stat.r1, stat.r2	Kendt populationsstandardafvigelse for datasekvenserne <i>Liste 1</i> og <i>Liste 2</i>

zTest

Katalog > 

zTest $\mu_0, \sigma, \text{Liste}, [\text{Hyppighed}, \text{Hypot}]$

(Datalisteinput)

zTest $\mu_0, \sigma, \bar{x}, n, [\text{Hypot}]$

(Sammenfatning, stat input)

Udfører en z test med hyppigheden *Hyppighedsliste*. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 187.)

Test $H_0: \mu = \mu_0$, mod en af de følgende:

Til $H_a: \mu < \mu_0$, sæt *Hypot*<0

Til $H_a: \mu \neq \mu_0$ (standard), sæt *Hypot*=0

Til $H_a: \mu > \mu_0$, sæt *Hypot*>0

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.z	$(\bar{x} - \mu_0) / (\sigma / \sqrt{\text{kvrod}(n)})$
stat.P Værdi	Mindste sandsynlighed, ved hvilken nul-hypotesen kan forkastes

Output-variabel	Beskrivelse
stat. $\bar{x}$	Middelværdi af stikprøver af datasekvensen i <i>Liste</i>
stat.sx	Standardmåleafvigelse for datasekvensen. Returneres kun for <i>Datainput</i> .
stat.n	Stikprøvens størrelse

zTest_1Prop

Katalog > 

zTest_1Prop $p0, x, n[, Hypot]$

Beregner en én-proportion z test. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 187.)

x er et ikke-negativt heltal.

Test $H_0: p = p0$ mod en af følgende:

Til $H_a: p > p0$, sæt *Hypot*=0

til $H_a: p \neq p0$ (standard), sæt *Hypot*=0

til $H_a: p < p0$, sæt *Hypot*<0

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.p0	Antaget populationsproportion
stat.z	Standardnormalværdi beregnet for proportionen
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat. $\hat{p}$	Estimeret stikprøveproportion
stat.n	Stikprøvens størrelse

zTest_2Prop

Katalog > 

zTest_2Prop $x1, n1, x2, n2[, Hypot]$

Beregner en toproportional z test. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 187.)

$x1$ og $x2$ er ikke-negative heltal.

Test $H_0: p_1 = p_2$ mod en af følgende:

Til $H_a: p_1 > p_2$, sæt *Hypot*>0

til $H_a: p_1 \neq p_2$ (standard), sæt *Hypot*=0

til $H_a: p_1 < p_2$, sæt *Hypot*<0

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.z	Standardnormalværdi beregnet som differens af proportioner
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat. $\hat{p}_1$	Første stikprøves proportionsestimat
stat. $\hat{p}_2$	Anden stikprøves proportionsestimat
stat. $\hat{p}$	Puljet stikprøves proportionsestimat
stat.n1, stat.n2	Antal stikprøver taget i forsøg 1 og 2

zTest_2Samp

zTest_2Samp $\sigma_1, \sigma_2, \text{Liste1}, \text{Liste2}$
 $[\text{,Hyppighed1}, \text{Hyppighed2}, \text{Hypot}]$

(Datalisteinput)

zTest_2Samp $\sigma_1, \sigma_2, \bar{x}_1, n_1, \bar{x}_2, n_2, \text{Hypot}$

(Sammenfatning, stat input)

Beregner en to-prøvers z test. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 187.)

Test $H_0: \mu_1 = \mu_2$, mod af de følgende:

Til $H_a: \mu_1 < \mu_2$, sæt *Hypot*<0

til $H_a: \mu_1 \neq \mu_2$ (standard), sæt *Hypot*=0

til $H_a: \mu_1 > \mu_2$, *Hypot*>0

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 266.

Output-variabel	Beskrivelse
stat.z	Standardnormalværdi beregnet som differens af gennemsnit
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat. $\bar{x}_1$, stat. $\bar{x}_2$	Middelværdi af stikprøver for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.sx1, stat.sx2	Stikprøve standardafvigelse for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Størrelse på stikprøverne

Symboler

+ (adder)		+ -tast
$Udtr1 + Udtr2 \Rightarrow udtryk$	56	56
Returnerer summen af de to argumenter.	56+4	60
	60+4	64
	64+4	68
	68+4	72
$Liste1 + Liste2 \Rightarrow liste$	$\left\{ 22, \pi, \frac{\pi}{2} \right\} \rightarrow l1$	$\left\{ 22, \pi, \frac{\pi}{2} \right\}$
$Matrix1 + Matrix2 \Rightarrow matrix$	$\left\{ 10, 5, \frac{\pi}{2} \right\} \rightarrow l2$	$\left\{ 10, 5, \frac{\pi}{2} \right\}$
Returnerer en liste (eller matrix), der indeholder summerne af tilsvarende elementer i <i>Liste1</i> og <i>Liste2</i> (eller <i>Matrix1</i> og <i>Matrix2</i>).	$l1+l2$	$\{ 32, \pi+5, \pi \}$
	$Ans+\{ \pi, -5, \pi \}$	$\{ \pi+32, \pi, 0 \}$
	$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} a+1 & b \\ c & d+1 \end{bmatrix}$
Argumenternes dimensioner må være ens.		
$Udtr + Liste1 \Rightarrow liste$	$15+\{ 10, 15, 20 \}$	$\{ 25, 30, 35 \}$
$Liste1 + Udtr \Rightarrow liste$	$\{ 10, 15, 20 \}+15$	$\{ 25, 30, 35 \}$
Returnerer en liste med summerne af <i>Udtr</i> og hvert element i <i>Liste1</i> .		
$Udtr + Matrix1matrix\ Matrix1 + Udtr \Rightarrow matrix$	$20+\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
Returnerer en matrix med <i>Udtr</i> føjet til hvert element på diagonalen af <i>Matrix1</i> . <i>Matrix1</i> skal være kvadratisk.		
Bemærk: Anvend .+ (punktum plus) til at addere et udtryk til hvert element.		
-(subtraher)		- -tast
$Udtr1-Udtr2 \Rightarrow udtryk$	6-2	4
Returnerer <i>Udtr1</i> minus <i>Udtr2</i> .	$\pi-\frac{\pi}{6}$	$\frac{5 \cdot \pi}{6}$
$Liste1-Liste2 \Rightarrow liste$	$\left\{ 22, \pi, \frac{\pi}{2} \right\} - \left\{ 10, 5, \frac{\pi}{2} \right\}$	$\{ 12, \pi-5, 0 \}$
$Matrix1-Matrix2 \Rightarrow matrix$	$\begin{bmatrix} 3 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & 2 \end{bmatrix}$

-(subtraher)

 -tast

Subtraherer hvert element i *Liste2* (eller *Matrix1*) fra det tilsvarende element i *Liste1* (eller *Matrix1*), og returnerer resultaterne.

Argumenternes dimensioner må være ens.

$Udtr - Liste1 \Rightarrow liste$

$$15 - \{10, 15, 20\} \quad \{5, 0, -5\}$$

$Liste1 - Udtr \Rightarrow liste$

$$\{10, 15, 20\} - 15 \quad \{-5, 0, 5\}$$

Subtraherer hvert *Liste1* element fra *Udtr* eller subtraherer *Udtr* fra hvert *Liste1* element og returnerer en liste med resultaterne.

$Udtr - Matrix1 \Rightarrow matrix$

$$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$$

$Matrix1 - Udtr \Rightarrow matrix$

$Udtr - Matrix1$ returnerer en matrix med *Udtr* gange identitetsmatricen minus *Matrix1*. *Matrix1* skal være kvadratisk.

$Matrix1 - Udtr$ returnerer en matrix med *Udtr* gange identitetsmatricen subtraheret fra *Matrix1*. *Matrix1* skal være kvadratisk.

Bemærk: Anvend .- (punktum minus) til at subtrahere et udtryk fra hvert element.

.(multiplicer)

 -tast

$Udtr1 \cdot Udtr2 \Rightarrow udtryk$

$$2 \cdot 3.45 \quad 6.9$$

Returnerer produktet af de to argumenter.

$$x \cdot y \cdot x \quad x^2 \cdot y$$

$Liste1 \cdot Liste2 \Rightarrow liste$

$$\{1, 2, 3\} \cdot \{4, 5, 6\} \quad \{4, 10, 18\}$$

Returnerer en liste med produkterne af de tilsvarende elementer i *Liste1* og *Liste2*.

$$\left\{ \frac{2}{a}, \frac{3}{2} \right\} \cdot \left\{ a^2, \frac{b}{3} \right\} \quad \left\{ 2 \cdot a, \frac{b}{2} \right\}$$

Listernes dimensioner må være ens.

· (multiplicer)

✖-tast

$Matrix1 \cdot Matrix2 \Rightarrow matrix$

Returnerer matrixproduktet af $Matrix1$ og $Matrix2$.

Antallet af kolonner i $Matrix1$ skal være lig med antallet af rækker i $Matrix2$.

$Udtr \cdot Liste1 \Rightarrow liste$

$Liste1 \cdot Udtr \Rightarrow liste$

Returnerer en liste med produkterne af $Udtr$ og hvert element i $Liste1$.

$Udtr \cdot Matrix1 \Rightarrow matrix$

$Matrix1 \cdot Udtr \Rightarrow matrix$

Returnerer en matrix med produkterne af $Udtr$ og hvert element i $Matrix1$.

Bemærk: Anvend $\cdot$ (punktum gangetegn) til at gange et udtryk med hvert element.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix} = \begin{bmatrix} a+2 \cdot b+3 \cdot c & d+2 \cdot e+3 \cdot f \\ 4 \cdot a+5 \cdot b+6 \cdot c & 4 \cdot d+5 \cdot e+6 \cdot f \end{bmatrix}$$

$$\pi \cdot \{4, 5, 6\} = \{4 \cdot \pi, 5 \cdot \pi, 6 \cdot \pi\}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 0.01 = \begin{bmatrix} 0.01 & 0.02 \\ 0.03 & 0.04 \end{bmatrix}$$

$$\lambda \cdot \text{identity}(3) = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}$$

/ (divider)

÷-tast

$Udtr1 / Udtr2 \Rightarrow udtryk$

Returnerer kvotienten af $Udtr1$ divideret med $Udtr2$.

Bemærk: Se også **Brøkskabelon**, side 1.

$Liste1 / Liste2 \Rightarrow liste$

Returnerer en liste med kvotienterne af $Liste1$ divideret med $Liste2$.

Listernes dimensioner må være ens.

$Udtr / Liste1 \Rightarrow liste$

$Liste1 / Udtr \Rightarrow liste$

Returnerer en liste med kvotienterne af $Udtr$ divideret med $Liste1$ eller $Liste1$ divideret med $Udtr$.

$Matrix1 / Udtr \Rightarrow matrix$

Returnerer en matrix med kvotienterne af $Matrix1 / Udtr$.

$$\frac{2}{3.45} = 0.57971$$

$$\frac{x^3}{x} = x^2$$

$$\frac{\{1, 2, 3\}}{\{4, 5, 6\}} = \left\{0.25, \frac{2}{5}, \frac{1}{2}\right\}$$

$$\frac{a}{\{3, a, \sqrt{a}\}} = \left\{\frac{a}{3}, 1, \sqrt{a}\right\}$$

$$\frac{\{a, b, c\}}{a \cdot b \cdot c} = \left\{\frac{1}{b \cdot c}, \frac{1}{a \cdot c}, \frac{1}{a \cdot b}\right\}$$

$$\frac{\begin{bmatrix} a & b & c \end{bmatrix}}{a \cdot b \cdot c} = \begin{bmatrix} \frac{1}{b \cdot c} & \frac{1}{a \cdot c} & \frac{1}{a \cdot b} \end{bmatrix}$$

Bemærk: Anvend $\cdot /$ (punktum divisionstegn) til at dividere et udtryk med hvert element.

$Udtr1 \wedge Udtr2 \Rightarrow \text{udtryk}$

$$\frac{4^2}{\{a, 2, c\}^{\{1, b, 3\}}} \quad \frac{16}{\{a, 2^b, c^3\}}$$

$Liste1 \wedge Liste2 \Rightarrow \text{liste}$

Returnerer det første argument opløftet til potensen af det andet argument.

Bemærk: Se også **Eksponentskabelon**, side 1.

Ved en liste, returneres elementerne i *Liste1* opløftet til potensen af de tilsvarende elementer i *Liste2*.

I det reelle domæne anvender brøkpotenser, der har reducerede eksponenter med ulige nævnere, den reelle gren, i stedet for den principale gren i kompleks tilstand.

$Udtr \wedge Liste1 \Rightarrow \text{liste}$

Returnerer *Udtr* opløftet til potensen af elementerne i *Liste1*.

$$\frac{p^{\{a, 2, -3\}}}{\left\{p^a, p^2, \frac{1}{p^3}\right\}}$$

$Liste1 \wedge Udtr \Rightarrow \text{liste}$

Returnerer elementerne i *Liste1* opløftet til potensen i *Udtr*.

$$\frac{\{1, 2, 3, 4\}^{-2}}{\left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}\right\}}$$

$kvadratMatrix1 \wedge heltal \Rightarrow \text{matrix}$

Returnerer *kvadratMatrix1* opløftet til *heltal*spotensen.

kvadratMatrix1 skal være en kvadratisk matrix.

Hvis *heltal* = -1, beregnes den inverse matrix.

Hvis *heltal* < -1, beregnes den inverse matrix til en passende positiv potens.

$$\begin{array}{l} \left[\begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array} \right]^2 \quad \left[\begin{array}{cc} 7 & 10 \\ 15 & 22 \end{array} \right] \\ \left[\begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array} \right]^{-1} \quad \left[\begin{array}{cc} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{array} \right] \\ \left[\begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array} \right]^{-2} \quad \left[\begin{array}{cc} \frac{11}{2} & -\frac{5}{2} \\ -15 & 7 \\ 4 & 4 \end{array} \right] \end{array}$$

x² (kvadrat) **x^2 -tast***Udtr*¹² ⇒ *udtryk*

$$4^2 \qquad 16$$

Returnerer kvadratet af argumentet.

$$\{2,4,6\}^2 \qquad \{4,16,36\}$$

*Liste*¹² ⇒ *liste*

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2 \qquad \begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$$

Returnerer en liste med kvadraterne på elementerne i *Liste*¹.*kvadratMatrix*¹² ⇒ *matrix*

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix} \wedge 2 \qquad \begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$$

Returnerer matrix i anden potens af *kvadratMatrix*¹. Dette er ikke det samme som at beregne kvadratet på hvert element. Brug $\wedge 2$ til at beregne kvadratet på hvert element.

.+ (punktum plustegn) **$\cdot +$ -taster***Matrix*¹ .+ *Matrix*² ⇒ *matrix*

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot + \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} \qquad \begin{bmatrix} a+c & 6 \\ b+5 & d+3 \end{bmatrix}$$

Udtr .+ *Matrix*¹ ⇒ *matrix*

$$x \cdot + \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} \qquad \begin{bmatrix} x+c & x+4 \\ x+5 & x+d \end{bmatrix}$$

*Matrix*¹ .+ *Matrix*² returnerer en matrix, der er summen af hvert par tilsvarende elementer i *Matrix*¹ og *Matrix*².

Udtr .+ *Matrix*¹ returnerer en matrix, der er summen af *Udtr* og hvert element i *Matrix*¹.

.- (punktum minus.) **$\cdot -$ -taster***Matrix*¹ .- *Matrix*² ⇒ *matrix*

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot - \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix} \qquad \begin{bmatrix} a-c & -2 \\ b-d & -2 \end{bmatrix}$$

Udtr .- *Matrix*¹ ⇒ *matrix*

$$x \cdot - \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix} \qquad \begin{bmatrix} x-c & x-4 \\ x-d & x-5 \end{bmatrix}$$

*Matrix*¹ .- *Matrix*² returnerer en matrix, der er differensen mellem hvert par af tilsvarende elementer i *Matrix*¹ og *Matrix*².

Udtr .- *Matrix*¹ returnerer en matrix, der er differensen af *Udtr* og hvert element i *Matrix*¹.

. · (punktum mult.)

  -taster

$Matrix1 \cdot Matrix2 \Rightarrow matrix$

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} a \cdot c & 8 \\ 5 \cdot b & 3 \cdot d \end{bmatrix}$$

$Udtr \cdot Matrix1 \Rightarrow matrix$

$$x \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a \cdot x & b \cdot x \\ c \cdot x & d \cdot x \end{bmatrix}$$

$Matrix1 \cdot Matrix2$ returnerer en matrix, der er produktet af hvert par tilsvarende elementer i $Matrix1$ og $Matrix2$.

$Udtr \cdot Matrix1$ returnerer en matrix med produkterne af $Udtr$ og hvert element i $Matrix1$.

. / (punktum divider)

  -taster

$Matrix1 \cdot / Matrix2 \Rightarrow matrix$

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot / \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} \frac{a}{c} & \frac{1}{2} \\ \frac{b}{5} & \frac{3}{d} \end{bmatrix}$$

$Udtr \cdot / Matrix1 \Rightarrow matrix$

$Matrix1 \cdot / Matrix2$ returnerer en matrix, der er kvotient af hvert par af tilsvarende elementer i $Matrix1$ og $Matrix2$.

$$x \cdot / \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} \frac{x}{c} & \frac{x}{4} \\ \frac{x}{5} & \frac{x}{d} \end{bmatrix}$$

$Udtr \cdot / Matrix1$ returnerer en matrix, der er kvotienten af $Udtr$ og hvert element i $Matrix1$.

. ^ (punktum potens)

  -taster

$Matrix1 \wedge Matrix2 \Rightarrow matrix$

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \wedge \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} a^c & 16 \\ b^5 & 3^d \end{bmatrix}$$

$Udtr \wedge Matrix1 \Rightarrow matrix$

$Tal \wedge Matrix1 \Rightarrow matrix$

$$x \wedge \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} x^c & x^4 \\ x^5 & x^d \end{bmatrix}$$

$Matrix1 \wedge Matrix2$ returnerer en matrix, hvor hvert element i $Matrix2$ er eksponent til det tilsvarende element i $Matrix1$.

$Udtr \wedge Matrix1$ returnerer en matrix, hvor hvert element i $Matrix1$ er eksponenten til $Udtr$.

-(neger)

 -tast

$-Udtr1 \Rightarrow udtryk$

$-Liste1 \Rightarrow liste$

$-Matrix1 \Rightarrow matrix$

Returnerer negationen til argumentet.

Ved en liste eller matrix, returneres alle elementer negeret.

Hvis argumentet er et binært eller hexadecimalt heltal, giver negationen 2's komplement.

-2.43	-2.43
$-\{-1,0.4,1.2E19\}$	$\{1,-0.4,-1.2E19\}$
$-a \cdot b$	$a \cdot b$

I binær tilstand:

Vigtigt: Tallet nul, ikke bogstavet O

-0b100101
0b11111111111111111111111111111111▶

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

% (procent)

  -taster

$Udtr1 \% \Rightarrow udtryk$

$Liste1 \% \Rightarrow liste$

$Matrix1 \% \Rightarrow matrix$

argument

Returnerer 100

For en liste eller matrix returneres en liste eller matrix med hvert element divideret med 100.

Bemærk: Sådan gennemtvinges et tilnærmet resultat,

Håndholdt: Tryk på  .

Windows®: Tryk **Ctrl+Enter**.

Macintosh®: Tryk på **⌘+Enter**.

iPad®: Hold **ENTER** nede, og vælg .

13%	0.13
-----	------

$\{\{1,10,100\}\}\%$	$\{0.01,0.1,1.\}$
----------------------	-------------------

= (lig med)

 -tast

$Udtr1 = Udtr2 \Rightarrow Boolsk udtryk$

$Liste1 = Liste2 \Rightarrow Boolsk liste$

$Matrix1 = Matrix2 \Rightarrow Boolsk matrix$

Returnerer sand, hvis $Udtr1$ bestemmes til at være lig med $Udtr2$.

Returnerer false, hvis $Udtr1$ bestemmes til at være forskelligt fra $Udtr2$.

Eksempelfunktion, der anvender matematiske testsymboler: =, ≠, <, ≤, >, ≥

= (lig med)

 -tast

Alt andet returnerer en reduceret form af ligningen.

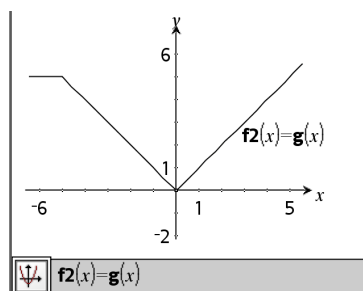
For lister og matricer returneres sammenligninger element for element.

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

```
Define g(x)=Func
  If x<=5 Then
    Return 5
  ElseIf x>5 and x<0 Then
    Return -x
  ElseIf x≥0 and x≠10 Then
    Return x
  ElseIf x=10 Then
    Return 3
  EndIf
EndFunc
```

Done

Tegnet resultat af grafen $g(x)$



≠ (forskellig fra)

  -taster

$Udtr1 \neq Udtr2 \Rightarrow$ Boolsk udtryk

Se "=" (lig med)-eksemplet.

$Liste1 \neq Liste2 \Rightarrow$ Boolsk liste

$Matrix1 \neq Matrix2 \Rightarrow$ Boolsk matrix

Returnerer true, hvis $Udtr1$ bestemmes til at være forskellig fra $Udtr2$.

Returnerer false, hvis $Udtr1$ bestemmes til at være lig med $Udtr2$.

Alt andet returnerer en reduceret form af uligheden.

For lister og matricer returneres sammenligninger element for element.

$\neq$ (forskellig fra)

  -taster

Bemærk: Du kan indsætte denne operator fra tastaturet ved at skrive $\neq$

$<$ (mindre end)

  taster

$Udtr1 < Udtr2 \Rightarrow$ Boolsk udtryk

Se " $=$ " (lig med)-eksemplet.

$Liste1 < Liste2 \Rightarrow$ Boolsk liste

$Matrix1 < Matrix2 \Rightarrow$ Boolsk matrix

Returnerer true, hvis $Udtr1$ bestemmes til at være mindre end $Udtr2$.

Returnerer false, hvis $Udtr1$ bestemmes til at være større end $Udtr2$.

Alt andet returnerer en reduceret form af ligningen.

For lister og matricer returneres sammenligninger element for element.

$\leq$ (mindre end eller lig med)

  taster

$Udtr1 \leq Udtr2 \Rightarrow$ Boolsk udtryk

Se " $=$ " (lig med)-eksemplet.

$Liste1 \leq Liste2 \Rightarrow$ Boolsk liste

$Matrix1 \leq Matrix2 \Rightarrow$ Boolsk matrix

Returnerer true, hvis $Udtr1$ bestemmes til at være mindre end eller lig med $Udtr2$.

Returnerer false, hvis $Udtr1$ bestemmes til at være større end eller lig med $Udtr2$.

Alt andet returnerer en reduceret form af uligheden.

For lister og matricer returneres sammenligninger element for element.

Bemærk: Du kan indsætte denne operator fra tastaturet ved at skrive $\leq$

> (større end)

  taster

$Udtr1 > Udtr2 \Rightarrow \text{Boolsk udtryk}$

Se “=” (lig med)-eksemplet.

$Liste1 > Liste2 \Rightarrow \text{Boolsk liste}$

$Matrix1 > Matrix2 \Rightarrow \text{Boolsk matrix}$

Returnerer true, hvis $Udtr1$ bestemmes til at være større end $Udtr2$.

Returnerer false, hvis $Udtr1$ bestemmes til at være mindre end eller lig med $Udtr2$.

Alt andet returnerer en reduceret form af uligheden.

For lister og matricer returneres sammenligninger element for element.

$\geq$ (større end eller lig med)

  taster

$Udtr1 \geq Udtr2 \Rightarrow \text{Boolsk udtryk}$

Se “=” (lig med)-eksemplet.

$Liste1 \geq Liste2 \Rightarrow \text{Boolsk liste}$

$Matrix1 \geq Matrix2 \Rightarrow \text{Boolsk matrix}$

Returnerer true, hvis $Udtr1$ bestemmes til at være større end eller lig med $Udtr2$.

Returnerer false, hvis $Udtr1$ bestemmes til at være mindre end $Udtr2$.

Alt andet returnerer en reduceret form af uligheden.

For lister og matricer returneres sammenligninger element for element.

Bemærk: Du kan indsætte denne operator fra tastaturet ved at skrive $\geq$

⇒ (medfører)

ctrl **=** -taster

BoolskUdtryk1 ⇒ BoolskUdtryk2
returnerer *Boolsk udtryk*

$5 > 3$ or $3 > 5$	true
--------------------	------

BoolskListe1 ⇒ BoolskListe2
returnerer *Boolsk liste*

$5 > 3 ⇒ 3 > 5$	false
-----------------	-------

3 or 4	7
--------	---

$3 ⇒ 4$	-4
---------	----

BoolskMatrix1 ⇒ BoolskMatrix2
returnerer *Boolsk matrix*

$\{1, 2, 3\}$ or $\{3, 2, 1\}$	$\{3, 2, 3\}$
--------------------------------	---------------

$\{1, 2, 3\} ⇒ \{3, 2, 1\}$	$\{-1, -1, -3\}$
-----------------------------	------------------

Heltal1 ⇒ Heltal2 returnerer *Heltal*

Beregner udtrykket **not** <argument1> **or** <argument2> og returnerer true, false eller en forenklet form af ligningen.

For lister og matricer returneres sandhedsværdier element for element.

Bemærk: Du kan indsætte denne operator fra tastaturet ved at skrive **=>**

⇔ (ensbetydende med)

ctrl **=** -taster

BoolskUdtryk1 ⇔ BoolskUdtryk2
returnerer *Boolsk udtryk*

$5 > 3$ xor $3 > 5$	true
---------------------	------

BoolskListe1 ⇔ BoolskListe2
returnerer *Boolsk liste*

$5 > 3 ⇔ 3 > 5$	false
-----------------	-------

3 xor 4	7
---------	---

$3 ⇔ 4$	-8
---------	----

BoolskMatrix1 ⇔ BoolskMatrix2
returnerer *Boolsk matrix*

$\{1, 2, 3\}$ xor $\{3, 2, 1\}$	$\{2, 0, 2\}$
---------------------------------	---------------

$\{1, 2, 3\} ⇔ \{3, 2, 1\}$	$\{-3, -1, -3\}$
-----------------------------	------------------

Heltal1 ⇔ Heltal2 returnerer *Heltal*

Returnerer negationen af en logisk **XOR** Boolsk operation anvendt på de to argumenter. Returnerer true eller false eller en forenklet form af ligningen.

For lister og matricer returneres sandhedsværdierne element for element.

Bemærk: Du kan indsætte denne operator fra tastaturet ved at skrive **<=>**

! (fakultet)

 -tast

Udtr1! $\Rightarrow$ *udtryk*

5!

120

Liste1! $\Rightarrow$ *liste*

$\{\{5,4,3\}\}!$

$\{120,24,6\}$

Matrix1! $\Rightarrow$ *matrix*

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

$\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$

Returnerer fakultetværdien af argumentet.

For en liste eller matrix returneres en liste eller matrix af elementerne.

& (tilføj)

  -taster

Streng1 & *Streng2* $\Rightarrow$ *streng*

"Hello " & "Nick"

"Hello Nick"

Returnerer en tekststreng, der er *Streng2* adderet til *Streng1*.

d() (differentialkvotient)

Katalog > 

Udtr1, *Var*[,*Orden*]] $\Rightarrow$ *udtryk*

$\frac{d}{dx}(f(x) \cdot g(x))$

$\frac{d}{dx}(f(x)) \cdot g(x) + \frac{d}{dx}(g(x)) \cdot f(x)$

d(*l*, *Var*[,*Orden*]] $\Rightarrow$ *liste*

$\frac{d}{dy}\left(\frac{d}{dx}(x^2 \cdot y^3)\right)$

$6 \cdot y^2 \cdot x$

d(*Matrix1*, *Var*[,*Orden*]] $\Rightarrow$ *matrix*

$\frac{d}{dx}\left(\left\{x^2, x^3, x^4\right\}\right)$

$\left\{2 \cdot x, 3 \cdot x^2, 4 \cdot x^3\right\}$

Returnerer differentialkvotienten af første orden med hensyn til variabelen *Var*.

Orden skal være et heltal, hvis det medtages. Hvis ordenen er mindre end nul, vil resultatet være en stamfunktion.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **derivative (...)**.

d() følger ikke den normale beregningsmekanisme, der reducerer argumenterne fuldt ud, og derefter anvender funktionsdefinitionen på disse fuldt reducerede argumenter. I stedet udfører *d()* følgende trin:

1. Reducerer kun andet argument i den udstrækning, det ikke fører til en ikke-variabel.

2. Reducerer kun første argument i den udstrækning, det ikke henter en lagret værdi for variabelen, der er bestemt af trin 1.
3. Bestemmer den symbolske differentialkvotient af resultatet af trin 2 med hensyn til variabelen fra trin 1.

Hvis variabelen fra trin 1 har en lagret værdi, der er angivet med ("|")-betingelses-operatoren, substitueres denne værdi i resultatet fra trin 3.

Bemærk: Se også **Differentialkvotient af første orden** side 5 **Differentialkvotient af anden orden**, side 6 eller **differentialkvotient af Nte orden**, side 6.

f() (integrer)

$f(Udtr1, Var[, Nedre, Øvre]) \Rightarrow$ udtryk

$f(Udtr1, Var[, Konstant]) \Rightarrow$ udtryk

Returnerer integralet af *Udtr1* med hensyn til variabelen *var* fra *Nedre* til *Øvre*.

Bemærk: Se også **Bestemt** eller **ubestemt integral-skabelonen**, side 6.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **integral (...)**.

Returnerer en stamfunktion, hvis *Nedre* og *Øvre* udelades. En symbolsk integrationskonstant udelades, medmindre du angiver argumentet *Konstant*.

$$\int_a^b x^2 dx \quad \frac{b^3}{3} - \frac{a^3}{3}$$

$$\int x^2 dx \quad \frac{x^3}{3}$$

$$\int (a \cdot x^2, x, c) \quad \frac{a \cdot x^3}{3} + c$$

Stamfunktioner kan adskille sig fra hinanden med en numerisk konstant. En sådan konstant kan være skjult—især når en stamfunktion indeholder logaritmer eller inverse trigonometriske funktioner. Desuden tilføjes stykkevise konstantudtryk nogen gange for at gøre en stamfunktion gyldig over et større interval end den sædvanlige formel.

∫() returnerer sig selv for dele af *Udtr1*, som ikke kan bestemmes som en eksPLICIT endelig kombination af dens indbyggede funktioner og operatorer.

Når du angiver *Nedre* og *Øvre*, forsøges det at finde eventuelle diskontinuiteter eller diskontinuære differentialkvotienter i intervallet *Nedre* < *Var* < *Øvre* og underdele intervallet på disse steder.

For Auto-indstillingen af **Auto eller tilnærmet**-tilstanden anvendes numerisk integration, hvor den er relevant, hvis en stamfunktion eller en grænse ikke kan bestemmes

For Approks.-indstillingen prøves først numerisk integration, hvis det er relevant. Stamfunktioner bestemmes kun, hvor en sådan numerisk integration ikke er relevant eller ikke fungerer.

∫() kan indlejres og dermed klare multiple integraler. Integrationsgrænser kan afhænge af integrationsvariable uden for dem.

Bemærk: Se også **nInt()**, side 130.

$$\int b \cdot e^{-x^2} + \frac{a}{x^2 + a^2} dx \quad b \cdot \int e^{-x^2} dx + \tan^{-1}\left(\frac{x}{a}\right)$$

Bemærk: Sådan gennemtvinges et tilnærmet resultat,

Håndholdt: Tryk på **ctrl** **enter**.

Windows®: Tryk **Ctrl+Enter**.

Macintosh®: Tryk på **⌘+Enter**.

iPad®: Hold **ENTER** nede, og vælg .

$$\int_{-1}^1 e^{-x^2} dx \quad 1.49365$$

$$\int_0^a \int_0^x \ln(x+y) \, dy \, dx \quad \frac{a^2 \cdot \ln(a)}{2} + \frac{a^2 \cdot (4 \cdot \ln(2) - 3)}{4}$$

$\sqrt{()}$ (kvadratrod)

ctrl x²-taster

$\sqrt{()}\Rightarrow$ udtryk

$$\sqrt{4} \qquad 2$$
$$\sqrt{\{9,a,4\}} \qquad \{3,\sqrt{a},2\}$$

Returnerer kvadratroden af argumentet.

For en liste returneres kvadratrødderne af alle elementer i *Liste1*.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **sqrt** (...)

Bemærk: Se også **Kvadratrods-kabelon**, side 1.

$\Pi()$ (prodSeq)

Katalog > 

$\Pi(Udtr1, Var, Lav, Høj)\Rightarrow$ udtryk

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **prodSeq** (...).

Beregner *Udtryk1* for hver værdi af *Var* fra *Lav* til *Høj* og returnerer produktet af resultaterne.

Bemærk: Se også **Produkt-kabelon** (Π), side 5.

$\Pi(Udtr1, Var, Lav, Lav-1)\Rightarrow 1$

$\Pi(Udtr1, Var, Lav, Høj) \Rightarrow 1/\Pi(Udtr1, Var, Høj+1, Lav-1)$ hvis $Høj < Lav-1$

$$\prod_{n=1}^5 \left(\frac{1}{n}\right) \qquad \frac{1}{120}$$
$$\prod_{k=1}^n (k^2) \qquad (n!)^2$$
$$\prod_{n=1}^5 \left\{ \left(\frac{1}{n}, n, 2\right) \right\} \qquad \left\{ \frac{1}{120}, 120, 32 \right\}$$
$$\prod_{k=4}^3 (k) \qquad 1$$

De anvendte produktformler stammer fra følgende reference:

Ronald L. Graham, Donald E. Knuth og Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\prod_{k=4}^1 \left(\frac{1}{k}\right) \qquad 6$$
$$\prod_{k=4}^1 \left(\frac{1}{k}\right) \cdot \prod_{k=2}^4 \left(\frac{1}{k}\right) \qquad \frac{1}{4}$$

$\Sigma()$ (sumSeq)
Katalog > 
 $\Sigma(Udtr1, Var, Lav, Høj) \Rightarrow \text{udtryk}$

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive **sumSeq (...)**.

Beregner *Udtr1* for hver værdi af *Var* fra *Lav* til *Høj* og returnerer summen af resultaterne.

Bemærk: Se også **Sumskabelon**, side 5.

 $\Sigma(Udtr1, Var, Lav, Lav-1) \Rightarrow 0$
 $\Sigma(Udtr1, Var, Lav, Høj) \Rightarrow -\Sigma(Udtr1, Var, Høj+1, Lav-1)$ hvis $Høj < Lav-1$

De anvendte summationsformler stammer fra følgende reference:

Ronald L. Graham, Donald E. Knuth og Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\sum_{n=1}^5 \left(\frac{1}{n} \right) \quad \frac{137}{60}$$

$$\sum_{k=1}^n (k^2) \quad \frac{n \cdot (n+1) \cdot (2 \cdot n+1)}{6}$$

$$\sum_{n=1}^{\infty} \left(\frac{1}{n^2} \right) \quad \frac{\pi^2}{6}$$

$$\sum_{k=4}^3 (k) \quad 0$$

$$\sum_{k=4}^1 (k) \quad -5$$

$$\sum_{k=4}^1 (k) + \sum_{k=2}^4 (k) \quad 4$$

 $\Sigma\text{Int}()$
Katalog > 
 $\Sigma\text{Int}(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [afrundVærdi]) \Rightarrow \text{værdi}$
 $\Sigma\text{Int}(1, 3, 12, 4, 75, 20000, , 12, 12) \quad -218.11$
 ΣInt
 $(NPmt1, NPmt2, amortTabel) \Rightarrow \text{værdi}$

Amortiseringsfunktion, der beregner summen af renter under en angivet række af betalinger.

NPmt1 og *NPmt2* definerer starten og slutningen af betalingsrækken.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen over TVM-argumenter på side 206.

- Hvis du udelader Pmt , sættes den som standard til $Pmt = \text{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$.
- Hvis du udelader FV , sættes den som standard til $FV = 0$.
- Standarderne for PpY , CpY og $PmtAt$ er de samme som for TVM-funktionerne.

afrundVærdi angiver antallet af decimaler til afrunding.
Standardværdi=2.

$\Sigma\text{Int}(NPmt1, NPmt2, \text{amortTabel})$ beregner summen af renten baseret på amortiseringstabellen *amortTabel*. *amortTabel*-argumentet skal være en matrix i formen beskrevet under **amortTbl()**, side 8.

Bemærk: Se også $\Sigma\text{Prn}()$ nedenfor og **Bal()**, side 17.

<i>tbl:=amortTbl(12,12,4.75,20000,,,12,12)</i>			
0	0.	0.	20000.
1	-79.17	-1630.69	18369.3
2	-72.71	-1637.15	16732.2
3	-66.23	-1643.63	15088.5
4	-59.73	-1650.13	13438.4
5	-53.19	-1656.67	11781.7
6	-46.64	-1663.22	10118.5
7	-40.05	-1669.81	8448.7
8	-33.44	-1676.42	6772.28
9	-26.81	-1683.05	5089.23
10	-20.14	-1689.72	3399.51
11	-13.46	-1696.4	1703.11
12	-6.74	-1703.12	-0.01

$\Sigma\text{Int}(1, 3, \text{tbl})$ -218.11

ΣPrn()

$\Sigma\text{Prn}(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [afrundVærdi]) \Rightarrow \text{værdi}$

$\Sigma\text{Prn}(1, 3, 12, 4.75, 20000, \text{tbl})$ -4911.47

$\Sigma\text{Prn}(NPmt1, NPmt2, \text{amortTabel}) \Rightarrow \text{værdi}$

Amortiseringsfunktion, der beregner summen af afdraget under en angivet række af betalinger.

$NPmt1$ og $NPmt2$ definerer starten og slutningen af betalingsrækken.

$N, I, PV, Pmt, FV, PpY, CpY$ og $PmtAt$ er beskrevet i tabellen over TVM-argumenter på side 206.

- Hvis du udelader Pmt , sættes den som standard til $Pmt = \text{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$.
- Hvis du udelader FV , sættes den som standard til $FV = 0$.
- Standardværdierne for PpY , CpY og

<i>tbl:=amortTbl(12,12,4.75,20000,,,12,12)</i>			
0	0.	0.	20000.
1	-79.17	-1630.69	18369.3
2	-72.71	-1637.15	16732.2
3	-66.23	-1643.63	15088.5
4	-59.73	-1650.13	13438.4
5	-53.19	-1656.67	11781.7
6	-46.64	-1663.22	10118.5
7	-40.05	-1669.81	8448.7
8	-33.44	-1676.42	6772.28
9	-26.81	-1683.05	5089.23
10	-20.14	-1689.72	3399.51
11	-13.46	-1696.4	1703.11
12	-6.74	-1703.12	-0.01

$\Sigma\text{Prn}(1, 3, \text{tbl})$ -4911.47

PmtAt er de samme som for TVM-funktionerne.

afrundVærdi angiver antallet af decimaler til afrunding.
Standardværdi=2.

ΣPrn(*NPmt1*,*NPmt2*,*amortTabel*)
beregner summen af afdrag på hovedstolen baseret på amortiseringstabellen *amortTabel*.
amortTabel-argumentet skal være en matrix i formen beskrevet under **amortTbl()**, side 8.

Bemærk: Se også ΣInt() ovenfor og Bal(), side 17.

(henvisning)

  -taster

varNavnestreng

Kalder variablen, hvis navn er *varNavnestreng*. Dermed kan du anvende strenge til at oprette variabelnavne fra en funktion.

#{"x"&"y"&"z"} xyz

Opretter eller kalder variablen xyz.

10 → *r* 10

"1" → *s1* "1"

#*s1* 10

Returnerer værdien af variablen (*r*), hvis navn er lagret i variablen *s1*.

E (videnskabelig notation)

 -tast

mantisseEeksponent

Indtaster et tal i videnskabelig notation. Tallet fortolkes som *mantisse* × 10^{*eksponent*}.

23000. 23000.

2300000000.+4.1E15 4.1E15

3·10⁴ 30000

Tip: Hvis du vil indtaste en 10. potens uden at give et decimalværdiresultat, skal du bruge 10[^]*heltal*.

E (videnskabelig notation)

EE-tast

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive **@E**. Skriv for eksempel **2.3@E4** for at indtaste 2.3E4.

g (nygrader)

1-tast

Udtr1g ⇒ *udtryk*

I vinkeltilstanden Grader, Nygrader eller Radianer:

Udtr1g ⇒ *udtryk*

$$\cos(50^g) \quad \frac{\sqrt{2}}{2}$$

Liste1g ⇒ *liste*

$$\cos(\{0, 100^g, 200^g\}) \quad \{1, 0, -1\}$$

Matrix1g ⇒ *matrix*

Denne funktion giver mulighed for at angive en vinkel i nygrader, mens programmet er i vinkeltilstanden Grader eller Radianer.

I vinkeltilstanden Radian ganges *Udtr1* med $\pi/200$.

I vinkeltilstanden Grader ganges *Udtr1* med $g/100$.

I vinkeltilstanden Nygrader returneres *Udtr1* uændret.

Bemærk: Du kan indsætte dette symbol fra computerens tastatur ved at skrive **@g**.

r (radian)

1-tast

Udtr1r ⇒ *udtryk*

I vinkeltilstanden Grader, Nygrader eller Radianer:

Liste1r ⇒ *liste*

Matrix1r ⇒ *matrix*

$$\cos\left(\frac{\pi}{4^r}\right) \quad \frac{\sqrt{2}}{2}$$

Denne funktion giver mulighed for at angive en vinkel i radianer, mens programmet er i vinkeltilstanden Grader eller Nygrader.

$$\cos\left(\left\{0^r, \frac{\pi}{12}, \left(\frac{\pi}{2}\right)^r\right\}\right) \quad \left\{1, \frac{(\sqrt{3}+1)\sqrt{2}}{4}, -1\right\}$$

ʀ (radian)

1-tast

I vinkeltilstanden Grader ganges argumentet med $180/\pi$.

I vinkeltilstanden Radian, returneres argument uændret.

I vinkeltilstanden Nygrader ganges argumentet med $200/\pi$.

Tip: Anvend ʀ, hvis du vil fremtvinge radianer i en funktionsdefinition uanset den aktuelle tilstand, når funktionen anvendes.

Bemærk: Du kan indsætte dette symbol fra computerens tastatur ved at skrive @r.

° (grader)

1-tast

$Udtr I^{\circ} \Rightarrow udtryk$

$Liste I^{\circ} \Rightarrow liste$

$Matrix I^{\circ} \Rightarrow matrix$

Denne funktion giver mulighed for at angive en vinkel i grader, mens programmet er i vinkeltilstanden Nygrader eller Radianer.

I vinkeltilstanden Radian ganges argumentet med $\pi/180$.

I vinkeltilstanden grader returneres argument uændret.

I vinkeltilstanden Nygrader ganges argumentet med $10/9$.

Bemærk: Du kan indsætte dette symbol fra computerens tastatur ved at skrive @d.

I vinkeltilstanden Grader, Nygrader eller Radianer:

$$\frac{\cos(45^{\circ})}{2} = \frac{\sqrt{2}}{2}$$

I vinkeltilstanden Radianer:

Bemærk: Sådan gennemtvinges et tilnærmet resultat,

Håndholdt: Tryk på **ctrl** **enter**.

Windows®: Tryk **Ctrl+Enter**.

Macintosh®: Tryk på **⌘+Enter**.

iPad®: Hold **ENTER** nede, og vælg $\approx$.

$$\cos\left(\left\{0, \frac{\pi}{4}, 90^{\circ}, 30.12^{\circ}\right\}\right) = \{1, 0.707107, 0, 0.864976\}$$

°, ', " (grader/minutter/sekunder)

ctrl  -taster

$gg^{\circ}mm'ss.ss'' \Rightarrow udtryk$

I vinkeltilstanden Grader:

ggEt positivt eller negativt tal

<i>mm</i> Et ikke-negativt tal	25°13'17.5"	25.2215
<i>ss.ss</i> Et ikke-negativt tal	25°30'	<u>51</u>
		2

Returnerer $gg+(mm/60)+(ss.ss/3600)$.

Med denne indtastning i 60-talsformat kan du:

- Indtaste en vinkel i grader/minutter/sekunder uden hensyn til den aktuelle vinkeltilstand.
- Indtaste tiden som timer/minutter/sekunder.

Bemærk: Efterfølg *ss.ss* med to apostroffer ("), ikke et citationstegn (").

∠ (vinkel)

[*Radius, ∠_Vinkel*] ⇒ *vektor*
(polært input)

[*Radius, ∠_Vinkel, Z_Koordinat*]
⇒ *vektor*
(cylindrisk input)

[*Radius, ∠_Vinkel, ∠_Vinkel*]
⇒ *vektor*
(sfærisk input)

Returnerer koordinater som en vektor afhængigt af tilstandsindstillingen for vektorformat: rektangulær, cylindrisk eller sfærisk.

Bemærk: Du kan indsætte dette symbol fra computerens tastatur ved at skrive $\angle$.

(*Størrelse ∠ Vinkel*) ⇒ *kompleks Værdi*
(polært input)

Indtaster en kompleks værdi i den polære form ($r \angle \theta$). *Vinkel* fortolkes efter den aktuelle indstilling af vinkeltilstand.

I vinkeltilstanden Radian og vektorformatet indstillet til:

rektangulær

$$\left[5 \angle 60^\circ \angle 45^\circ \right] \left[\frac{5 \cdot \sqrt{2}}{4} \quad \frac{5 \cdot \sqrt{6}}{4} \quad \frac{5 \cdot \sqrt{2}}{2} \right]$$

cylindrisk

$$\left[5 \angle 60^\circ \angle 45^\circ \right] \left[\frac{5 \cdot \sqrt{2}}{2} \quad \angle \frac{\pi}{3} \quad \frac{5 \cdot \sqrt{2}}{2} \right]$$

sfærisk

$$\left[5 \angle 60^\circ \angle 45^\circ \right] \left[5 \angle \frac{\pi}{3} \angle \frac{\pi}{4} \right]$$

I vinkeltilstanden radian og rektangulært komplekst format:

$$5+3 \cdot i \left(10 \angle \frac{\pi}{4} \right) \quad 5-5 \cdot \sqrt{2} + (3-5 \cdot \sqrt{2}) \cdot i$$

∠ (vinkel)

-taster

Bemærk: Sådan gennemtvinges et tilnærmet resultat,

Håndholdt: Tryk på .

Windows®: Tryk **Ctrl+Enter**.

Macintosh®: Tryk på **⌘+Enter**.

iPad®: Hold **ENTER** nede, og vælg .

$$5+3\cdot i-\left(10\angle\frac{\pi}{4}\right) \quad -2.07107-4.07107\cdot i$$

' (mærke)

-tast

variabel '

variabel ''

Indsætter et mærketegn i en differentialligning. Et enkelt mærketegn markerer en differentialligning af første orden, to mærketegn markerer anden orden osv.

$$\text{deSolve}\left(y''=y^{\frac{-1}{2}} \text{ and } y(0)=0 \text{ and } y'(0)=0,t,y\right)$$
$$\frac{\frac{3}{2}\cdot y^{\frac{4}{3}}}{3}=t$$

_ (understregning som et tomt element)

Se |“Tomme (ugyldige) elementer” side 266, .

_ (understregning som en enhedsbenævnelse)

-taster

Udtr_Enhed

$$3\cdot \text{m} \rightarrow \text{ft} \quad 9.84252\cdot \text{ft}$$

Angiver enhederne i et *Udtr*. Alle enhedsnavne skal begyndes med en understregning.

Du kan anvende prædefinerede enheder eller oprette dine egne enheder. Du får en liste med prædefinerede enheder ved at åbne Katalog og vise fanen med enhedsomregninger. Du kan vælge enhederne fra Katalog eller skrive enhederne direkte.

Variabel_

Bemærk: Du kan finde konverteringstegnet, , i Katalog. Klik på , og klik derefter på **Matematiske operatører**.

Det antages, at *z* er udefineret:

_ (understregning som en enhedsbenævnelse)

  -taster

Når *Variabel* ikke har nogen værdi, behandles den, som om den repræsenterer et komplekst tal. Uden _ behandles variabelen som reel.

Hvis *Variabel* har en værdi, ignoreres _ og *Variabel* bevarer sin oprindelige datatype.

Bemærk: Du kan lagre et komplekst tal i en variabel uden

at anvende _. Men for det bedste resultat i beregninger som **cSolve()** og **cZeros()** anbefales _.

$\text{real}(z)$	z
$\text{real}(z_)$	$\text{real}(z_)$
$\text{imag}(z)$	0
$\text{imag}(z_)$	$\text{imag}(z_)$

► (konverter)

  -taster

Udtr_Enhed1 ► *_Enhed2* ⇒ *Udtr_Enhed2*

$3 \cdot \text{_m} \text{►} \text{_ft}$ $9.84252 \cdot \text{_ft}$

Konverterer et udtryk mellem enheder.

Understregningstegnet _ benævner enhederne. Enhederne skal være i samme kategori, for eksempel Længde eller Areal.

Du får en liste med prædefinerede enheder ved at åbne Catalog og vise fanen med enhedsomregninger:

- Du kan vælge et enhed fra listen.
- Du kan vælge konverteringsoperatoren ► øverst på listen.

Du kan også skrive enhederne manuelt. For at skrive “_”, når du indtaster enheder på den håndholdte, trykker du

 .

Bemærk: Temperaturenhederne konverteres med **tmpCnv()** og **ΔtmpCnv()**. Konverteringsoperatoren ► håndterer ikke temperaturenheder.

10^()

Katalog >

10^ (Udtr1)⇒udtryk

$$10^{1.5} \quad 31.6228$$

10^ (Liste1)⇒liste

$$10^{\{0, -2, 2, a\}} \quad \left\{ 1, \frac{1}{100}, 100, 10^a \right\}$$

Returnerer 10 opløftet til potensen af argumentet.

I en liste returneres 10 opløftet til potensen af elementerne i *Liste1*.

10^(kvadratMatrix1)⇒kvadratmatrix

$$10^{\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}} \quad \begin{bmatrix} 1.14336\text{E}7 & 8.17155\text{E}6 & 6.67589\text{E}6 \\ 9.95651\text{E}6 & 7.11587\text{E}6 & 5.81342\text{E}6 \\ 7.65298\text{E}6 & 5.46952\text{E}6 & 4.46845\text{E}6 \end{bmatrix}$$

Returnerer 10 opløftet til potensen af *kvadratMatrix1*. Dette er ikke det samme som at beregne 10 opløftet til potensen af hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

KvadratMatrix1 skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

^-1(reciprok)

Katalog >

Udtr1 ^-1⇒udtryk

$$(3.1)^{-1} \quad 0.322581$$

Liste1 ^-1⇒liste

$$\{a, 4, -0.1, x, -2\}^{-1} \quad \left\{ \frac{1}{a}, \frac{1}{4}, -10, \frac{1}{x}, \frac{1}{2} \right\}$$

Returnerer den reciprokke værdi af argumentet.

For en liste returneres de reciprokke værdier af elementerne i *Liste1*.

kvadratMatrix1 ^-1⇒kvadratMatrix

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ a & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} \frac{-2}{a-2} & \frac{1}{a-2} \\ \frac{a}{2 \cdot (a-2)} & \frac{-1}{2 \cdot (a-2)} \end{bmatrix}$$

Returnerer den inverse værdi af *kvadratMatrix1*.

kvadratMatrix1 skal være en ikke-singulær kvadratisk matrix.

| (betingelses-operator)

-taster

Udtryk | BoolskUdtryk1 [andBoolskUdtryk2]...

$$x+1|x=3 \quad 4$$

$$x+y|x=\sin(y) \quad \sin(y)+y$$

Udtryk | BoolskUdtryk1 [orBoolskUdtryk2]...

$$x+y|\sin(y)=x \quad x+y$$

("|")-betingelses-tegnet fungerer som en binær operator. Argumentet til venstre for | er et udtryk. Argumentet til højre for | angiver en eller flere betingelser, der skal tages hensyn til i reduktionen af udtrykket. Hvis der er flere betingelser efter |, skal de samles med en logisk "and" eller "or"-kommando.

Betingelses-operatoren bruges på tre grundlæggende måder:

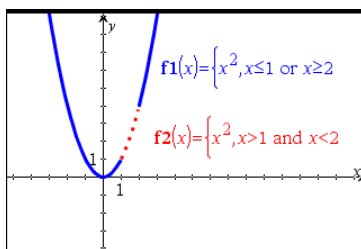
- Substitutioner
- Intervalafgrænsninger
- Udelukkelse

Substitutioner er i form af en ligning, som $x=3$ eller $y=\sin(x)$. For at være så effektiv som muligt skal venstre side af betingelsen være en simpel variabel. *Udtryk | Variabel = værdi* vil substituere *værdi* for enhver forekomst af *Variabel* i *Udtryk*.

Intervalafgrænsning antager form af en eller flere uligheder samlet af logiske "and eller" or-kommandoer. Intervalafgrænsninger tillader også reduktioner, der ellers ville være ugyldige eller ikke kunne beregnes.

$x^3-2\cdot x+7 \rightarrow f(x)$	Done
$f(x) _{x=\sqrt{3}}$	$\sqrt{3}+7$
$(\sin(x))^2+2\cdot \sin(x)-6 \sin(x)=d$	$d^2+2\cdot d-6$

$\text{solve}(x^2-1=0,x) _{x>0 \text{ and } x<2}$	$x=1$
$\sqrt{x}\cdot \sqrt{\frac{1}{x}} _{x>0}$	1
$\sqrt{x}\cdot \sqrt{\frac{1}{x}}$	$\sqrt{\frac{1}{x}}\cdot \sqrt{x}$



| (betingelses-operator)

ctrl  -tast

Udelukkelser benytter "forskellig fra"-kommandoen (/= eller $\neq$) til at udelukke en specifik værdi fra reduktionen. De anvendes primært til at udelukke en eksakt løsning ved anvendelse af **cSolve()**, **cZeros()**, **fMax()**, **fMin()**, **solve()**, **zeros()** osv.

$$\text{solve}(x^2 - 1 = 0, x) | x \neq 1 \quad x = -1$$

→ (lagring)

ctrl  var -tast

Udtr → *Var*

$$\frac{\pi}{4} \rightarrow \text{myvar} \quad \frac{\pi}{4}$$

Liste → *Var*

$$2 \cdot \cos(x) \rightarrow yI(x) \quad \text{Done}$$

Matrix → *Var*

$$\{1, 2, 3, 4\} \rightarrow \text{lst5} \quad \{1, 2, 3, 4\}$$

Udtr → *Funktion*(*Param1*, ...)

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow \text{matg} \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

List → *Funktion*(*Param1*, ...)

$$\text{"Hello"} \rightarrow \text{str1} \quad \text{"Hello"}$$

Matrix → *Funktion*(*Param1*, ...)

Hvis variablen *Var* ikke findes, oprettes *Var* og initialiseres til *Udtr*, *Liste* eller *Matrix*.

Hvis *Var* findes i forvejen og ikke er låst eller beskyttet, erstattes dens indhold med *Udtr*, *Liste* eller *Matrix*.

Tip: Hvis du vil udføre symbolske beregninger med udefinerede variable, skal du undgå at lagre noget i almindeligt brugte et-bogstavsvariable som a, b, c, x, y, z osv.

Bemærk: Du kan indsætte denne operator fra tastaturet ved at skrive =: som en genvej. Skriv for eksempel **pi/4 =: myvar**.

:= (tildel)

ctrl  -taster

Var := *Udtr*

Var := *Liste*

Var := *Matrix*

Funktion(*Param1*,...) := *Udtr*

Funktion(*Param1*,...) := *Liste*

Funktion(*Param1*,...) := *Matrix*

Hvis variablen *Var* ikke findes, oprettes *Var* og initialiseres til *Udtr*, *Liste* eller *Matrix*.

Hvis *Var* findes i forvejen og ikke er låst eller beskyttet, erstattes dens indhold med *Udtr*, *Liste* eller *Matrix*.

Tip: Hvis du vil udføre symbolske beregninger med udefinerede variable, skal du undgå at lagre noget i almindeligt brugte et-bogstavsvariable som a, b, c, x, y, z osv.

$myvar := \frac{\pi}{4}$	$\frac{\pi}{4}$
$y1(x) := 2 \cdot \cos(x)$	Done
$lst5 := \{1, 2, 3, 4\}$	$\{1, 2, 3, 4\}$
$matg := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
$str1 := \text{"Hello"}$	"Hello"

© (kommentar)

ctrl  -taster

© [*tekst*]

© behandler *tekst* som en kommentarlinje, så du kan skrive kommentarer til funktioner og programmer, du opretter.

© kan være i starten eller overalt i linjen. Alt til højre for © til linjens slutning er kommentaren.

Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

Define $g(n) = \text{Func}$	
© Declare variables	
Local <i>i</i> , <i>result</i>	
<i>result</i> := 0	
For <i>i</i> , 1, <i>n</i> , 1 © Loop <i>n</i> times	
<i>result</i> := <i>result</i> + i^2	
EndFor	
Return <i>result</i>	
EndFunc	
	Done
$g(3)$	14

Ob, Oh

  -taster,   -taster

Ob *binærtTal*

I decimal tilstand:

0h hexadecimalTal

0b10+0hF+10	27
-------------	----

Betegner henholdsvis binært eller hexadecimalt tal. For at indtaste et binært eller hexadecimalt tal skal du indtaste præfikset 0b eller 0h uanset talsystemet. Uden præfiks behandles tallet som decimaltal (10 talssystem).

I binær tilstand:

0b10+0hF+10	0b11011
-------------	---------

Resultatet vises i den valgte tilstand for talsystem.

I hexadecimal tilstand:

0b10+0hF+10	0h1B
-------------	------

TI-Nspire™ CX II - Tegn kommandoer

Dette er et supplerende dokument til TI-Nspire™ Referencevejledningen og TI-Nspire™ CAS Referencevejledningen. Alle TI-Nspire™ CX II-kommandoer vil blive inkorporerede og offentliggjort i version 5.1 i TI-Nspire™ Referencevejledningen og TI-Nspire™ CAS Referencevejledningen.

Grafikprogrammering

Nye kommandoer er blevet tilføjet på TI-Nspire™ CX II håndholdte og TI-Nspire™ desktop applikationer til grafisk programmering.

TI-Nspire™ CX II håndholdte skifter til denne grafiktilstand, mens du udfører grafikkommandoer og skifter tilbage til den kontekst, hvor programmet blev udført efter programmets afslutning.

Skærmen viser "Running ..." i øverste linje mens programmet udføres. Det vil vise "Finished", når programmet fuldendes. Enhver tastetryk vil tage systemet ud af grafikfunktionen.

- Overgangen til grafiktilstand udløses automatisk, når en af Draw (grafik) - kommandoerne opstår under udførelsen af TI-Basic-programmet.
- Denne overgang vil kun ske, når der udføres et program fra lommeregneren; i et dokument eller en lommeregner i scratchpad.
- Overgangen fra grafiktilstand sker ved afslutning af programmet.
- Grafiktilstanden er kun tilgængelig på visninger på TI-Nspire™ CX II håndholdte og TI-Nspire™ CX II håndholdte. Det betyder, at den ikke er tilgængelig i computerdokumentvisningen på skrivebordet eller iOS.
 - Hvis der opstår en grafikkommando under udførelsen af et TI-Basic-program fra den forkerte kontekst, vises en fejlmeddelelse, og TI-Basic-programmet afsluttes.

Grafikskærm

Grafikskærmen vil indeholde en overskrift øverst på skærmen, der ikke kan skrives til af grafikkommandoer.

Grafikskærmens tegningsområde bliver ryddet (farve = 255.255.255), når grafikskærmen initialiseres.

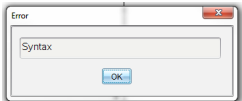
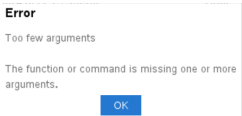
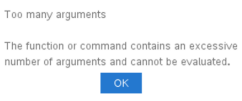
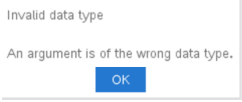
Grafikskærm	Standard
Højde	212
Bredde	318
Farve	hvid: 255,255,255

Standardvisning og indstillinger

- Statusikonerne i øverste bjælke (batteristatus, tryk-til-test-status, netværksindikator osv.) vises ikke, mens et grafikprogram kører.
- Standard tegningsfarve: Sort (0,0,0)
- Standard pennefacon - normal, glat
 - Tykkelse: 1 (tynd), 2 (normal), 3 (tykkest)
 - Typografi 1 (glat), 2 (punkteret), 3 (stiplet)
- Alle tegningskommandoer bruger de nuværende indstillinger for farve og pen enten standardværdier eller dem, der blev indstillet via TI-Basic-kommandoer.
- Tekstens skrifttype er fast og kan ikke ændres.
- Enhver udgang til grafikskærmen tegnes inden for et klipvindue, som er størrelsen på grafikskærmens tegneområde. Enhver tegnet output, der strækker sig uden for dette klippede grafiske skærmområde, tegnes ikke. Der vises ingen fejlmeddelelse.
- Alle x, y-koordinater, der er angivet til tegningskommandoer, er defineret således, at 0,0 er øverst til venstre i grafikskærmens tegneområde.
 - **Undtagelser:**
 - **DrawText** bruger koordinaterne som nederste venstre hjørne af grænsekassen til teksten.
 - **SetWindow** bruger nederste venstre hjørne af skærmen
- Alle parametre for kommandoerne kan gives som udtryk, der evalueres til et tal, som derefter afrundes til nærmeste heltal.

Fejlmeddelelser på grafiskskærmen

Hvis validering mislykkes, vises en fejlmeddelelse.

Fejlmeddelelse	Beskrivelse	Vis
Fejl Syntaks	Hvis syntakskontrollen finder syntaksfejl, viser den en fejlmeddelelse, og prøver at placere markøren tæt på den første fejl, så du kan rette den.	
Fejl For få argumenter	Funktionen eller kommandoen mangler et eller flere argumenter	
Fejl For mange argumenter	Funktionen eller kommandoen indeholder for mange argumenter og kan ikke beregnes.	
Fejl Ugyldig datatype	Et argument er af forkert datatype.	

Ugyldige kommandoer i grafiktilstand

Nogle kommandoer er ikke tilladt, når programmet skifter til grafiktilstand. Hvis disse kommandoer opstår i grafiktilstand og fejl vises, og programmet afsluttes.

Ikke-tilladt kommando	Fejlmeddelelse
Anmodning	Anmodning kan ikke udføres i grafiktilstand
RequestStr	RequestStr kan ikke udføres i grafiktilstand
tekst	Tekst kan ikke udføres i grafiktilstand

Kommandoerne, der udskriver tekst til lommeregneren - **disp** and **dispAt** - vil være understøttede kommandoer i grafiktilstanden. Teksten fra disse kommandoer vil blive sendt til lommeregnerskærmen (ikke på grafik) og vil være synlig, når programmet udgår, og systemet skifter tilbage til Lommeregner-appen

Ryd (Clear)Katalog > 
CXII**Ryd** x, y , *bredde*, *højde*

Ryd

Rydder hele skærmen, hvis der ikke er angivet nogen parametre.

Rydder hele skærmen

Hvis x, y , *bredde* og *højde* er angivet, vil rektanglet der defineres af parametrene vil blive ryddet.

Ryd 10,10,100,50

Rydder et rektangelområde med øverste venstre hjørne på (10, 10) og med bredde 100, højde 50

DrawArc
 Katalog > 
CXII

DrawArc *x, y, bredde, højde, startAngle, arcAngle*

Tegner en bue i det afgrænsende rektangel med de tilvejebragte start- og buevinkler.

x, y: øverste venstre koordinat af afgrænsende rektangel

bredde, højde: dimensioner af et afgrænset rektangel

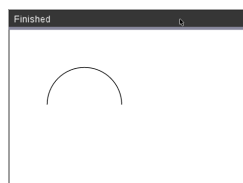
"Buevinklen" definerer buens størrelse.

Disse parametre kan gives som udtryk, der evalueres til et tal, som derefter afrundes til nærmeste heltal.

DrawArc 20,20,100,100,0,90



DrawArc 50,50,100,100,0,180



Se også: [FillArc](#)

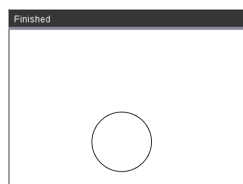
DrawCircle
 Katalog > 
CXII

DrawCircle *x, y, radius*

x, y: koordinat af centrum

radius: cirkelns radius

DrawCircle 150,150,40



Se også: [FillCircle](#)

DrawLine

Katalog > 
CXII

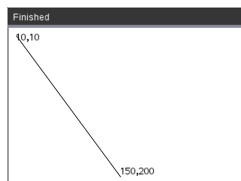
DrawLine $x1, y1, x2, y2$

Tegn en linje fra $x1, y1, x2, y2$.

Udtryk der evalueres til et tal, som derefter afrundes til nærmeste heltal.

Skærmgrænser: Hvis de angivne koordinater forårsager, at en del af linjen tegnes uden for grafiskskærmen, bliver den del af linjen klippet, og der vises ingen fejlmeddelelse.

DrawLine 10,10,150,200



DrawPoly

Katalog > 
CXII

Kommandoerne har to varianter:

DrawPoly $xlist, ylist$

or

DrawPoly $x1, y1, x2, y2, x3, y3...xn, yn$

Bemærk: DrawPoly $xlist, ylist$

Form vil forbinde $x1, y1$ til $x2, y2, x2, y2$ til $x3, y3$ og så videre.

Bemærk: DrawPoly $x1, y1, x2, y2, x3,$

$y3...xn, yn$

xn, yn vil **IKKE** blive automatisk forbundet med $x1, y1$.

Udtryk der evalueres til en liste over reelle floats

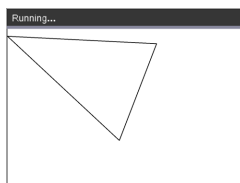
$xlist, ylist$

Udtryk der evalueres til en enkelt reel float
 $x1, y1...xn, yn$ = koordinater for polygonens hjørner

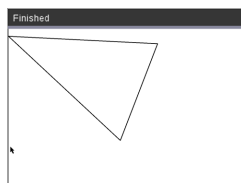
$xlist:=\{0,200,150,0\}$

$ylist:=\{10,20,150,10\}$

DrawPoly $xlist,ylist$



DrawPoly 0,10,200,20,150,150,0,10



Bemærk: DrawPoly: Indtast størrelsesdimensioner (bredde/højde) i forhold til tegnede linjer. Linjerne er tegnet i en afgrænsningskasse omkring det angivne koordinat og dimensioner således, at den faktiske størrelse af den tegnede polygon vil være større end bredden og højden.

Se også: [FillPoly](#)

DrawRect

DrawRect *x, y, bredde, højde*

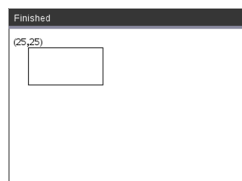
x, y: øverste venstre koordinat af rektangel

bredde, højde: bredde og højde af rektangel (rektangel trukket ned og til højre fra startkoordinat).

Bemærk: Linjerne er tegnet i en afgrænsningskasse omkring den angivne koordinat og dimensioner således, at den faktiske størrelse af det tegnede rektangel vil være større end bredden og højden der er indikeret.

Se også: [FillRect](#)

DrawRect 25,25,100,50



DrawText

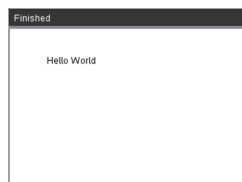
DrawText *x, y, exprOrString1*
[,exprOrString2]...

x, y: koordinat af tekstoutput

Tegner teksten i *exprOrString* ved den specificerede *x, y* koordinatplacering.

Reglerne for *exprOrString* er de samme som for **Disp** – **DrawText** kan tage flere argumenter.

DrawText 50,50,"Hello World"



FillArc

 Katalog > 
 CXII

FillArc *x, y, bredde, højde startAngle, arcAngle*

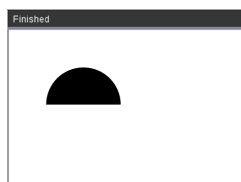
x, y: øverste venstre koordinat af afgrænsende rektangel

Tegn og fyld en bue i det afgrænsende rektangel med de tilvejebragte start- og buevinkler.

Standardfyldfarve er sort. Fyldfarven kan indstilles af [SetColor](#)-kommandoen

"Buevinklen" definerer buens svej.

FillArc 50,50,100,100,0,180



FillCircle

 Katalog > 
 CXII

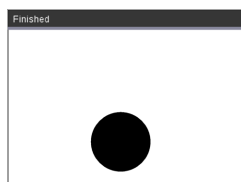
FillCircle *x, y, radius*

x, y: koordinat af centrum

Tegn og fyld en cirkel på det specificerede center med den angivne radius.

Standardfyldfarve er sort. Fyldfarven kan indstilles af [SetColor](#)-kommandoen.

FillCircle 150,150,40



Her!

FillPoly

 Katalog > 
 CXII

FillPoly *xlist, ylist*

or

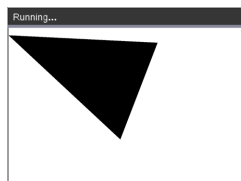
FillPoly *x1, y1, x2, y2, x3, y3...xn, yn*

Bemærk: Linjen og farven er angivet af [SetColor](#) og [SetPen](#)

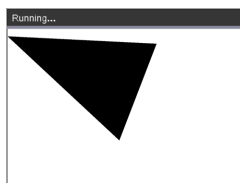
xlist:={0,200,150,0}

ylist:={10,20,150,10}

FillPoly xlist,ylist



FillPoly 0,10,200,20,150,150,0,10



FillRect

FillRect *x, y, bredde, højde*

x, y: øverste venstre koordinat af rektangel

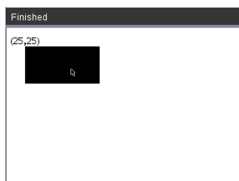
bredde, højde: bredde og højde af rektangel

Tegn og udfyld et rektangel med øverste venstre hjørne ved koordinaten angivet af (*x,y*)

Standardfyldfarve er sort. Fyldfarven kan indstilles af [SetColor](#)-kommandoen

Bemærk: Linjen og farven er angivet af [SetColor](#) og [SetPen](#)

FillRect 25,25,100,50



getPlatform()Katalog > 
CXII**getPlatform()**

getPlatform()

"dt"

Returnerer:

"dt" på desktop-softwareapplikationer

"hh" på TI-Nspire™ CX håndholdte

"ios" på TI-Nspire™ CX iPad® app

PaintBufferKatalog > 
CXII**PaintBuffer**

Grafisk buffer til skærmen

Denne kommando bruges sammen med UseBuffer for at øge visningshastighed på skærmen, når programmet genererer flere grafiske objekter.

UseBuffer

For n,1,10

x:=randInt(0,300)

y:=randInt(0,200)

radius:=randInt(10,50)

Wait 0,5

DrawCircle x,y,radius

EndFor

PaintBuffer

Dette program viser alle de 10
cirkler på én gang.

Hvis kommandoen "UseBuffer" fjernes,
vises hver cirkel, som den er
tegnet.

Se også: [UseBuffer](#)

PlotXY $x, y, shape$ x, y : koordinat til at tegne form $shape$: et tal mellem 1 og 13, der angiver formen

1 - Udfyldt cirkel

2 - Tom cirkel

3 - Udfyldt firkant

4 - Tom firkant

5 - Kryds

6 - Plus

7 - Tynd

8 - medium punkt, udfyldt

9 - medium punkt, tomt

10 - større punkt, udfyldt

11 - større punkt, tomt

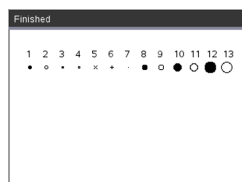
12 - største punkt, udfyldt

13 - største punkt, tomt

PlotXY 100,100,1

For $n, 1, 13$ DrawText $1+22*n, 40, n$ PlotXY $5+22*n, 50, n$

EndFor



SetColor
Katalog > 
CXII
SetColor

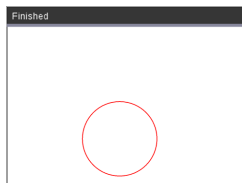
Rød-værdi, Grøn-værdi, Blå-værdi

Værdierne for Rød, Grøn, Blå skal være mellem 0 og 255

Indstiller farven til efterfølgende tegningskommandoer

SetColor 255,0,0

DrawCircle 150,150,100

**SetPen**
Katalog > 
CXII
SetPen

tykkelse, stil

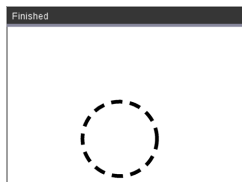
tykkelse: 1 <= tykkelse <= 3 | 1 er tyndest, 3 er tykkest

stil: 1 = Glat, 2 = Punkteret, 3 = Stiplet

Indstiller pennefaconen til efterfølgende tegningskommandoer

SetPen 3,3

DrawCircle 150,150,50

**SetWindow**
Katalog > 
CXII
SetWindow

xMin, xMax, yMin, yMax

Etablerer et logisk vindue, der er knyttet til det grafiske tegneområde. Alle parametre er påkrævede.

Hvis delen af det tegnede objekt er uden for vinduet, vil resultatet blive klippet (ikke vist), og der vises ingen fejlmeddelelse.

SetWindow 0,160,0,120

vil indstille outputvinduet til 0,0 i nederste venstre hjørne med en bredde på 160 og en højde på 120

DrawLine 0,0,100,100

SetWindow 0,160,0,120

SetPen 3,3

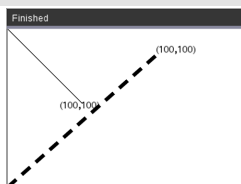
DrawLine 0,0,100,100

Hvis $x_{\min}$ er større end eller lig med $x_{\max}$ eller $y_{\min}$ er større end eller lig med $y_{\max}$, vises en fejlmeddelelse.

Eventuelle objekter tegnet før en SetWindow-kommando vil ikke blive tegnet igen i den nye konfiguration.

For at nulstille vindueparametrene til standardværdien skal du bruge:

SetWindow 0,0,0,0



UseBuffer

 Katalog > 
 CXII

UseBuffer

Tegn til grafikbuffer i stedet for skærm (for at øge ydeevnen)

Denne kommando bruges sammen med PaintBuffer for at øge visningshastighed på skærmen, når programmet genererer flere grafiske objekter.

Med UseBuffer vises al grafikken først, når den næste PaintBuffer-kommando er udført.

UseBuffer skal kun kaldes en gang i programmet, dvs. hver brug af PaintBuffer behøver ikke en tilsvarende UseBuffer

Se også: [PaintBuffer](#)

UseBuffer

```

For n,1,10
  x:=randInt(0,300)
  y:=randInt(0,200)
  radius:=randInt(10,50)
  Wait 0,5
  DrawCircle x,y,radius
EndFor
PaintBuffer

```

Dette program viser alle de 10 cirkler på én gang.

Hvis kommandoen "UseBuffer" fjernes, vises hver cirkel, som den er tegnet.

Tomme (ugyldige) elementer

Ved analyse af data fra virkeligheden har du muligvis ikke et komplet datasæt. TI-Nspire™ CAS tillader tomme eller ugyldige dataelementer, så du kan fortsætte med det næsten komplette sæt data i stedet for at skulle starte forfra eller kassere de ukomplette forekomster.

Du kan finde et eksempel på data, der medtager tomme elementer i kapitlet Lister og regneark under *“Tegning af regnearkdata”*.

Med funktionen **delVoid()** kan du fjerne tomme elementer fra en liste. Med funktionen **isVoid()** kan du teste for et tomt element. Nærmere beskrivelse findes under **delVoid()**, side 52, og **isVoid()**, side 100.

Bemærk: Du kan indtaste et tomt element manuelt i et matematisk udtryk ved at skrive “_” eller nøgleordet **void**. Nøgleordet **void** konverteres automatisk til symbolet “_”, når udtrykket evalueres. Du skriver “_” på den håndholdte ved at trykke  .

Beregninger, der omfatter ugyldige elementer

De fleste beregninger med et ugyldigt input giver et ugyldigt resultat. Se de specielle tilfælde i det følgende.

	—
$\gcd(100, _)$	—
$3 + _$	—
$\{5, _, 10\} - \{3, 6, 9\}$	$\{2, _, 1\}$

Listeargumenter med ugyldige elementer

Følgende funktioner og kommandoer ignorerer (springer over) ugyldige elementer i listeargumenter.

count, **countIf**, **cumulativeSum**, **freqTable**  **list**, **frequency**, **max**, **mean**, **median**, **product**, **stDevPop**, **stDevSamp**, **sum**, **sumIf**, **varPop** og **varSamp** samt regressionsberegninger, **OneVar**, **TwoVar**, og **FiveNumSummary** statistik, konfidensintervaller, og statistiske tests

$\text{sum}(\{2, _, 3, 5, 6, 6\})$	16.6
$\text{median}(\{1, 2, _, _, 3\})$	2
$\text{cumulativeSum}(\{1, 2, _, 4, 5\})$	$\{1, 3, _, 7, 12\}$
$\text{cumulativeSum}\left(\begin{bmatrix} 1 & 2 \\ 3 & - \\ 5 & 6 \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 2 \\ 4 & - \\ 9 & 8 \end{bmatrix}$

Listeargumenter med ugyldige elementer

SortA og **SortD** flytter alle ugyldige elementer i det første argument til bunden.

$\{5,4,3,_,1\} \rightarrow list1$	$\{5,4,3,_,1\}$
$\{5,4,3,2,1\} \rightarrow list2$	$\{5,4,3,2,1\}$
SortA list1,list2	Done
list1	$\{1,3,4,5,_{-}\}$
list2	$\{1,3,4,5,2\}$

$\{1,2,3,_,5\} \rightarrow list1$	$\{1,2,3,_,5\}$
$\{1,2,3,4,5\} \rightarrow list2$	$\{1,2,3,4,5\}$
SortD list1,list2	Done
list1	$\{5,3,2,1,_{-}\}$
list2	$\{5,3,2,1,4\}$

I regressioner indfører en ugyldig værdi i en X- eller Y-liste en ugyldig værdi for det tilsvarende element i residualerne.

$l1:=\{1,2,3,4,5\}; l2:=\{2,_,3,5,6,6\}$	$\{2,_,3,5,6,6\}$
LinRegMx l1,l2	Done
stat.Resid	$\{0.434286,_, -0.862857, -0.011429, 0.44\}$
stat.XReg	$\{1,_,3,4,5\}$
stat.YReg	$\{2,_,3,5,6,6\}$
stat.FreqReg	$\{1,_,1,1,1,1\}$

En udeladt kategori i regressioner indfører en ugyldig værdi for det tilsvarende element i residualerne.

$l1:=\{1,3,4,5\}; l2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
cat:={"M", "M", "F", "F"}: incl:={"F"}	$\{ "F" \}$
LinRegMx l1,l2,1,cat,incl	Done
stat.Resid	$\{_,_,0,0,0\}$
stat.XReg	$\{_,_,4,5\}$
stat.YReg	$\{_,_,5,6,6\}$
stat.FreqReg	$\{_,_,1,1,1\}$

En hyppighed på 0 i regressioner indfører en ugyldig værdi i det tilsvarende element i residualerne.

$l1:=\{1,3,4,5\}; l2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
LinRegMx l1,l2,{1,0,1,1}	Done
stat.Resid	$\{0.069231,_, -0.276923, 0.207692\}$
stat.XReg	$\{1,_,4,5\}$
stat.YReg	$\{2,_,5,6,6\}$
stat.FreqReg	$\{1,_,1,1,1\}$

Genveje til indtastning af matematiske udtryk

Med genveje kan du indtaste elementer i matematiskeudtryk ved at skrive i stedet for at anvende Katalog eller Symbolpaletten. Du kan for eksempel indtaste udtrykket $\sqrt{6}$, ved at skrive `sqrt(6)` i indtastningslinjen. Når du trykker på `enter`, ændres udtrykket `sqrt(6)` til $\sqrt{6}$. visse genveje er nyttige fra både den håndholdtes og computerens tastatur. Andre er hovedsageligt nyttige på computerens tastatur.

Fra den håndholdtes eller computerens tastatur

Indsæt dette:	- ved at skrive denne genvej:
π	<code>pi</code>
θ	<code>theta</code>
∞	<code>uendeligt</code>
$\leq$	<code><=</code>
$\geq$	<code>>=</code>
$\neq$	<code>/=</code>
$\Rightarrow$ (medfører)	<code>=></code>
$\Leftrightarrow$ (ensbetydende med)	<code><=></code>
$\rightarrow$ (gem operator)	<code>=:</code>
$ $ (absolut værdi)	<code>abs (...)</code>
$\sqrt{()}$	<code>sqrt (...)</code>
$d()$	<code>derivative (...)</code>
$\int()$	<code>integral (...)</code>
$\Sigma()$ (Sumskabelon)	<code>sumSeq (...)</code>
$\Pi()$ (Produktskabelon)	<code>prodSeq (...)</code>
$\sin^{-1}()$, $\cos^{-1}()$, ...	<code>arcsin (...)</code> , <code>arccos (...)</code> , ...
$\Delta\text{List}()$	<code>deltaList (...)</code>
ΔtmpCnv	<code>deltaTmpCnv (...)</code>

På computerens tastatur

Indsæt dette:	- ved at skrive denne genvej:
$c1, c2, \dots$ (konstanter)	<code>@c1, @c2, ...</code>
$n1, n2, \dots$ (heltalskonstanter)	<code>@n1, @n2, ...</code>
i (imaginær konstant)	<code>@i</code>

Indsæt dette:	- ved at skrive denne genvej:
e (naturlig logbase e)	@ e
E (videnskabelig notation)	@ E
T (transponere)	@ t
r (radianer)	@ r
° (grader)	@ d
g (nygrader)	@ g
∠ (vinkel)	@ <
► (konvertering)	@ >
► Decimal , ► approxFraction() , osv.	@ >Decimal , @ >approxFraction() , osv.

Hierarkiet i EOS™ ligningsoperativsystemet (Equation Operating System)

I dette afsnit beskrives EOS™ (Equation Operating System) der anvendes af TI-Nspire™ CAS læringsteknologien til matematik og naturfag. Tal, variable og funktioner indtastes i en nem og enkel rækkefølge. EOS™ softwaren beregner udtryk og ligninger ved hjælp af parenteser og efter de prioriteringer, der er beskrevet nedenfor.

Operatorhierarkiet

Niveau	Operator
1	Parenteser (), kantparenteser [], krølparenteser { }
2	Henviſning (nr.)
3	Funktionskald
4	Postoperatorer: grader-minutter-sekunder ([°] , ', "), fakultet (!), procentsats (%), radian (^r), lavtstillet ([]), skifte side for lighedstegnet (^T)
5	Eksponentiering, potensoperator ([^])
6	Negation (-)
7	Strengsammenkædning &
8	Multiplikation (•), division (/)
9	Addition (+), subtraktion (-)
10	Lighedsrelationer: lig med (=), forskellig fra (≠ eller ≠), mindre end (<), mindre end eller lig med (≤ eller ≤), større end (>), større end eller lig med (≥ eller ≥)
11	Logisk not
12	logisk and
13	Logisk or
14	xor, nor, nand
15	medfører, ⇒
16	ensbetydende med (⇔)
17	Betingelses-operator (" ")
18	Gem (→)

Parenteser, kantparenteser og krølparenteser

Alle beregninger inde i parenteser, kantede parenteser eller krøllede parenteser foretages først. F.eks. beregner EOS™ softwaren i udtrykket 4(1+2) først den del af udtrykket, der er i parentes, 1+2, og multiplicerer derefter resultatet, 3, med 4.

Antallet af start- og slutparenteser, -kantparenteser og -krølparenteser skal være det samme i et udtryk eller en ligning. Hvis ikke, vises en fejlmeddelelse, der viser det manglende element. For eksempel vil $(1+2)/(3+4)$ udløse fejlmeddelelsen "Mangler)."

Bemærk: Da du i TI-Nspire™ CAS softwaren kan definere dine egne funktioner, tolkes et variabelnavn fulgt af et udtryk i parentes som et funktionskald i stedet for en underforstået multiplikation. For eksempel er $a(b+c)$ funktionen a beregnet af $b+c$. For at gange udtrykket $b+c$ med variabelen a anvendes explicit multiplikation: $a*(b+c)$.

henvisning

Henvisningsoperatoren (#) konverterer en streng til en variabel eller et funktionsnavn. For eksempel opretter #("x"&"y"&"z") variabelnavnet xyz. Henvisning muliggør også oprettelse og ændring af variable inde i et program. Hvis for eksempel $10 \rightarrow r$ og $r \rightarrow s1$, så $\#s1=10$.

Post operatorer

Postoperatorer er operatorer, der følger direkte efter et argument som $5!$, 25% , eller $60^\circ 15' 45''$. Argumenter efterfulgt af en postoperator, beregnes på fjerde prioritetsniveau. I udtrykket $4^3!$ beregnes $3!$ først. Resultatet, 6, bliver derefter eksponenten af 4, så det bliver 4096.

Eksponentiering

Eksponentiering (^) og elementvis eksponentiering (.^) beregnes fra højre mod venstre. Udtrykket 2^3^2 beregnes for eksempel på samme måde som $2^{(3^2)}$ og giver 512. Dette er forskelligt fra $(2^3)^2$, der giver 64.

Negation

Negative tal indtastes ved at trykke på $\boxed{-}$ efterfulgt af tallet. Postoperationer og eksponentiering udføres før negation. For eksempel er resultatet af $-x^2$ et negativt tal, og $-9^2 = -81$. Med parenteser kvadreres et negativt tal som $(-9)^2$, så det giver 81.

Betingelse ("|")

Det argument, der følger efter ("|")-betingelses-operatoren, angiver et sæt betingelser, der skal tages hensyn til under beregningen af det argument, der kommer før operatoren.

TI-Nspire CX II - TI-Grundlæggende programmeringsfunktioner

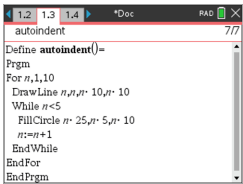
Automatisk indrykning i programmeringseditoren

TI-Nspire™-programeditor indrykker nu automatisk sætninger inde i en blokkommando.

Blokkommandoer er If/EndIf, For/EndFor, While/EndWhile, Loop/EndLoop, Try/EndTry

Editoren forbereder automatisk mellemrum for programkommandoer inde i en blokkommando. Lukkekommandoen for blokken vil blive justeret med åbningskommandoen.

Eksemplet herunder viser automatisk indrykning i indlejrede blokkommandoer.



Kodefragmenter, der kopieres og indsættes, beholder den originale indrykning.

Åbning af et program, der er oprettet i en tidligere version af softwaren, bevarer den oprindelige indrykning.

Forbedrede fejlmeddelelser til TI-Basic

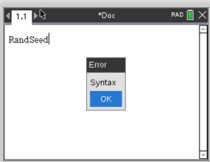
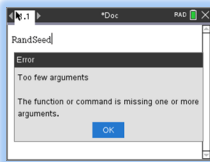
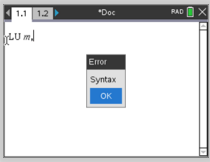
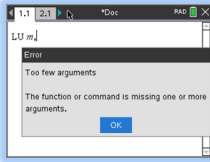
Fejlmeddelelser

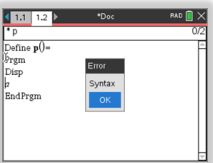
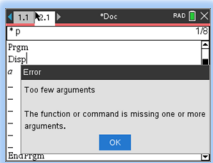
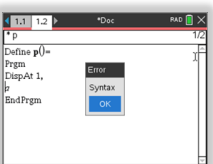
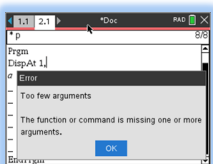
Fejltilstand	Ny meddelelse
Fejl i betingelseserklæring (If/While)	En betinget erklæring løstes ikke til TRUE eller FALSE BEMÆRK: Med ændringen for at placere markøren på linjen med fejlen, behøver vi ikke længere at angive, om fejlen er i en "If"-erklæring eller en "While"-erklæring.
Manglende EndIf	Forventede EndIf , men fandt en anden sluterklæring
Manglende EndFor	Forventede EndFor , men fandt en anden sluterklæring
Manglende EndWhile	Forventede EndWhile , men fandt en anden sluterklæring
Manglende EndLoop	Forventede EndLoop , men fandt en anden

Fejltilstand	Ny meddelelse
	sluterklæring
Manglende EndTry	Forventede EndTry , men fandt en anden sluterklæring
" Then " udeladt efter If <condition>	Manglende If..Then
" Then " udeladt efter Elseif <condition>	Then mangler i blok: Elseif .
Når " Then ", " Else " og " Elseif " opstod uden for kontrolblokke	Else ugyldig uden for blokke: If..Then..EndIf eller Try..EndTry
" Elseif " vises uden for " If..Then..EndIf "-blokke	Elseif ugyldig uden for blok: If..Then..EndIf
" Then " vises uden for " If....EndIf "-blokke	Then ugyldig uden for blok: If..EndIf

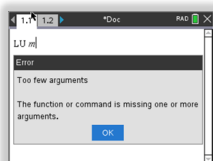
Syntaksfejl

Hvis kommandoer, der forventer en eller flere argumenter, kaldes med en ufuldstændig liste over argumenter, vil en "**Too few argument-fejl**" blive udstedt i stedet for en "**syntax**"-fejl

Aktuel adfærd	Ny CX II-adfærd
 The screenshot shows the TI-84 Plus CE calculator interface. The cursor is on the line 'RandSeed'. An error dialog box is displayed with the text 'Error Syntax' and an 'OK' button.	 The screenshot shows the TI-84 Plus CE calculator interface. The cursor is on the line 'RandSeed'. An error dialog box is displayed with the text 'Error Too few arguments' and 'The function or command is missing one or more arguments.' and an 'OK' button.
 The screenshot shows the TI-84 Plus CE calculator interface. The cursor is on the line 'L1 U m'. An error dialog box is displayed with the text 'Error Syntax' and an 'OK' button.	 The screenshot shows the TI-84 Plus CE calculator interface. The cursor is on the line 'L1 U m'. An error dialog box is displayed with the text 'Error Too few arguments' and 'The function or command is missing one or more arguments.' and an 'OK' button.

Aktuel adfærd	Ny CX II-adfærd
	
	

Bemærk: Når en ufuldstændig liste over argumenter ikke følges af et komma, er fejlmeldelsen: “too few arguments”. Dette er det samme som tidligere udgivelser.



Konstanter og værdier

Det følgende skema oplister konstanter og deres værdier, som er tilgængelige, når der udføres enhedskonvertering (NIST 2018). De kan indtastes manuelt eller vælges via

Konstant listen i **Værktøjer > Enhedskonverteringer** (Håndholdt: Tryk  **3**).

Konstant	Navn	Værdi
_c	Lysets hastighed	299792458 _m/_s
_Cc	Coulombs konstant	8987551792.261 _m/_F
_Fc	Faradays konstant	96485.33212 _coul/_mol
_g	Tyngdeaccelerationen	9.80665 _m/_s ²
_Gc	Tyngdekonstant	6.6743E-11 _m ³ /_kg/_s ²
_h	Plancks konstant	6.62607015E-34 _J _s
_k	Boltzmanns konstant	1.380649E-23 _J/_°K
_μ0	Vakuumpermeabiliteten	1.25663706127E-6 _N/_A ²
_μb	Bohr-magneton	9.274009994E-24 _J _m ² /_Wb
_Me	Elektronens hvilemasse	9.10938371390E-31 _kg
_Mμ	Myonens masse	1.883531627E-28 _kg
_Mn	Neutronens hvilemasse	1.67492750058E-27 _kg
_Mp	Protonens hvilemasse	1.67262192595E-27 _kg
_Na	Avogadros konstant	6.02214076E23 /_mol
_q	Elementarladningen	1.602176634E-19 _coul
_R	Trykkonstant for vandpotentiale	0.0831 L bars/mole K
_Rb	Bohr-radius	5.29177210903E-11 _m
_Rc	Gaskonstanten	8.314462618 _J/_mol/_°K
_Rdb	Rydbergs konstant	10973731.568160/_m
_Re	Elektronradius	2.8179403262E-15 _m
_u	Atommasse	1.66053906893E-27 _kg
_Vm	Det molare volumen	2.241396954E-2 _m ³ /_mol
_ε0	Permittivitet for et vakuum (konstant)	8.8541878188E-12 _F/_m
_σ	Stefan-Boltzmanns konstant	5.670367E-8 _W/_m ² /_°K ⁴
_φ0	Magnetisk fluxkvantum	2.067833831E-15 _Wb

Fejlkode og fejlmeddelelser

Hvis der opstår en fejl, tildeles dens kode variablen *errCode*. Brugerdefinerede programmer og funktioner kan undersøge *errCode* og bestemme årsagen til en fejl. Du finder et eksempel på anvendelse af *errCode* i Eksempel 2 under kommandoen **Try**, her.

Bemærk: Visse fejltilstande gælder kun for *TI-Nspire™ CAS-produkter*, og andre gælder kun for *TI-Nspire™-produkter*.

Fejlkode	Beskrivelse
10	En funktion returnerede ikke en værdi
20	En test gav ikke resultatet SAND eller FALSK. Generelt kan udefinerede variable ikke sammenlignes. For eksempel forårsager testen <code>If a<b</code> denne fejl, enten fordi <code>a</code> eller <code>b</code> er udefineret, når sætningen eksekveres.
30	Argumentet kan ikke være et navn på en mappe.
40	Argumentfejl
50	Argumentuoverensstemmelse To eller flere argumenter skal være af samme type.
60	Argumentet skal være et boolsk udtryk eller et heltal
70	Argumentet skal være et decimaltal
90	Argumentet skal være en liste
100	Argumentet skal være en matrix
130	Argumentet skal være en streng
140	Argumentet skal være et variabelnavn. Kontroller, at navnet: • ikke begynder med et tal • ikke indeholder mellemrum eller specialtegn • ikke anvender understregning eller punktum forkert • ikke overskrider begrænsningerne for længden I afsnittet Regner i dokumentationen er der flere oplysninger.
160	Argumentet skal være et udtryk
165	Batterierne er for flade til at kunne sende/modtage Isæt nye batterier, før du sender eller modtager
170	Grænse Nedre grænse skal være mindre end øvre grænse for at definere søgeintervallet.

Fejlkode	Beskrivelse
180	Afbrudt Der er blevet trykket på  eller  -tasten under en længere beregning eller under en programeksekvierung.
190	Cirkulær definition Denne meddelelse vises for at undgå at løbe tør for hukommelse under uendelig erstatning af variabelværdier under reduktion. For eksempel vil $a+1 \rightarrow a$, hvor a er en udefineret variabel give denne fejl.
200	Ugyldigt begrænsningsudtryk For eksempel vil $\text{solve}(3x^2-4=0,x) \mid x<0 \text{ or } x>5$ give denne fejlmeddelelse, fordi begrænsningen er adskilt med "or" i stedet for "and."
210	Ugyldig datatype Et argument er af forkert datatype.
220	Afhængig grænse
230	Dimension Et liste- eller et matrixindex er ikke gyldigt. Hvis listen {1,2,3,4} for eksempel gemmes i L1, vil L1[5] være en dimensionsfejl, fordi L1 kun indeholder fire elementer.
235	Dimensionsfejl. Der er ikke nok elementer i listerne.
240	Dimensionsuoverensstemmelse To eller flere argumenter skal være af samme dimension. For eksempel er $[1,2]+[1,2,3]$ en dimensionsuoverensstemmelse, fordi matricerne indeholder et andet antal elementer.
250	Division med 0
260	Domænefejl Et argument skal være i et angivet domæne. For eksempel er $\text{rand}(0)$ ikke gyldig.
270	Variabelnavn findes allerede
280	Else og Elseif er ikke gyldige uden for blokken If..EndIf
290	EndTry mangler den tilsvarende Else-erklæring
295	For stor iteration
300	Forventede en liste med 2 eller 3 elementer eller en matrix
310	Første argument af nSolve skal være en ligning i en enkelt variabel. Det kan ikke indeholde en variabel uden en værdi bortset fra den variabel, der skal undersøges.
320	Første argument for solve eller cSolve skal være en ligning eller en ulighed

Fejlkode	Beskrivelse
	For eksempel er $\text{solve}(3x^4-4,x)$ ugyldig, fordi det første argument ikke er en ligning.
345	Inkonsistente enheder
350	Indeks uden for området
360	Henvisningsstrengen er ikke et gyldigt variabelnavn
380	Udefineret Ans Enten gav den foregående beregning ikke en Ans, eller der blev ikke indtastet en tidligere beregning.
390	Ugyldig tildeling
400	Ugyldig tildelingsværdi
410	Ugyldig kommando
430	Ugyldigt for de aktuelle indstillinger
435	Ugyldigt gæt
440	Ugyldig implicit multiplikation for eksempel er $x(x+1)$ ugyldig, mens $x*(x+1)$ er den korrekte syntaks. Dette er for at undgå forveksling mellem implicit multiplikation og funktionskald.
450	Ugyldig i en funktion eller i det aktuelle udtryk Kun visse kommandoer er gyldige i en brugerdefineret funktion.
490	Ugyldig i Try..EndTryin Try..EndTry-blok
510	Ugyldig liste eller matrix
550	Ugyldig uden for en funktion eller et program Et antal kommandoer er ugyldige uden for en funktion eller et program. For eksempel kan Local ikke bruges, medmindre det er i en funktion eller program.
560	Ugyldig uden for blokkene Loop..EndLoop, For..EndFor eller While..EndWhile For eksempel er Exit-kommandoen kun gyldig inde i disse to løkkeblokke.
565	Ugyldig uden for et program
570	Ugyldigt stinavn For eksempel er \var ugyldigt.
575	Ugyldigt pol-kompleks
580	Ugyldig programreference

Fejlkode	Beskrivelse
	Programmer kan ikke kaldes i funktioner eller udtryk som $1+p(x)$, hvor p er et program.
600	Ugyldig tabel
605	Ugyldig brug af enheder
610	Ugyldigt variabelnavn i en Local-sætning
620	Ugyldigt variabel- eller funktionsnavn
630	Ugyldig variabelhenvisning
640	Ugyldig vektorsyntaks
650	Link-overførsel En overførsel mellem to enheder blev ikke gennemført. Kontroller, at forbindelseskablet er godt tilsluttet i begge ender.
665	Matricen kan ikke diagonaliseres
670	Lav hukommelse 1. Slet nogle data i dette dokument 2. Gem og luk dette dokument Hvis 1 og 2 ikke lykkes, skal batterierne tages ud og sættes på plads igen
672	Resourcer opbrugt
673	Resourcer opbrugt
680	Mangler (
690	Mangler)
700	Mangler “
710	Mangler]
720	Mangler }
730	Mangler starten eller slutningen af bloksyntaksen
740	Mangler Then i blokken If..Endif
750	Navnet er ikke en funktion eller et program
765	Ingen funktion er valgt
780	Ingen løsning blev fundet
800	Ikke-reelt resultat Hvis softwaren for eksempel er i indstillingen Reel, er $\sqrt{-1}$ ugyldig.

Fejlkode	Beskrivelse
	For at kunne tillade komplekse resultater skal du ændre tilstanden Reel eller Kompleks til REKTANGULÆR eller POLÆR
830	Overløb
850	Programmet blev ikke fundet Et programkald inde i et andet program blev ikke fundet i den angivne sti under eksekvering.
855	Rand funktioner er ikke tilladt graftegning
860	For mange rekursioner
870	Reserveret navn eller systemvariabel
900	Argumentfejl Median-median-modellen kunne ikke anvendes på datasættet.
910	Syntaksfejl
920	Teksten blev ikke fundet
930	For få argumenter Funktionen eller kommandoen mangler et eller flere argumenter.
940	For mange argumenter Udtrykket eller ligningen indeholder for mange argumenter og kan ikke beregnes.
950	For mange lavtstillede tegn
955	For mange udefinerede variable
960	Variablen er ikke defineret Der er ikke tildelt en værdi til variabelen. Anvend en af følgende kommandoer: • sto → • := • Define for at tildele værdier til variable.
965	OS uden licens
970	Variabel i brug, så referencer eller ændringer er ikke tilladt
980	Variablen er beskyttet
990	Ugyldigt variabelnavn Kontroller, at navnet ikke overskrider begrænsningerne for længden

Fejlkode	Beskrivelse
1000	Talområde for Vindue variabel
1010	Zoom
1020	Intern fejl
1030	Overskridelse af beskyttet hukommelse
1040	Funktionen understøttes ikke. Denne funktion kræver Computer Algebra System. Prøv TI-Nspire™ CAS.
1045	Operatoren understøttes ikke. Denne operator kræver Computer Algebra System. Prøv TI-Nspire™ CAS.
1050	Funktionen understøttes ikke. Denne operator kræver Computer Algebra System. Prøv TI-Nspire™ CAS.
1060	Input-argumentet skal være et tal. Der er kun tilladt input med numeriske værdier.
1070	Trig-funktionsargumentet er for stort til en nøjagtig reduktion
1080	Ikke-understøttet anvendelse af Ans. Denne applikation understøtter ikke Ans.
1090	Funktionen er ikke defineret. Anvend en af følgende kommandoer: • Define • := • sto → til at definere en funktion.
1100	Ikke-reel beregning Hvis softwaren for eksempel er i indstillingen Reel, er $\sqrt{-1}$ ugyldig. For at kunne tillade komplekse resultater skal du ændre tilstanden Reel eller Komplex til REKTANGULÆR eller POLÆR
1110	Ugyldige grænser
1120	Intet fortegnsskift
1130	Argument kan ikke være en liste eller matrix
1140	Argumentfejl Det første argument skal være et polynomielt udtryk i det andet argument. Hvis det andet argument udelades, vil softwaren forsøge at vælge en standardværdi.
1150	Argumentfejl De første to argumenter skal være polynomielle udtryk i det tredje argument. Hvis det tredje argument udelades, vil softwaren forsøge at vælge en standardværdi.
1160	Ugyldigt biblioteksstnavn

Fejlkode	Beskrivelse
	Et stinavn skal være på formen <code>xxx\yyy</code>, hvor: <code>xxx</code>-delen kan have 1 til 16 tegn. <code>yyy</code>-delen kan have 1 til 15 tegn. Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1170	Ugyldig brug af biblioteksstinavn - En værdi kan ikke tildeles et stinavn med Define, <code>:=</code> eller <code>sto</code> →. - Et stinavn kan ikke erklæres som en lokal variabel eller anvendes som en parameter i en funktion eller en programdefinition.
1180	Ugyldigt biblioteksvariabelnavn Kontroller, at navnet: - Ikke indeholder et punktum - Ikke begynder med en understregning - Ikke er længere end 15 tegn Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1190	Biblioteksdokumentet blev ikke fundet. - Kontroller, at biblioteket er i MyLib folder. - Opdater biblioteker. Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1200	Biblioteksvariablen blev ikke fundet. - Kontroller, at biblioteksvariablen findes i den første opgave i biblioteket. - Kontroller, at biblioteksvariablen er defineret som LibPub eller LibPriv. - Opdater biblioteker. Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1210	Ugyldigt navn på biblioteksgenvej Kontroller, at navnet: - Ikke indeholder et punktum - Ikke begynder med en understregning - Ikke er længere end 16 tegn - Ikke er et reserveret navn Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1220	Domænefejl: Funktioner <code>tangentLine</code> og <code>normalLine</code> understøtter kun reelle funktioner.
1230	Domænefejl.

Fejlkode	Beskrivelse
	Trigonometriske konverteringsoperatorer understøttes ikke i vinkeltilstandene Grader eller Nygrader.
1250	Argumentfejl Brug et system af lineære ligninger . Eksempel på et system med to lineære ligninger med variablerne x og y: $3x+7y=5$ $2y-5x=-1$
1260	Argumentfejl: Det første argument til nfMin eller nfMax skal være et udtryk i en enkelt variabel. Det kan ikke indeholde en variabel uden en værdi bortset fra den variabel, der skal undersøges.
1270	Argumentfejl Differentialkvotientens orden skal være 1 eller 2.
1280	Argumentfejl Brug et polynomium i udvidet form i en variabel.
1290	Argumentfejl Brug et polynomium i en variabel.
1300	Argumentfejl Koefficienter i polynomiet skal beregnes til talværdier.
1310	Argumentfejl: En funktion kunne ikke beregnes i et eller flere af den argumenter.
1380	Domænefejl Indlejrede funktionskald er ikke tilladt i domain()-funktionen.

Fejlkode- og meddelelser

Du kan bruge funktionen **warnCodes()** til at gemme fejlkode genereret ved beregning af et udtryk. Denne tabel viser de numeriske fejlkode og de tilhørende meddelelser. Se **warnCodes()** for at få et eksempel på lagring af fejlkode, her.

Fejlkode	Meddelelse
10000	Operationen kan måske indføre falske løsninger. Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10001	Differentiation af en ligning kan give en falsk ligning.
10002	Tvivlsom løsning Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10003	Tvivlsom præcision Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10004	Operationen kan miste løsninger. Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10005	cSolve kan give flere nulpunkter.
10006	Solve kan give flere nulpunkter. Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10007	Der kan være flere løsninger. Forsøg at specificere relevante nedre og øvre grænser og/eller et gæt. Eksempler med brug af solve(): • <code>solve(Ligning, Var=Gæt) nedreGrænse<Var<øvreGrænse</code> • <code>solve(Ligning, Var) nedreGrænse<Var<øvreGrænse</code> • <code>solve(Ligning,Var=Gæt)</code> Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10008	Domænet af resultatet kan være mindre end domænet af inputtet.
10009	Domænet af resultatet kan være større end domænet af inputtet.
10012	Ikke-reel beregning
10013	∞^0 eller undef^0 erstattet med 1
10014	undef^0 erstattes med 1
10015	1^∞ eller 1^undef erstattet med 1
10016	1^undef erstattes med 1

Fejlkode	Meddelelse
10017	Overløb erstattes med ∞ eller $-\infty$
10018	Operationen kræver og returnerer en 64 bit-værdi.
10019	Resursen er udtømt, reduktionen er måske ufuldstændig.
10020	Trig-funktionsargumentet er for stort til en nøjagtig reduktion.
10021	Inputtet indeholder en udefineret parameter. Resultatet er muligvis ikke gyldigt for alle mulige parameterværdier.
10022	Angivelse af korrekte nedre og øvre grænser kan give en løsning.
10023	Skalar er blevet multipliceret med identitetsmatrixen.
10024	Resultatet er opnået ved hjælp af approksimativ aritmetik.
10025	Ækvivalensen kan ikke verificeres i tilstanden EXACT.
10026	Begrænsning ignoreres muligvis. Specificer begrænsning på formen "\" Variable MathTestSymbol Constant" eller en konjunktion af disse formler eksempelvis "x<3 and x>-12"

Generelle oplysninger

Online hjælp

education.ti.com/eguide

Vælg dit land for at få flere produktoplysninger.

Kontakt TI-Support

education.ti.com/ti-cares

Vælg dit land for at finde ressourcer for teknisk support og andre supportressourcer.

Service og garanti

education.ti.com/warranty

Vælg dit land for at få oplysninger om varigheden og betingelserne for garantien, eller om produktservice.

Begrænset reklamationsret. Denne garanti påvirker ikke dine lovbestemte rettigheder.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243

Indeks

!		—	
!, fakultet	234	—, enhedsbenævnelse	244
"		 	
", sekundnotation	242	, betingelses-operator	246
#		,	
#, henvisning	240	', mærke	244
#, henvisningsoperator	271	', minutnotation	242
%		+	
%, procent	229	+, adder	223
&		=	
&, tilføj	234	≠, forskellig fra	230
*		=, lig med	229
*, multiplicere	224	>	
-		>, større end	232
-, subtrahere	223	Π	
.		Π, produkt	237
., punktum multiplikation	228	Σ	
.-, punktum subtraktion	227	Σ(), sum	238
./, punkt division	228	ΣInt()	238
.^, punktum potens	228	ΣPrn()	239
., punktum addition	227	√	
/		√, kvadratrodd	237
/, dividere	225	∫	
:		∫, integrer	235
:=, tildel	249	≤	
^		≤, mindre end eller lig med	231
^, potens	226	≥	
^-1, reciprok	246	≥, større end eller lig med	232

►		#	
►, konvertere enheder	245	0h, hexadecimal indikator	249
►approxFraction()	14	#	
►Base2, vise som binær	18	10^(), tiende potens	246
►Base10, vise som decimalt heltal ..	19	A	
►Base16, vise som hexadecimalt	20	abs(), absolut værdi	8
►cos, vises i termer af cosinus	31	Absolut værdi	
►Cylind, vise som cylindrisk vektor ..	45	skabelon til	3-4
►DD, vise som decimal vinkel	48	adder, +	223
►Decimal, vise resultat som decimal	48		
►DMS, vise som		Æ	
grader/minutter/sekunder	58	ægte brøk, propFrac	147
►exp, viser i termer af e	68	A	
►Grad, konvertere til vinkelmål i		afrund, round()	164
nygrader	92	A	
►Polar, vise som polær vektor	140	afslut løkke, EndLoop	116
►Rad, konverter til vinkelmål i		A	
radianer	151	afslut, Exit	67
►Rect, vis som rektangulær vektor ..	154	A	
►sin, vis udtryk i termer af sinus	176	afslutte funktion, EndFunc	81
►Sphere, vise som en kuglevektor ..	185	A	
►tmpCnv()	201	amortiseringstabel, amortTbl()	8, 17
→		A	
→, lagring	248	amortTbl(), amortiseringstabel	8, 17
⇒		A	
⇒, medfører[*]	233, 268	and, Boolsk operator	9
↔		A	
↔, ensbetydende med[*]	233	anden afledede	
©		skabelon til	6
©, kommentar	249		
°			
°, grader/minutter/sekunder	242		
°, gradnotation	242		
#			
Ob, binær indikator	249		

A		A	
andengradsregression, QuadReg ...	148	arccsch()	15
A		A	
angle(), vinkel	10	arcLen(), buemål	15
A		A	
Anmodning	157	arcsec()	15
A		A	
ANOVA, envejs variansanalyse	11	arcsech()	16
A		A	
ANOVA2way, to-vejs variansanalyse	11	arcsin()	16
A		A	
ans, sidste resultat	13	arcsinh()	16
A		A	
approx(), tilnærmet	14	arctan()	16
A		A	
approxRational()	14	artanh()	16
A		A	
arccos()	15	arcuscosinus, $\cos^{-1}()$	33
A		A	
arccosh()	15	arcussinus, $\sin^{-1}()$	178
A		A	
arccot()	15	arcustangens, $\tan^{-1}()$	194
A		A	
arccoth()	15	argumenter i TVM-funktioner	206
A		A	
arccsc()	15	augment(), udvid/sammenkæd	16

A		
avgRC(), gennemsnitlig		
ændringshastighed	17	
B		
beregne polynomium, polyEval() ...	142	
beregning, rækkefølge af	270	
Bestemt integral		
skabelon til	6	
betingelses-operator " "	246	
betingelses-operator, rækkefølge for		
en beregning	270	
Bibliotek (Library)		
danner genveje til objekter	102	
binær		
indikator, Ob	249	
vise, ►Base2	18	
binomCdf()	21, 98	
binomPdf()	21	
Boolske operatoren		
⇒	233, 268	
⇐	233	
and	9	
nand	126	
nor	130	
not	132	
or	137	
xor	213	
brøker		
ægteBrøk	147	
skabelon til	1	
brugerdefinerede funktioner	49	
brugerdefinerede funktioner og		
programmer	50	
buemål, arcLen()	15	
C		
Cdf()	74	
ceiling(), rund op	21	
centralDiff()	22	
cFactor(), kompleks faktor	23	
char(), tegnstreng	24	
charPoly()	24	
ClearAZ	27	
ClrErr, slet fejl	27	
colAugment	27	
colDim(), matrix kolonnedimension	28	
colNorm(), matrix kolonnenorm ...	28	
comDenom(), fællesnævner	28	
completeSquare(), complete square	29	
conj(), komplekst konjugat	30	
constructMat(), konstruer matrix ..	30	
corrMat(), korrelationsmatrix	31	
cos(), cosinus	32	
cos ⁻¹ , arcuscosinus	33	
cosh(), hyperbolsk cosinus	34	
cosh ⁻¹ (), hyperbolsk arcuscosinus ..	35	
cosinus		
viser udtryk ud fra	31	
cosinus, cos()	32	
cot(), cotangens	35	
cot ⁻¹ (), arcuscotangens	36	
cotangens, cot()	35	
coth(), hyperbolsk cotangens	36	
coth ⁻¹ (), hyperbolsk arcuscotangens	37	
count(), tælle elementer i en liste ..	37	
countif(), tælle elementer i en liste		
betinget	37	
cPolyRoots()	38	
crossP(), vektorprodukt	39	
csc(), cosecans	39	
csc ⁻¹ (), invers cosecans	39	
csch(), hyperbolsk cosecans	40	
csch ⁻¹ (), invers hyperbolsk cosecans	40	
cSolve(), kompleks løse	40	
CubicReg, tredjegradsregression	43	
cumulative sum, cumulativeSum() ..	44	
cumulativeSum(), cumulative sum ..	44	
Cycle, cyklus	45	
cyklus, Cycle	45	
cylindrisk vektorvisning, ►Cylind	45	
cZeros(), komplekse nulpunkter	45	
D		
d(), første differentialkvotient	234	
dage mellem datoer, dbd()	47	
dbd(), dage mellem datoer	47	
decimal		
heltal vise, ►Base10	19	
vinkelvisning, ►DD	48	
Define	49	
Define LibPriv	50	
Define LibPub	50	
Define, definere	49	

definere		dot	
offentlig funktion eller program	50	produkt, dotP()	61
privat funktion eller program	50	dotP(), prikprodukt	61
definere, Define	49		
definitions­mængde-kommando,		E	
domain()	59	e eksponent	
delmatrix, subMat()	190, 192	skabelon til	2
deltalist()	51	e til en potens, e^()	62, 68
deltaTmpCnv()	51	E, eksponent	240
DelVar, slette variabel	51	e, viser udtryk i termer af	68
delVoid(), fjerner ugyldige		e^(), e til en potens	62
elementer	52	eff), omregn nominel til effektiv	
derivative()	52	rente	62
deSolve(), løsning	52	effektiv rente, eff()	62
det(), matrixdeterminant	55	egenværdi, eigVl()	63
diag(), matrixdiagonal	55	egenvektor, eigVc()	63
differentialkvotient eller		eigVc(), egenvektor	63
differentialkvotienten af nte		eigVl(), egenværdi	63
orden		eksakt, exact()	67
skabelon til	6	eksponent, E	240
differentialkvotienter		eksponenter	
første differentialkvotient, d()	234	skabelon til	1
numerisk differentialkvotient,		eksponentiel regression, ExpReg	71
nDeriv()	129	ellers hvis, Elseif	64
numerisk differentialkvotient,		ellers, Else	92
nDerivative()	128	Elseif, ellers hvis	64
dim(), dimension	56	end	
dimension, dim()	56	for, EndFor	77
Disp, vis data	56, 167	løkke...EndLoop	116
DispAt	56	EndWhile, slut while-kompleks	213
distributionsfunktioner		enheder	
binomCdf()	21	konvertere	245
binomPdf()	21	enhedsvektor, unitV()	209
normCdf()	132	ensbetydende med, ⇔	233
normPdf()	132	EOS ligningsoperativsystem	
poissCdf()	140	(Equation Operating	
poissPdf()	140	System)	270
tCdf()	196	Equation Operating System (EOS)	
tPdf()	202	ligningsoperativsystem	270
χ^2 2way()	25	etiket, Lbl	101
χ^2 Cdf()	25	euler(), Euler function	65
χ^2 GOF()	26	exact(), eksakt	67
χ^2 Pdf()	26	Exit, afslut	67
dividere, P	225	exp(), e til en potens	68
division af heltal, intDiv()	96	expHist(), udtryk til liste	69
domain(), definitions­mængde-		expand(), ledform	69
kommando	59	expr(), streng til udtryk	71, 113
dominantTerm(), dominerende led	60	ExpReg, eksponentiel regression	71
dominerende led, dominantTerm()	60		

F		funktioner og variable	
factor(), opløse i faktorer	72	kopiere	31
fællesnævner, comDenom()	28	fyld	258-259
fakultet, !	234	G	
fejlkoder og -meddelelser	284	g, nygrader	241
fejlmædelse og fejlfinding		gå til, Goto	92
slette fejl, ClrErr	27	gcd(), største fælles divisor	82
videresende fejl, PassErr	139	gennemsnitlig ændringshastighed,	
Fill, matrixudfyldning	74	avgRC()	17
finansfunktioner, tvnFV()	205	genveje, tastatur	268
finansfunktioner, tvnI()	205	geomCdf()	83
finansfunktioner, tvnN()	205	geomPdf()	83
finansfunktioner, tvnPmt()	206	Get	83, 260
finansfunktioner, tvnPv()	206	getDenom(), hente/returnere	
FiveNumSammendrag	75	nævner	84
fjerdegradsregression, QuartReg	149	getKey()	84
floor(), runde ned	76	getLangInfo(), hent/returner	
flyt, shift()	173	sproginformation	88
fMax(), funktionsmaksimum	76	getLockInfo(), tester låsestatus af en	
fMin(), funktionsminimum	77	variabel eller	
For	77	variabelgruppe	88
For, for	77	getModel(), hente	
for, For	77	tilstandsindstillinger	89
fordelingsfunktioner		getNum(), hente/returnere tal	90
binomCdf()	98	GetStr	90
invNorm()	99	getType(), get type of variable	90
invT()	99	getVarInfo(), hent/returner	
Invx ² ()	97	variabeloplysninger	91
format(), formateret streng	78	Goto, gå til	92
formateret streng, format()	78	grader/minutter/sekunder-notation	242
forskellig fra, ≠	230	gradnotation, g	242
første orden		grænseværdi	
skabelon til	5	skabelon til	7
fortegn, sign()	175	grupper, låse og låse op	112, 209
fpart(), funktionsdel	79	grupper, teste låsestatus	88
freqTable()	79	H	
frequency()	80	heltal, int()	96
Frobenius-norm, norm()	131	heltalsdel, iPart()	99
F-test- med 2 målinger	80	hente/returnere	
Func, funktion	81	nævner, getDenom()	84
Func, programfunktion	81	tal, getNum()	90
funktioner		variabeloplysninger, getVarInfo()	88, 91
brugerdefinerede	49	henvisning, #	240
del, fpart()	79	henvisningsoperator (#)	271
maksimum, fMax()	76		
minimum, fMin()	77		
programfunktion, Func	81		

hexadecimal		K	
indikator, Oh	249	kombinationer, nCr()	127
vise, ►Base16	20	kommentar, ©	249
højre, right()	97, 160-161	kompleks	
hvis, If	92	faktor, cFactor()	23
hvisFn()	94	løse, cSolve()	40
hyperbolsk		nulpunkter, cZeros()	45
arctangens, tanh ⁻¹ ()	195	komplekst	
arcuscosinus, cosh ⁻¹ ()	35	konjugat, conj()	30
arcussinus, sinh ⁻¹ ()	179	konstant	
cosinus, cosh()	34	i solve()	182
sinus, sinh()	178	konstanter	
tangens, tanh()	195	genveje til	268
		i cSolve()	42
I		i cZeros()	47
i streng, inString()	96	i deSolve()	53
identitetsmatrix, identity()	92	i solve()	184
identity(), identitetsmatrix	92	Konstruer matrix, constructMat()	30
If, hvis	92	konverter	
imag(), imaginær del	95	►Rad	151
imaginær del, imag()	95	konvertere	
ImpDif(), implicit		4Grad	92
differentialkvotient	95	enheder	245
implicit differentialkvotient, Impdif()	95	kopiere variabel eller funktion,	
indstille		CopyVar	31
tilstand, setMode()	171	korrelationsmatrix, corrMat()	31
indstillinger, hente aktuelle	89	kuglevektor vise, ►Sphere	185
Input, input	95	kvadratrod	
input, Input	95	skabelon til	1
inString(), i streng	96	kvadratrod, v()	186, 237
int(), heltal	96		
intDiv(), division af heltal	96	L	
integrere, f	235	længde af streng	56
interpolate(), interpolation	97	lagre	
invers kumuleret normalfordeling		symbol, &	248-249
(invNorm()	99	låse variable og variabelgrupper	112
invers, $\wedge^{-1}$	246	Lbl, etiket	101
invF()	97	lcm, mindste fælles multiplum	101
invNorm(), invers kumuleret		ledform, expand()	69
normalfordeling)	99	left(), venstre	101
invf()	99	LibPriv	50
InvX ² ()	97	LibPub	50
iPart(), heltalsdel	99	libShortcut(), danner genveje til	
irr(), intern rente		biblioteksobjekter	102
intern rente, irr()	99	lig med, =	229
isPrime(), primtalstest	100	ligningssystem (2-ligninger)	
isVoid(), test for ugyldighed	100	skabelon til	3

ligningssystem (N-ligning)		logistisk regression, LogistikD	115
skabelon til	3	lokal variabel, Local	112
limit		lokal, Local	112
lim()	103	løkke, Loop	116
limit()	103	Loop, løkke	116
limit() eller lim(), limit	103	løs, solve()	181
Lineær reg t-test	107	løsning, deSolve()	52
lineær regression, LinRegAx	105	LU, matrix opløsning i øvre-nedre	117
lineær regression, LinRegBx	104, 106		
LinRegBx, lineær regression	104	M	
LinRegMx, lineær regression	105	mærke, '	244
LinRegtIntervals, lineær regression	106	maksimum, max()	118
linSolve()	108	matHist(), matrix til liste	117
listMat(), liste til matrix	109	matricer	
liste til matrix, listeMat()	109	cumulative sum, cumulativeSum	
liste, tælle elementer betinget	37	()	44
liste, tælle elementer i	37	delmatrix, subMat()	190, 192
lister		determinant, det()	55
cumulative sum, cumulativeSum		diagonal, diag()	55
()	44	dimension, dim()	56
differens, @list()	109	egenværdi, eigVl()	63
differenser i en liste, @list()	109	egenvektor, eigVc()	63
liste til matrix, listMat()	109	identitet, identity()	92
maksimum, max()	118	kolonnedimension, colDim()	28
matrix til liste, matHist()	117	kolonnenorm, colNorm()	28
midt-streng, mid()	121	liste til matrix, listMat()	109
minimum, min()	121	maksimum, max()	118
ny, newList()	128	matrix til liste, matHist()	117
prikprodukt, dotP()	61	minimum, min()	121
produkt, product()	146	multiplikation og addition af	
sortere faldende, SortD	185	rækker, mRowAdd()	123
sortere stigende, SortA	184	ny, newMat()	128
summation, sum()	191	opløsning i øvre-nedre, LU	117
tomme elementer i	266	produkt, product()	146
udtryk til liste, expHist()	69	punktum addition, .+	227
udvid/sammenkæd, augment()	16	punktum division, ./	228
vektorprodukt, crossP()	39	punktum multiplikation, .*	228
ln(), naturlig logaritme	110	punktum potens, .^	228
LnReg, logaritmisk regression	110	punktum subtraktion, .-	227
Local, lokal variabel	112	QR opløse i faktorer, QR	147
Lock, låse variabel eller		rækkeoperation, mRow()	123
variabelgruppe	112	reduceret række-echelonform,	
Log		rref()	165
skabelon til	2	summation, sum()	191
logaritmer	110	transponere, T	193
logaritmisk regression, LnReg	110	udfylde, Fill	74
Logistik, logistisk regression	114	udvid/sammenkæd, augment()	16
LogistikD, logistisk regression	115	vilkårlige, randMat()	153
logistisk regression, Logistik	114		

matrix (1 × 2)		N	
skabelon til	4	nævner	28
matrix (2 × 1)		nand, Boolsk operator	126
skabelon til	4	når, when()	212
matrix (2 × 2)		naturlig logaritme, ln()	110
skabelon til	4	nCr(), kombinationer	127
matrix (m × n)		nDerivative(), numerisk	
skabelon til	4	differentialkvotient	128
matrix til liste, mat►list()	117	negation, indtaste negative tal	271
matrixer		nettonutidsværdi, npv()	134
rækkeaddition, rowAdd()	164	newList(), ny liste	128
rækkedimension, rowDim()	164	newMat(), ny matrix	128
række-echelonform, ref()	155	nfMax(), numerisk	
rækkenorm, rowNorm()	165	funktionsmaksimum	129
rækkeskift, rowSwap()	165	nfMin(), numerisk	
max(), maksimum	118	funktionsminimum	129
mean(), middel	118	nInt(), numerisk integral	130
med, 	246	nom), omregn effektiv til nominel	
medfører, ⇒	233, 268	rente	130
median(), median	119	nominel rente, nom()	130
median, median()	119	nor, Boolsk operator	130
medium-medium linjeregression,		norm(), Frobenius-norm	131
MedMed	120	normal linje, normalLine()	132
MedMed, medium-medium		normalLine()	132
linjeregression	120	normalsandsynlighedsfordeling,	
mens, While	213	normCdf()	132
mid(), midt-streng	121	normCdf()	132
middel, mean()	118	normPdf()	132
midt-streng, mid()	121	not, Boolsk operator	132
min(), minimum	121	nPr(), permutationer	133
mindre end eller lig med, {	231	npv(), nettonutidsværdi	134
mindste fælles multiplum, lcm	101	nSolve(), numerisk løsning	134
minimum, min()	121	nnte rod	
minutnotation,	242	skabelon til	1
mirr(), modifieret intern		nulpunkter, zeroes()	214
forrentning	122	numerisk	
mod(), modulo	123	differentialkvotient, nDerivative()	128
modifieret intern forrentning, mirr(		differentialkvotient, nDeriv() ..	129
)	122	integral, nInt()	130
modulo, mod()	123	løsning, nSolve()	134
mRow(), matrixrækkeoperation	123	ny	
mRowAdd(), multiplikation og		liste, newList()	128
addition af matrixrækker ..	123	matrix, newMat()	128
Multipel lineær regression ttest	125	nygrader notation, g	241
multiplicere, *	224		
MultReg	124		
MultRegIntervals()	124		
MultRegTests()	125		

O

objekter	
danner genveje til bibliotek	102
OneVar, statistik med en variabel	135
operatorer	
rækkefølge af beregning	270
oplåsning af variable og	
variabelgrupper	209
opløse i faktorer, factor()	72
or (Boolsk), or	137
or, Boolske operatorer	137
ord(), numerisk tegnkode	138

P

P►Rx(), rektangulært x-koordinat	138
P►Ry(), rektangulært y-koordinat	138
PassErr, videresende fejl	139
Pdf()	79
permutationer, nPr()	133
piecewise()	140
poissCdf()	140
poissPdf()	140
polær	
koordinat, R►Pr()	151
koordinat, R►Pθ()	151
vektorvisning, ►Polar	140
polyCoef()	141
polyDegree()	142
polyEval(), beregne polynomium	142
polyGcd()	143
polynomier	
beregne, polyEval()	142
vilkårlige, randPoly()	153
PolyRoots()	144
potens, ^	226
PotensReg, potensregression	144
potensregression,	
PowerReg	144, 157, 159, 198
Prgm, definere program	145
primtalstest, isPrime()	100
procent, %	229
prodSeq()	146
product(), produkt	146
produkt Π()	
skabelon til	5
produkt, Π()	237
produkt, product()	146

programmer	
definere privat bibliotek	50
programmer og programmering	
slette fejl, ClrErr	27
vis I/O-skærm, Disp	56
vis I/O-skærm, disp.	167
programmering	
definere program, Prgm	145
videresende fejl, PassErr	139
vis data, Disp	56, 167
propFrac, ægte brøk	147
punktum	
addition, .+	227
division, ./	228
multiplikation, *	228
potens, .^	228
subtraktion, .-	227

Q

QR opløse i faktorer, QR	147
QuadReg, andengradsregression	148
QuartReg, fjerdegradsregression	149

R

R, radian	241
R►Pr(), polær koordinat	151
R►Pθ(), polær koordinat	151
radian, R	241
række, series()	170
række-echelonform, ref()	155
rand(), vilkårligt tal	152
randBin, vilkårligt tal	152
randInt(), vilkårligt heltal	152
randMat(), vilkårlig matrix	153
randNorm(), vilkårlig norm	153
randPoly(), vilkårligt polynomium	153
randSamp()	153
RandSeed, vilkårligt tal-seed	154
real(), reel	154
real, reel()	154
reciprok, ^-1	246
reduceret række-echelonform, rref()	165
ref(), række-echelonform	155
RefreshProbeVars	156
regressioner	
andengrads, QuadReg	148
eksponentiel, ExpReg	71

fjerdegrads, QuartReg	149	()	
lineær regression, LinRegAx	105	sec(), sekans	166
lineær regression, LinRegBx	104	sec ⁻¹ (), invers sekans	166
logistiske, Logistik	115	sech(), hyperbolsk sekans	166
potensregression,		sech ⁻¹ (), invers hyperbolsk sekans ..	167
PowerReg ... 144, 157, 159, 198		sekundnotation, "	242
sinus, SinReg	179	sekvens, seq()	168
tredjegrads, CubicReg, CubicReg	43	seq(), sekvens	168
Regressioner		seqGen()	168
lineær regression, LinRegBx	106	seqn()	169
logaritmisk, LnReg	110	sequence, seq()	168-169
Logistisk	114	series(), række	170
medium-medium linje, MedMed	120	setMode(), indstil tilstand	171
MultReg	124	shift(), flyt	173
rektangulær vektorvisning, ►Rect ..	154	sign(), fortegn	175
rektangulært x-koordinat, P►Rx() ..	138	simult(), sammenhørende ligninger ..	175
rektangulært y-koordinat, P►Ry() ..	138	sin(), sinus	177
remain(), rest	157	sin ⁻¹ (), arcussinus	178
RequestStr	159	sinh(), hyperbolsk sinus	178
rest, remain()	157	sinh ⁻¹ (), hyperbolsk arcussinus	179
resultat		SinReg, sinusregression	179
vis i termer af cosinus	31	sinus	
viser i termer af e	68	viser udtryk ud fra	176
viser ud fra sinus	176	sinus, sin()	177
resultat (sidste), ans	13	sinusregression, SinReg	179
resultater, statistiske	187	skabeloner	
resultatværdier, statistik	188	Absolut værdi	3-4
Return, returner	160	anden afledede	6
returner, Return	160	bestemt integral	6
right, right()	29, 65, 211	brøk	1
rk23(), Runge Kutta-funktion	161	differentialkvotient eller	
rotate(), roter	162	differentialkvotienten	
roter, rotate()	162	af nte orden	6
round(), afrund	164	e eksponent	2
rowAdd(), addition af matrixrækker ..	164	eksponent	1
rowDim(), matrixrækkedimension ..	164	første orden	5
rowNorm(), matrixrækkenorm	165	grænseværdi	7
rowSwap(), matrixrækkeskift	165	kvadratrods	1
rref(), reduceret række-		ligningssystem (2-ligninger)	3
echelonform	165	ligningssystem (N-ligning)	3
rund op, ceiling()	21	Log	2
runde ned, floor()	76	matrix (1 × 2)	4
runde op, ceiling()	22, 38	matrix (2 × 1)	4
Ryd	254	matrix (2 × 2)	4
		matrix (m × n)	4
		nte rod	1
		produkt $\prod$ ()	5
		stykkevis funktion (2 stykker) ..	2
		stykkevis-skabelon (N stykker) ..	3
S			
sammenhørende ligninger, simult() ..	175		
sandsynlighedsdensitivitet, normPdf ..	132		

sum $\Sigma()$	5	streng	
ubestemt integral	6	dimension, dim()	56
slet		længde	56
fejlf, ClrErr	27	streng	
slette		anvende til at skabe	
ugyldige elementer fra liste	52	variabelnavne	271
variabel, DelVar	51	flyt, shift()	173
slut		formater	78
funktion, EndFunc	81	formater, format()	78
hvis, EndIf	92	henvi, #	240
mens, EndWhile	213	højre, right()	97, 160-161
slut på hvis, EndIf	92	i, inString	96
slut while-kompleks, EndWhile	213	midt-streng, mid()	121
solve(), løs	181	roter, rotate()	162
som af rentebetalinger	238	streng til udtryk, expr()	71, 113
SortA, sorter stigende	184	tegnkode, ord()	138
SortD, sortere faldende	185	tegnstreng, char()	24
sortere		tilføj, &	234
faldende, SortD	185	udtryk til streng, string()	190
stigende, SortA	184	venstre, left()	101
sprog		string(), udtryk til streng	190
hent sproginformation	88	strings	
sqrt(), kvadratrodd	186	right, right()	29, 65, 211
standardafvigelse, stdDev()	188-189, 209	student-t sandsynlighedsfordeling,	
stat.results	187	tCdf()	196
stat.values	188	student-t sandsynlighedstæthed,	
statistik		tPdf()	202
fakultet, !	234	stykkewis funktion (2 stykker)	
kombinationer, nCr()	127	skabelon til	2
median, median()	119	stykkewis-skabelon (N stykker)	
middel, mean()	118	skabelon til	3
permutationer, nPr()	133	subMat(), delmatrix	190, 192
standardafvigelse, stdDev() 188-189, 209		substitution med " " -operator	246
statistik med en variabel,		subtrahere, -	223
OneVar	135	sum $\Sigma()$	
to-variabel-resultater, TwoVar ..	207	skabelon til	5
varians, variance()	210	sum af hovedstolsbetalinger	239
vilkårlig norm, randNorm()	153	sum(), summation	191
vilkårligt tal-seed, RandSeed	154	sum, S()	238
statistik med en variabel, OneVar	135	sumif()	191
stdDevPop(), population		summation, sum()	191
standardafvigelse	188	sumSeq()	192
stdDevSamp(), standardafvigelse for			
måling	189	T	
Stop-kommando	190	T (transponere)	193
større end eller lig med, 	232	t test, tTest	203
større end, >	232	tælle dage mellem datoer, dbd()	47
største fælles divisor, gcd()	82	tælle elementer i en liste betinget,	37

countif()		tredjegradsregression, CubicReg	43
tælle elementer i en liste, count() ..	37	trigonometrisk ekspansion, tExpand()	197
tan(), tangens	193	trigonometrisk opsamling, tCollect() ..	197
tan ⁻¹ (), arcustangens	194	Try, fejlhåndteringskommando	202
tangens, tan()	193	tTest, t test	203
tangent linje, tangentLine()	195	tTest_2Samp, to-målingers t test ...	204
tangentLine()	195	TVM-argumenter	206
tanh(), hyperbolsk tangens	195	tvmFV()	205
tanh ⁻¹ (), hyperbolsk arcustangens ..	195	tvmI()	205
tastaturgenveje	268	tvmN()	205
taylor(), Taylor-polynomium	196	tvmPmt()	206
Taylor-polynomium, taylor()	196	tvmPV()	206
tCdf(), student t sandsynlighedsfordeling ...	196	TwoVar, to-variabel-resultater	207
tCollect(), trigonometrisk opsamling	197		
tegn	255-257	U	
numerisk kode, ord()	138	ubestemt integral	
streng, char()	24	skabelon til	6
tegnstreng, char()	24	udelukkelse med " " -operator	246
test for ugyldighed, isVoid()	100	udtryk	
Test_2S, F test med 2 målinger	80	streng til udtryk, expr()	71, 113
tExpand(), trigonometrisk ekspansion	197	udtryk til liste, exprHist()	69
ekspansion	197	udvid/sammenkæd, augment()	16
Text-kommando	198	ugyldige elementer	266
tidsdiskonterede pengeværdier, aktuel værdi	206	ugyldige elementer, fjerne	52
tidsdiskonterede pengeværdier, antal betalinger	205	ugyldighed, test for	100
tidsdiskonterede pengeværdier, betalingsbeløb	206	understregning, _	244
tidsdiskonterede pengeværdier, Fremtidig værdi	205	unitV(), enhedsvektor	209
tidsdiskonterede pengeværdier, Rente	205	unLock, oplåse variabel eller variabel gruppe	209
tiende potens, 10^()	246		
tilføj, &	234	V	
tilnærmet, approx()	14	variabel	
tilstande		oprette navn fra en tegnstreng ..	271
indstille, setMode()	171	slette alle enkeltbogstaver	27
tilstandsindstillinger, getMode() ...	89	variable	
tInterval, t konfidensinterval	199	lokal, Local	112
tInterval_2Samp, to-målingers t konfidensinterval	199	slette, DelVar	51
tmpCnv()	200-201	variable og funktioner	
tomme (ugyldige) elementer	266	kopiere	31
to-variabel-resultater, TwoVar	207	variable, låse og låse op	88, 112, 209
tPdf(), student sandsynlighedstæthed	202	varians, variance()	210
transponere, T	193	varPop()	209
		varSamp(), målingsvarians	210
		vektorer	
		cylindrisk vektorvisning, ►Cylind	45
		enhed, unitV()	209
		prikkprodukt, dotP()	61

vektorprodukt, crossP()	39	zTest	219
vektorprodukt, crossP()	39	zTest_1Prop, enproportional z test ..	220
venstre, left()	101	zTest_2Prop, toproportional z test ..	220
Vent på kommando	211	zTest_2Samp, to-prøvers z test	221
videresende fejl, PassErr	139		
vilkårlig		Δ	
matrix, randMat()	153	Δ list(), listedifferens	109
vilkårlig norm, randNorm()	153		
vilkårlig stikprøve	153	X	
vilkårligt		χ^2 2way	25
tal-seed, RandSeed	154	χ^2 Cdf()	25
vinkel, angle()	10	χ^2 GOF	26
vis data, Disp	56, 167	χ^2 Pdf()	26
vis som			
rektangulær vektor, ►Rect	154		
vis som			
binær, ►Base2	18		
cylindrisk vektor, ►Cylind	45		
decimal vinkel, ►DD	48		
decimalt heltal, ►Base10	19		
grader/minutter/sekunder,			
►DMS	58		
hexadecimal, ►Base16	20		
kuglevektor, ►Sphere	185		
polær vektor, ►Polar	140		
Viser grader/minutter/sekunder,			
►DMS	58		
W			
warnCodes(), Warning codes	211		
when(), når	212		
While, mens	213		
X			
x^2 , kvadrat	227		
XNOR	233		
xor, Boolsk exclusive or	213		
Z			
zeroes(), nulpunkter	214		
zInterval, z konfidensinterval	216		
zInterval_1Prop, en-proportionalt z			
konfidensinterval	217		
zInterval_2Prop, to-proportionalt z			
konfidensinterval	218		
zInterval_2Samp, z konfidensinterval			
med to-målinger	218		