TI-Nspire™ CX CAS
Opslagsvejledning

Få mere at vide om TI Technology via online-hjælpen på education.ti.com/eguide.

Vigtige oplysninger

Medmindre andet udtrykkeligt angives i den Licens, der følger med et program, stiller Texas Instruments ingen garantier, hverken udtrykkeligt eller underforstået, herunder men ikke begrænset til underforståede garantier om salgbarhed og egnethed til et bestemt formål, for programmer eller skriftligt materiale, og Texas Instruments stiller udelukkende sådant materiale til rådighed, som det foreligger. Texas Instruments kan under ingen omstændigheder holdes ansvarlig for særlige, indirekte, hændelige eller følgeskader i forbindelse med eller som følge af køb eller brug af dette materiale, og hele Texas Instruments' erstatningsansvar kan, uanset søgsmålets art, ikke overstige det beløb, der fremgår af programlicensen. Derudover kan Texas Instruments ikke holdes ansvarlig for nogen form for krav som følge af en anden parts brug af dette materiale.

© 2020 Texas Instruments Incorporated

De faktiske produkter kan variere let fra de viste billeder.

Indholdsfortegnelse

Udtryksskabeloner **1**

Alfabetisk oversigt **8**

A 8
B 17
C 21
D 46
E 60
F 70
G 80
I 91
L 99
M 115
N 125
O 132
P 135
Q 144
R 147
S 162
T 187
U 203
V 204
W 205
X 207
Z 208

Symboler **217**

TI-Nspire™ CX II - Tegn kommandoer **244**

Grafikprogrammering 244
Grafikskærm 244
Standardvisning og indstillinger 245
Fejlmeddelelser på grafikskærmen 246
Ugyldige kommandoer i grafiktilstand 246
C 247
D 248
F 251
G 253
P 254
S 256
U 258

Tomme (ugyldige) elementer **259**

Genveje til indtastning af matematiske udtryk **261**

Hierarkiet i EOS™ ligningsoperativsystemet (Equation Operating System) **263**

TI-Nspire CX II - TI-Grundlæggende programmeringsfunktioner **265**

Automatisk indrykning i programmeringseditoren 265
Forbedrede fejlmeddelelser til TI-Basic 265

Konstanter og værdier **268**

Fejlkoder og fejlmeddelelser **269**

Fejlkoder og -meddelelser **277**

Generelle oplysninger **279**

Online hjælp 279
Kontakt TI-Support 279
Service og garanti 279

Indeks **280**

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 219.

Eksempel:

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 220.

Eksempel:

Kvadratrodsskabelon

[ctrl] [x²]-taster

Bemærk: Se også **√() (kvadratrod)**, side 230.

Eksempel:

Nte rod-skabelon

[ctrl] [^]-taster

Bemærk: Se også **root()**, side 158.

Eksempel:

Nte rod-skabelon

ctrl ^ -taster

$\sqrt[3]{8}$ → 2

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

e ekponentskabelon

$e^{\square}$ e^x -taster

Den naturlige eksponentialfunktion *e* opløftet til en potens

Bemærk: Se også **e^()**, side 60.

Eksempel:

e^1 → e

$e^{1.}$ → 2.71828182846

Log-skabelon

$\log_{\square}(\square)$ ctrl 10^x -tasten

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

Bemærk: Se også **log()**, side 111.

Eksempel:

$\log_4(2.)$ → 0.5

Stykkevis-skabelon (2 stykker)

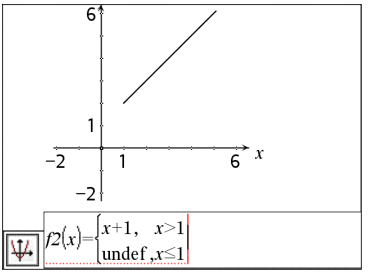
Katalog >

$\begin{cases}\square, & \square \\ \square, & \square\end{cases}$

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 **piecewise()**, side 136.

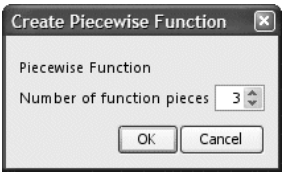
Eksempel:



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 136.

Eksempel:

Se eksemplet med stykkevis-skabelonen (2 stykker).

Skabelon til system med 2 ligninger

Katalog >

$\begin{cases} \square \\ \square \end{cases}$

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 187.

Eksempel:

$\text{solve}\left(\begin{cases} x+y=0 \\ x-y=5 \end{cases}, x, y\right) \quad x=\frac{5}{2} \text{ and } y=\frac{-5}{2}$

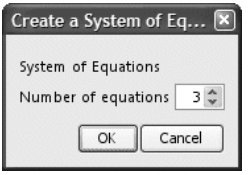
$\text{solve}\left(\begin{cases} y=x^2-2 \\ x+2\cdot y=-1 \end{cases}, x, y\right)$

$x=\frac{-3}{2} \text{ and } y=\frac{1}{4} \text{ or } x=1 \text{ and } y=-1$

Skabelon til system med N ligninger

Katalog >

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



Bemærk: Se også **system()**, side 187.

Eksempel:

Se eksemplet med ligningssystemskabelonen (2-ligninger).

Absolut værdi-skabelon

Katalog >

$|\square|$ **Bemærk:** Se også **abs()**, side 8.

Eksempel:

Absolut værdi-skabelon

Katalog >

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

dd°mm'ss.ss''-skabelon

Katalog >

□°□'□"

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.

Eksempel:

$30°15'10"$ $\frac{10891\cdot\pi}{64800}$

Matrix-skabelon (2 x 2)

Katalog >

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

Opretter en matrix 2 x 2.

Eksempel:

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

Matrix-skabelon (1 x 2)

Katalog >

$\begin{bmatrix}\square & \square\end{bmatrix}$.

Eksempel:

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

Matrix-skabelon (2 x 1)

Katalog >

$\begin{bmatrix}\square\\ \square\end{bmatrix}$

Eksempel:

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

Matrix-skabelon (m x n)

Katalog >

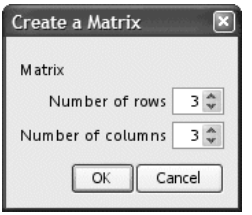
Skabelonen vises, efter at du er blevet bedt om at angive antallet af rækker og kolonner.

Eksempel:

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

Matrix-skabelon (m x n)

Katalog >



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

Sum-skabelon (Σ)

Katalog >

$$\sum_{\square=\square}^{\square}(\square)$$

Eksempel:

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

Bemærk: Se også **Σ() (sumSeq)**, side 232.

Produkt-skabelon (Π)

Katalog >

$$\prod_{\square=\square}^{\square}(\square)$$

Eksempel:

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

Bemærk: Se også **Π() (prodSeq)**, side 231.

Skabelon til differentialkvotient af første orden

Katalog >

$$\frac{d}{d\square}(\square)$$

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

Eksempel:

$$\frac{d}{dx}\left(x^3\right) \qquad 3\cdot x^2$$

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

Skabelon til differentialkvotient af første orden

Katalog >

Bemærk: Se også **d() (differentialkvotient)**, side 228.

Skabelon til differentialkvotient af anden orden

Katalog >

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

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

Bemærk: Se også **d() (differentialkvotient)**, side 228.

Eksempel:

$$\frac{d^2}{dx^2}\left(x^3\right) \qquad 6\cdot x$$

$$\frac{d^2}{dx^2}\left(x^3\right)\Big|_{x=3} \qquad 18$$

Skabelon til differentialkvotient af Nte orden

Katalog >

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

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

Bemærk: Se også **d() (differentialkvotient)**, side 228.

Eksempel:

$$\frac{d^3}{dx^3}\left(x^3\right)\Big|_{x=3} \qquad 6$$

Bestemt integral skabelon

Katalog >

$$\int_{\square}^{\square} \square\, d\square$$

Bemærk: Se også **∫() integral()**, side 217.

Eksempel:

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

Ubestemt integralskabelon

Katalog >

$$\int \square\, d\square$$

Eksempel:

Ubestemt integralskabelon

Katalog >

Bemærk: Se også ∫**() integral()**, side 217.

$$\int x^2 \, dx \qquad \frac{x^3}{3}$$

Grænseværdi skabelon

Katalog >

$$\lim_{\square \to \square^{\square}} (\square)$$

Anvend − eller (−) til grænseværdi fra venstre. Anvend + til grænseværdi fra højre.

Bemærk: Se også **limit()**, side 101.

Eksempel:

$$\lim_{x \to 5^{\square}} (2 \cdot x + 3) \qquad 13$$

Alfabetisk oversigt

Elementer, hvis navne ikke er alfabetiske (som f.eks. +, ! og >), er anført sidst i dette afsnit, startende (side 217). 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(*Udtr1***)⇒***udtryk*

abs(*Liste1***)⇒***liste*

abs(*Matrix1***)⇒***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.

$$\left|\left\{\frac{\pi}{2},\frac{-\pi}{3}\right\}\right| \qquad \left\{\frac{\pi}{2},\frac{\pi}{3}\right\}$$

$$|2-3\cdot i| \qquad \sqrt{13}$$

$$|z| \qquad |z|$$

$$|x+y\cdot i| \qquad \sqrt{x^2+y^2}$$

amortTbl() Katalog >

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

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 201).

- 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

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

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 **ΣInt()** og **ΣPrn()**, side 232 og **bal()**, side 17.

Boolsk Udtr1 **and** *Boolsk Udtr2* ⇒*Boolsk udtryk*

Boolsk Liste1 **and** *Boolsk Liste2*⇒*Boolsk liste*

Boolsk Matrix1 **and** *Boolsk Matrix2*⇒*Boolsk matrix*

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

$x\geq 3$ and $x\geq 4$ → $x\geq 4$

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

Heltal1 **and** *Heltal2*⇒*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:

37 and 0b100 → 4

and

Katalog >

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).

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

angle()

Katalog >

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:

$$\text{angle}(0+2\cdot i) \qquad 90$$

I vinkeltilstanden Nygrader:

$$\text{angle}(0+3\cdot i) \qquad 100$$

I vinkeltilstanden Radian:

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

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

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

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

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

Katalog >

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 variablen *stat.results*. (side 182.)

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

Katalog >

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 variablen *stat.results*. (side 182.)

LevRow=0 for Blok

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

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
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

Output-variabel	Beskrivelse
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⇒*værdi*

Returnerer resultatet af de sidst beregnede udtryk.

56	56
56+4	60
60+4	64

approx()

Katalog >

approx(*Udtr1***)⇒***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 ctrl enter.

approx(*Liste1***)⇒***liste*

approx(*Matrix1***)⇒***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.}

$\text{approx}([\sqrt{2}\quad\sqrt{3}])$ → [1.41421 1.73205]

$\text{approx}\left(\begin{bmatrix}\frac{1}{3} & \frac{1}{9}\end{bmatrix}\right)$ → [0.333333 0.111111]

$\text{approx}(\{\sin(\pi),\cos(\pi)\})$ → {0.,-1.}

$\text{approx}([\sqrt{2}\quad\sqrt{3}])$ → [1.41421 1.73205]

▸approxFraction()

Katalog >

Expr **▸approxFraction(**[*Tol*]**)⇒***udtryk*

List **▸approxFraction(**[*Tol*]**)⇒***liste*

Matrix **▸approxFraction(**[*Tol*]**)⇒***matrix*

Returnere inputtet som en brøk mede 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

0.83333333333333▸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.

$\text{approxRational}(0.333,5\cdot10^{-5})$ → $\frac{333}{1000}$

$\text{approxRational}(\{0.2,0.33,4.125\},5.E-14)$ → $\left\{\frac{1}{5},\frac{33}{100},\frac{33}{8}\right\}$

arccos() **Se cos⁻¹(), side 33.**

arccosh() **Se cosh⁻¹(), side 34.**

arccot() **Se cot⁻¹(), side 35.**

arccoth() **Se coth⁻¹(), side 36.**

arccsc() **Se csc⁻¹(), side 39.**

arccsch() **Se csch⁻¹(), side 39.**

arcLen() Katalog >

arcLen(*Udtr1*,*Var*,*Start*,*Slut***)** ⇒*udtryk*

Returnerer buelængden for *Udtr1* fra *Start* til *Slut* med hensyn til variablen *Var*.

Buelængden beregnes som et integral, der forudsætter er funktionsdefinition.

arcLen(*Liste1*,*Var*,*Start*,*Slut***)**⇒*liste*

Returnerer en liste med buelængder for hvert element i *Liste1* fra *Start* til *Slut* med hensyn til *Var*.

$$\text{arcLen}(\cos(x),x,0,\pi) \qquad 3.8202$$

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

$$\text{arcLen}(\{\sin(x),\cos(x)\},x,0,\pi) \qquad \{3.8202,3.8202\}$$

arcsec() **Se sec⁻¹(), side 162.**

arcsech() **Se sech^{-1}(), side 163.**

arcsin() **Se sin^{-1}(), side 173.**

arcsinh() **Se sinh^{-1}(), side 174.**

arctan() **Se tan^{-1}(), side 188.**

arctanh() **Se tanh^{-1}(), side 190.**

augment() Katalog >

augment(*Liste1, Liste2***)⇒***liste***

Returnerer en ny liste, der er *liste2* føjet til enden af *Liste1*.

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

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*.

$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} \rightarrow m1$ $\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}$

$\begin{bmatrix}5\\6\end{bmatrix} \rightarrow m2$ $\begin{bmatrix}5\\6\end{bmatrix}$

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

avgRC() Katalog >

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

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

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

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

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 variablen.

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

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

$\text{avgRC}(f(x),x,h)$	$\frac{f(x+h)-f(x)}{h}$
$\text{avgRC}(\sin(x),x,h)\|x=2$	$\frac{\sin(h+2)-\sin(2)}{h}$
$\text{avgRC}(x^2-x+2,x)$	$2.\cdot(x-0.4995)$
$\text{avgRC}(x^2-x+2,x,0.1)$	$2.\cdot(x-0.45)$
$\text{avgRC}(x^2-x+2,x,3)$	$2\cdot(x+1)$

B

bal() Katalog >

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

bal(*NPmt***,***amortTabel*)⇒*værdi*

Amortiseringsfunktion, der beregner saldo efter en angivet betaling.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen over TVM-argumenter (side 201).

NPmt angiver betalingsnummeret, hvorefter du vil have dataene beregnet.

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

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

$$\begin{bmatrix} 0 & 0. & 0. & 5000. \\ 1 & -23.35 & -825.63 & 4174.37 \\ 2 & -19.49 & -829.49 & 3344.88 \\ 3 & -15.62 & -833.36 & 2511.52 \\ 4 & -11.73 & -837.25 & 1674.27 \\ 5 & -7.82 & -841.16 & 833.11 \\ 6 & -3.89 & -845.09 & -11.98 \end{bmatrix}$$

$\text{bal}(4,tbl)$	1674.27

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen over TVM-argumenter (side 201).

- Hvis du udelader *Pmt*, bliver den som standard *Pmt*=**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.

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

bal(*NPmt,amortTabel***)** beregner saldoen efter betaling nummer *NPmt*, baseret på amortiseringstabel *amortTabel*. *amortTabel*-argumentet skal være en matrix i formen beskrevet under **amortTbl()**, side 8.

Bemærk: Se også **ΣInt()** og **ΣPrn()**, side 232.

▸Base2

Heltal1 **▸Base2**⇒*heltal*

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 hexadecimalt tal kan have op til 16.

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

256▸Base2	0b100000000
0h1F▸Base2	0b11111

▸Base2 Katalog >

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

−1 vises som

0hFFFFFFFFFFFFFFFF 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

▸Base10 Katalog >

Heltal1 ▸**Base10**⇒*heltal*

0b10011▸Base10	19
0h1F▸Base10	31

▸Base10 Katalog >

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

Konverterer *Heltal1* til et decimaltal (i titalssystemet). Binære eller hexadecimale 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.

▸Base16 Katalog >

Heltal1 ▸**Base16**⇒*heltal*

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

Konverterer *Heltal1* til et hexadecimalt tal. Binære eller hexadecimale 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.

256▸Base16	0h100
0b111100001111▸Base16	0hF0F

▶Base16 Katalog >

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() Katalog >

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≤X ≤*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 ≤ *øvreGrænse*), sæt *nedreGrænse*=0

binomPdf() Katalog >

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() Katalog >

ceiling(*Udtr1***)**⇒*heltal*

Returnerer det nærmeste heltal, der er ≥ argumentet.

ceiling(.456) 1.

ceiling() Katalog >

Argumentet kan være et reelt eller komplekst tal.

Bemærk: Se også **floor()**.

ceiling(*Liste***)⇒***liste*

ceiling(*Matrix1***)⇒***matrix*

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

$\text{ceiling}(\{-3.1,1,2.5\})$ $\{-3.,1,3.\}$

$\text{ceiling}\begin{bmatrix}0 & -3.2\cdot i\\ 1.3 & 4\end{bmatrix}$ $\begin{bmatrix}0 & -3.\cdot i\\ 2. & 4\end{bmatrix}$

centralDiff() Katalog >

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

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

Når *Værdi* er angivet, tilsidesætter den alle forudgående variabeltildelinger og alle nuværende "|" substitutioner for variablen.

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

Ved anvendelse af *Liste1* eller *Matrix1* bliver operationen mappet på tværs af værdierne i listen eller på tværs af matrixelementerne.

Bemærk: Se også og **d()**.

$\text{centralDiff}(\cos(x),x,h)$ $\frac{-(\cos(x-h)-\cos(x+h))}{2\cdot h}$

$\lim_{h\to 0}(\text{centralDiff}(\cos(x),x,h))$ $-\sin(x)$

$\text{centralDiff}(x^3,x,0.01)$ $3.\cdot(x^2+0.000033)$

$\text{centralDiff}(\cos(x),x)|x=\frac{\pi}{2}$ $-1.$

$\text{centralDiff}(x^2,x,\{0.01,0.1\})$ $\{2.\cdot x,2.\cdot x\}$

cFactor() Katalog >

cFactor(*Udtr1***[,***Var***])**⇒*udtryk*

cFactor(*Liste1***[,***Var***])**⇒*liste*

cFactor(*Matrix1*[**,***Var*]**)⇒***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-relle 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 variablen *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.

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 udrykkes 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()**.

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

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

$\text{cFactor}\left(x^2+3\right)$
x^2+3

$\text{cFactor}\left(x^2+a\right)$
x^2+a

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

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

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

$\text{cFactor}\left(x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3\right)$
$x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3$

$\text{cFactor}\left(x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3,x\right)$
$(x-0.964673)\cdot(x+0.611649)\cdot(x+2.12543)\cdot(x+$

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.

char(38)	"&"
char(65)	"A"

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*×*n* matrix *A*, betegnet ved $p_A(\lambda)$, er polynomiet defineret ved

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

hvor *I* betegner *n*×*n* identitetsmatrixen.

kvadratMatrix1 og *kvadratMatrix2* skal have de samme dimensioner.

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

χ²2way

Katalog >

χ²2way *obsMatrix*

chi2way *obsMatrix*

Beregner en χ^2 test til association på tovejstabellen med tællinger i den observerede matrix *obsMatrix*. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 182.)

Oplysninger om effekten af tomme elementer i en matrix findes “Tomme (ugyldige) elementer,” side 259.

Output-variabel	Beskrivelse
stat.χ^2	Chi-kvadrat stat: sum (observeret - forventet)^2/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^2Cdf()

Katalog >

χ^2Cdf(*nedreGrænse*,*øvreGrænse*,*df*)⇒*tal*
hvis *nedreGrænse* og *øvreGrænse* er tal, *liste* hvis *nedreGrænse* og *øvreGrænse* er lister

chi2Cdf(*nedreGrænse*,*øvreGrænse*,*df*)⇒*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($X \le$ *øvreGrænse*), sæt *nedreGrænse*=0.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

χ^2GOF

Katalog >

χ^2GOF *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 182.)

χ^2GOF

Katalog >

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

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

χ^2Pdf()

Katalog >

χ^2Pdf($XVal$**,**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 259.

ClearAZ

Katalog >

ClearAZ

Sletter alle enkelttegnsvariable 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 204.

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

ClrErr

Katalog

ClrErr

Sletter fejlstatus og indstiller systemvariabel *errCode* til nul.

Se et eksempel på **ClrErr**, i Eksempel 2 under **Try**-kommandoen, side 197.

ClrErr

Katalog

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 **PassErr**, side 136, og **Try**, side 197.

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

colAugment()

Katalog >

colAugment(*Matrix1***, *Matrix2*)⇒*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*.

$$\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\rightarrow m1 \qquad \begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$$

$$\begin{bmatrix}5 & 6\end{bmatrix}\rightarrow m2 \qquad \begin{bmatrix}5 & 6\end{bmatrix}$$

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

colDim()

Katalog >

colDim(*Matrix***)⇒*udtryk***

Returnerer antallet af kolonner i *Matrix*.

Bemærk: Se også **rowDim()**.

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

colNorm()

Katalog >

colNorm(*Matrix***)⇒*udtryk***

Returnerer maksimum for summerne af de absolutte værdier for elementerne i kolonnerne i *Matrix*.

Bemærk: Udefinerede matrixelementer er ikke tilladt. Se også **rowNorm()**.

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

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

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 at udelade *Var* sparer dette ofte tid, hukommelse og skærmplads, 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 uudviklet tæller og nævner. Sådanne resultater sparer normalt endnu mere tid, hukommelse og skærmplads. Sådanne resultater, der er delvist opløst i faktorer, gør også de efterfølgende operationer på resultatet hurtigere og mindre tilbøjelige til at fylde hukommelsen op.

Define $comden(exprn)=\text{comDenom}(exprn,abc)$
Done

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

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

$$comden(1234\cdot x^2\cdot(y^3-y)+2468\cdot x\cdot(y^2-1))$$
$$1234\cdot x\cdot(x\cdot y+2)\cdot(y^2-1)$$

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

completeSquare ()

Katalog >

completeSquare(*ExprOrEqn*, *Var*)⇒*udtryk eller ligning*

completeSquare(*ExprOrEqn*, *Var^Power*)⇒*udtryk eller ligning*

completeSquare(*ExprOrEqn*, *Var1, Var2 [,...]*)⇒*udtryk eller ligning*

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

Konverterer et kvadratisk, polynomielt udtryk af formen a·x²+b·x+c til formen a·(x-h)²+k

— eller —

Konverterer en kvadratisk ligning af formen a·x²+b·x+c=d til formen a·(x-h)²=k

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

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 kvadratetkomplettere med hensyn til variable *Var1, Var2* [,...]).

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

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

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

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

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

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

conj()

Katalog >

conj(*Udtr1*)⇒*udtryk*

conj(*Liste1*)⇒*liste*

conj(*Matrix1*)⇒*matrix*

Returnerer kompleks konjugerede af argumentet.

Bemærk: Alle udefinerede variable behandles som reelle variable.

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

$\text{conj}\begin{pmatrix}2 & 1-3\cdot i\\ -i & -7\end{pmatrix} \quad \begin{bmatrix}2 & 1+3\cdot i\\ i & -7\end{bmatrix}$

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

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

constructMat() Katalog >

constructMat
(*Udtr*,*Var1*,*Var2*,*antalRækker*,*antalKol*)
⇒*matrix*

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*.

$\text{constructMat}\left(\frac{1}{i+j}, i, j, 3, 4\right)$

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}$$

CopyVar Katalog >

CopyVar *Var1*, *Var2*

CopyVar *Var1*., *Var2*.

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

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.

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

aa.a:=45	45
aa.b:=6.78	6.78
CopyVar *aa.*,*bb.*	*Done*
getVarInfo()	*aa.a* "NUM" "" 0 *aa.b* "NUM" "" 0 *bb.a* "NUM" "" 0 *bb.b* "NUM" "" 0

CopyVar

Katalog >

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.

corrMat()

Katalog >

corrMat(Liste1,*Liste2[***,…[,Liste20**]])

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

▸cos

Katalog >

Udtr **▸cos**

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

Repræsentrer *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(...) modulo 1−cos(...)^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(...) ku optræder i det givne udtryk med lige potenser.

Bemærk: Konverteringsoperatorer 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.

$(\sin(x))^2 \blacktriangleright \cos \qquad 1-(\cos(x))^2$

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:

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

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden Radian:

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

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 (li) og egenvektorer (Vi) af A.

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

Dan matricerne:

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()

trig-tast

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

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.

$\cos^{-1}()$

trig-tast

$\cos^{-1}$(*Udtr1***)⇒***udtryk*

$\cos^{-1}$(*Liste1***)⇒***liste*

$\cos^{-1}$(*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***)⇒***kvadratMatrix***

Returnerer 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()**.

I vinkeltilstanden Grader:

$\cos^{-1}(1)$ → 0

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:

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

cos⁻¹() trig-tast

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.

cosh() Katalog >

cosh(*Udtr1***)⇒***udtryk*

cosh(*Liste1***)⇒***liste*

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*

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.

I vinkeltilstanden Grader:

$$\cosh\left(\left(\frac{\pi}{4}\right)^{r}\right) \qquad \cosh(45)$$

I vinkeltilstanden Radian:

$$\cosh\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix}$$

$$\begin{bmatrix}421.255 & 253.909 & 216.905\\ 327.635 & 255.301 & 202.958\\ 226.297 & 216.623 & 167.628\end{bmatrix}$$

cosh⁻¹() Katalog >

cosh⁻¹(*Udtr1***)⇒***udtryk*

cosh⁻¹(*Liste1***)⇒***liste*

$\cosh^{-1}(1)$	0
$\cosh^{-1}(\{1,2.1,3\})$	$\{0,1.37286,\cosh^{-1}(3)\}$

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() Katalog >

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:

$$\cosh^{-1}\left(\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix}\right)$$

$$\begin{bmatrix}2.52503+1.73485\cdot i & -0.009241-1.49086\\ 0.486969-0.725533\cdot i & 1.66262+0.623491\\ -0.322354-2.08316\cdot i & 1.26707+1.79018\end{bmatrix}$$

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

cot() trig-tast

cot(*Udtr1***)** ⇒ *udtryk*

cot(*Liste1***)** ⇒ *liste*

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.

I vinkeltilstanden Grader:

$$\cot(45) \qquad 1$$

I vinkeltilstanden Nygrader:

$$\cot(50) \qquad 1$$

I vinkeltilstanden Radian:

$$\cot(\{1,2.1,3\}) \qquad \left\{\frac{1}{\tan(1)},-0.584848,\frac{1}{\tan(3)}\right\}$$

cot⁻¹() trig-tast

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.

I vinkeltilstanden Grader:

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

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden Radian:

cot⁻¹() trig-tast

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

$\cot^{-1}(1)$ → $\frac{\pi}{4}$

coth() Katalog >

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*.

$\coth(1.2)$ → 1.19954

$\coth(\{1,3.2\})$ → $\left\{\frac{1}{\tanh(1)}, 1.00333\right\}$

coth⁻¹() Katalog >

coth⁻¹(*Udtr1***)⇒***udtryk*

coth⁻¹(*Liste1***)⇒***liste*

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(…)**.

$\coth^{-1}(3.5)$ → 0.293893

$\coth^{-1}(\{-2,2.1,6\})$ →

$$\left\{\frac{-\ln(3)}{2}, 0.518046, \frac{\ln\left(\frac{7}{5}\right)}{2}\right\}$$

count() Katalog >

count(*Værdi1ellerListe1* [,*Værdi2ellerListe2* [,…]]**)⇒***værdi*

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

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

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

$\text{count}(2,4,6)$ → 3

$\text{count}(\{2,4,6\})$ → 3

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

$\text{count}\left(\frac{1}{2}, 3+4\cdot i, \text{undef}, \text{"hello"}, x+5., \text{sign}(0)\right)$ → 2

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.

count() Katalog >

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 259.

countif() Katalog >

countif(*Liste*,*Kriterie***)⇒***værdi*

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

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 259.

Bemærk: Se også **sumIf()**, side 186, og **frequency()**, side 78.

countIf({1,3,"abc",undef,3,1},3) 2

Tæller antallet af elementer lig med 3.

countIf({"abc","def","abc",3},"def") 1

Tæller antallet af elementer lig med "def."

countIf({x^{-2},x^{-1},1,x,x^2},x) 1

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

countIf({1,3,5,7,9},?<5) 2

Tæller 1 og 3.

countIf({1,3,5,7,9},2<?<8) 3

Tæller 3, 5 og 7.

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

Tæller 1, 3, 7 og 9.

cPolyRoots() Katalog >

cPolyRoots(*Poly*,*Var*)⇒*liste*

cPolyRoots(*ListeAfKoeff*)⇒*liste*

Den første syntaks, **cPolyRoots(***Poly*,*Var*), returnerer en liste med komplekse rødder af polynomiet *Poly* med hensyn til variablen *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 141.

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

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

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

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.

crossP($\{a1,b1\},\{a2,b2\}$) $\{0,0,a1\cdot b2-a2\cdot b1\}$

crossP($\{0.1,2.2,-5\},\{1,-0.5,0\}$) $\{-2.5,-5.,-2.25\}$

crossP($[1\ \ 2\ \ 3],[4\ \ 5\ \ 6]$) $[-3\ \ 6\ \ -3]$

crossP($[1\ \ 2],[3\ \ 4]$) $[0\ \ 0\ \ -2]$

csc() trig-tast

csc(*Udtr1*)⇒*udtryk*

csc(*Liste1*)⇒*liste*

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

I vinkeltilstanden Grader:

csc(45) $\sqrt{2}$

I vinkeltilstanden Nygrader:

csc()

trig-tast

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

I vinkeltilstanden Radian:

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

csc^{-1}()

trig-tast

csc^{-1}(*Udtr1***)** ⇒ *udtryk*

csc^{-1}(*Liste1***)** ⇒ *liste*

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(...)`**.

I vinkeltilstanden Grader:

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

I vinkeltilstanden Nygrader:

$\csc^{-1}(1)$ 100.

I vinkeltilstanden Radian:

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

csch()

Katalog >

csch(*Udtr1***)** ⇒ *udtryk*

csch(*Liste1***)** ⇒ *liste*

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

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

$\operatorname{csch}(\{1,2.1,4\})$ $\left\{\frac{1}{\sinh(1)},0.248641,\frac{1}{\sinh(4)}\right\}$

csch^{-1}()

Katalog >

csch^{-1}(*Udtr1***)** ⇒ *udtryk*

csch^{-1}(*Liste1***)** ⇒ *liste*

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

$\operatorname{csch}^{-1}(\{1,2.1,3\})$ $\left\{\sinh^{-1}(1),0.459815,\sinh^{-1}\left(\frac{1}{3}\right)\right\}$

csch^{-1}() Katalog >

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

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

cSolve() Katalog >

cSolve(*Ligning***,** *Var***)⇒***Boolsk udtryk*

cSolve(*Ligning***,** *Var*=*Gæt***)⇒***Boolesk udtryk*

cSolve(*Ulighed***,** *Var***)⇒***Boolsk udtryk*

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 polynnomiel faktoropløsning.

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

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

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

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

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

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

I Vis cifre-tilstand for Fast 2:

$\text{exact}(\text{cSolve}(x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3=0,x))$

$x\cdot(x^4+4\cdot x^3+5\cdot x^2-6)=3$

$\text{cSolve}(Ans,x)$

$x=-1.11+1.07\cdot i \text{ or } x=-1.11-1.07\cdot i \text{ or } x=-2.$

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

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:

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 initielle 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.

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

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.

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

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 ck, hvor k er et heltalssuffix fra 1 til og med 255.

$$\text{cSolve}\left(u\cdot v-u=v \text{ and } v^2=-u,\{u,v,w\}\right)$$
$$u=\frac{1}{2}+\frac{\sqrt{3}}{2}\cdot i \text{ and } v=\frac{1}{2}-\frac{\sqrt{3}}{2}\cdot i \text{ and } w=c43 \text{ or}$$

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-polynomiel 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.

$$\text{cSolve}\left(e^z=w \text{ and } w=z^2,\{w,z=1+i\}\right)$$
$$w=0.149606+4.8919\cdot i \text{ and } z=1.58805+1.5402$$

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 polynomiel tredjegradsregression y = a ·x3+b · x2+c ·x+dpå listerne X og Y med frekvens $Frekv$. En sammenfatning af resultaterne lagres i *stat.resultat* variable. (side 182.)

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.

Kategory 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 259.

Output-variabel	**Beskrivelse**
stat.RegEqn	Regressionsligning: a ·x^3+b ·x^2+c ·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 *X-liste*, der faktisk bruges i regressionen ud fra begrænsninger af *Hyppighed*, *Kategoriliste*, og *Medtag kategorier*
stat.YReg	Liste af datapunkter i den modificerede *Y-liste*, der faktisk bruges i regressionen ud fra begrænsninger af *Hyppighed*, *Kategoriliste*, og *Medtag kategorier*
stat.FreqReg	Liste med hyppigheder, der svarer til *stat.XReg* og *stat.YReg*

cumulativeSum() Katalog >

cumulativeSum(*Liste1***)⇒***liste*

cumulativeSum({1,2,3,4}) {1,3,6,10}

cumulativeSum() Katalog >

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.

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 259.

$$\begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix} \rightarrow m1 \qquad \begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix}$$

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

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→temp
             For i,1,100,1
             If i=50
             Cycle
             temp+i→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∠θ, z].

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

$$\begin{bmatrix}2 & 2 & 3\end{bmatrix}\text{▸Cylind} \qquad \begin{bmatrix}2\cdot\sqrt{2} & \angle\frac{\pi}{4} & 3\end{bmatrix}$$

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*

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+*i*også.

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*].

I Vis cifre-tilstand med Fast 3:

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

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

$$\text{cZeros}\left(\left\{u\cdot v-u-v,v^2+u\right\},\{u,v\}\right)$$
$$\begin{bmatrix} 0 & 0 \\ \frac{1}{2}-\frac{\sqrt{3}}{2}\cdot i & \frac{1}{2}+\frac{\sqrt{3}}{2}\cdot i \\ \frac{1}{2}+\frac{\sqrt{3}}{2}\cdot i & \frac{1}{2}-\frac{\sqrt{3}}{2}\cdot i \end{bmatrix}$$

Udtræk række 2:

$$Ans[2] \qquad \begin{bmatrix} \frac{1}{2}-\frac{\sqrt{3}}{2}\cdot i & \frac{1}{2}+\frac{\sqrt{3}}{2}\cdot i \end{bmatrix}$$

cZeros() Katalog >

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

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

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

$$\text{cZeros}\left(\{u\cdot v-u-v, v^2+u\},\{u,v,w\}\right)$$
$$\text{cZero}\left(\{u\cdot (v-1)-v, u+v^2\},\{u,v,w\}\right)$$
$$\begin{bmatrix} 0 & 0 & c4 \\ \frac{1}{2}-\frac{\sqrt{3}}{2}\cdot i & \frac{1}{2}+\frac{\sqrt{3}}{2}\cdot i & c4 \\ \frac{1}{2}+\frac{\sqrt{3}}{2}\cdot i & \frac{1}{2}-\frac{\sqrt{3}}{2}\cdot i & c4 \end{bmatrix}$$

For polynomielle systemer afhænger beregningstiden eller hukommelsesforbruget stærkt af den rækkefølge, de ubekendte angives i. Hvis det initielle 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.

$$\text{cZeros}\left(\{u+v-e^w, u-v-i\},\{u,v\}\right)$$
$$\begin{bmatrix} \frac{e^w+i}{2} & \frac{e^w-i}{2} \end{bmatrix}$$

Hvis et system hverken er polynomielt 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.

$$\text{cZeros}\left(\{e^z-w, w-z^2\},\{w,z\}\right)$$
$$\begin{bmatrix} 0.494866 & -0.703467 \end{bmatrix}$$

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.

$$\text{cZeros}\left(\{e^z-w, w-z^2\},\{w,z=1+i\}\right)$$
$$\begin{bmatrix} 0.149606+4.8919\cdot i & 1.58805+1.54022\cdot i \end{bmatrix}$$

D

dbd() Katalog >

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

dbd()

Katalog >

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*

Liste1 ▸**DD**⇒*liste*

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 Grader:

$(1.5°)\blacktriangleright DD \quad 1.5°$

$(45°22'14.3")\blacktriangleright DD \quad 45.3706°$

$(\{45°22'14.3",60°0'0"\})\blacktriangleright DD \quad \{45.3706°,60°\}$

I vinkeltilstanden Nygrader:

$1\blacktriangleright DD \quad \frac{9}{10}^{g}$

I vinkeltilstanden Radian:

$(1.5)\blacktriangleright DD \quad 85.9437°$

▸Decimal

Katalog >

Udtr1 ▸**Decimal**⇒*udtryk*

Liste1 ▸**Decimal**⇒*udtryk*

Matrix1 ▸**Decimal**⇒*udtryk*

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

$\frac{1}{3}\blacktriangleright Decimal \quad 0.333333$

▸Decimal

Katalog >

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

Define

Katalog >

Define *Var* = *Udtryk*

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

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

Define *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.

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**).

Define $g(x,y)=$Func
If $x>y$ Then
Return x
Else
Return y
EndIf
EndFunc
	Done

$g(3,-7)$	3

Define

Katalog >

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 49 og **Define LibPub**, side 49.

```
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 48, og **Define LibPub**, side 49.

Define LibPub

Katalog >

Define LibPub *Var* **=** *Udtryk*

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

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

Define LibPub Katalog >

...) = **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 48 og **Define LibPriv**, side 49.

deltaList() Se ΔList(), side 107.

deltaTmpCnv() Se ΔtmpCnv(), side 195.

DelVar Katalog >

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

DelVar *Var*.

Sletter de angivne variable, eller variabelgruppe fra hukommelse.

Hvis en eller flere af variablene er låst, viser denne kommando en fejlmeddelelse og sletter kun de ulåste variable. Se **unLock,** side 204

$2 \rightarrow a$	2
$(a+2)^2$	16
DelVar a	*Done*
$(a+2)^2$	$(a+2)^2$

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
getVarInfo()	$\begin{bmatrix} aa.a & "NUM" & "" \\ aa.b & "NUM" & "" \\ aa.c & "NUM" & "" \end{bmatrix}$
DelVar *aa*.	*Done*
getVarInfo()	"NONE"

delVoid()

Catalog >

delVoid(*Liste1***)⇒***liste*

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

Yderligere oplysninger om tomme elementer findes på side 259.

delVoid({1,void,3})	{1,3}

derivative()

Se *d*(), side 228.

deSolve()

Katalog >

deSolve(*1.Eller2.OrdenODE***,** *Var***,** *afhVar***)** ⇒*en generel løsning*

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.

deSolve($y''+2\cdot y'+y=x^2,x,y$)	$y=(c3\cdot x+c4)\cdot e^{-x}+x^2-4\cdot x+6$
right(Ans)→*temp*	$(c3\cdot x+c4)\cdot e^{-x}+x^2-4\cdot x+6$
$\frac{d^2}{dx^2}(temp)+2\cdot\frac{d}{dx}(temp)+temp-x^2$	0
DelVar *temp*	*Done*

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

Den generelle løsning til en 1. grads ligning indeholder en arbitrær konstant af formen $_c$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.OrdenODE* **and** *startBeting*, *Var*, *afhVar*) ⇒*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. x0 og y0, der ikke har nogen lagrede værdier. Implicit differentiation kan hjælpe med at verificere implicitte løsninger.

$\text{deSolve}\left(y'=\left(\cos(y)\right)^2\cdot x, x, y\right) \quad \tan(y)=\frac{x^2}{2}+c4$

$\text{solve}(Ans, y) \quad y=\tan^{-1}\left(\frac{x^2+2\cdot c4}{2}\right)+n3\cdot\pi$

$Ans \mid c4=c-1 \text{ and } n3=0 \quad y=\tan^{-1}\left(\frac{x^2+2\cdot(c-1)}{2}\right)$

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

$\text{deSolve}(ode \text{ and } y(0)=0, x, y) \rightarrow soln \quad \frac{-\left(2\cdot\sin(y)+y^2\right)}{2}=-\left(e^x-1\right)\cdot e^{-x}\cdot\sin(y)$

$soln \mid x=0 \text{ and } y=0$	true
$ode \mid y'=\text{impDif}(soln, x, y)$	true
DelVar *ode*,*soln*	*Done*

deSolve
(
2.OrdensODE
and*startBeting1***and***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*

$$\text{deSolve}\left(y''=y^{\frac{-1}{2}} \text{ and } y(0)=0 \text{ and } y'(0)=0,t,y\right)$$
$$\frac{2\cdot y^{\frac{3}{4}}}{3}=t$$

$$\text{solve}\left(\frac{2\cdot y^{\frac{3}{4}}}{3}=t,y\right)$$
$$y=\frac{3^{\frac{1}{3}}\cdot 2^{\frac{2}{3}}\cdot t^{\frac{4}{3}}}{4} \text{ and } t\geq 0$$

deSolve
(
2.OrdensODE
and*begrBeting1***and***begrBeting2*, *Var*, *depVar*)⇒ *en partikulær løsning*

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

$$\text{deSolve}\left(y''=x \text{ and } y(0)=1 \text{ and } y'(2)=3,x,y\right)$$
$$y=\frac{x^3}{6}+x+1$$

$$\text{deSolve}\left(y''=2\cdot y' \text{ and } y(3)=1 \text{ and } y'(4)=2,x,y\right)$$
$$y=e^{2\cdot x-8}-e^{-2}+1$$

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

det() Katalog >

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

Returnerer determinanten af *kvadratMatrix*.

Ethvert matrixelement 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 ctrl enter 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 ·**max(dim(***kvadratMatrix***))**▫**rowNorm(***kvadratMatrix***)**

$\det\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ → $a\cdot d-b\cdot c$

$\det\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ → -2

$\det\left(\text{identity}(3)-x\cdot\begin{bmatrix} 1 & -2 & 3 \\ -2 & 4 & 1 \\ -6 & -2 & 7 \end{bmatrix}\right)$ → $-\left(98\cdot x^3-55\cdot x^2+12\cdot x-1\right)$

$\begin{bmatrix} 1.\text{E}20 & 1 \\ 0 & 1 \end{bmatrix}\rightarrow mat1$ → $\begin{bmatrix} 1.\text{E}20 & 1 \\ 0 & 1 \end{bmatrix}$

$\det(mat1)$ → 0

$\det(mat1,.1)$ → $1.\text{E}20$

diag() Katalog >

diag(*List***)⇒***matrix*

diag(*rækkeMatrix***)⇒***matrix*

diag(*kolonneMatrix***)⇒***matrix*

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

$\text{diag}([2\ \ 4\ \ 6])$ → $\begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix}$

diag(*kvadratMatrix***)⇒***rækkeMatrix*

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

kvadratMatrix skal være kvadratisk.

$\begin{bmatrix} 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$ → $\begin{bmatrix} 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$

$\text{diag}(Ans)$ → $[4\ \ 2\ \ 9]$

dim() Katalog >

dim(*Liste***)⇒***heltal*

$\dim(\{0,1,2\})$ → 3

dim()

Katalog >

Returnerer dimensionen af *liste*.

dim(matrix)⇒*liste*

Returnerer dimensionerne af matricen som en liste med to elementer {rækker, kolonner}.

$$\dim\begin{bmatrix}1 & -1\\ 2 & -2\\ 3 & 5\end{bmatrix} \quad \{3,2\}$$

dim(*Streng*)⇒*heltal*

Returnerer det antal tegn, der er indeholdt i tegnstrengen *Streng*.

```
dim("Hello")                5
dim("Hello "&"there")       11
```

Disp

Katalog >

Disp *udtrykEllerStreng1* [**,** *udtrykEllerStreng2*] ...

Viser argumenterne i *Calculator* historikken. Argumenterne vises efter hinanden med små mellemrum som separator.

Anvendes hovedsagelig i programmer og funktioner til at sikre at mellemregninger vises.

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

```
Define chars(start,end)=Prgm
                  For i,start,end
                  Disp i," ",char(i)
                  EndFor
                  EndPrgm
                                  Done

chars(240,243)
                                240 ð
                                241 ñ
                                242 ò
                                243 ó
                                  Done
```

DispAt

Katalog >

DispAt *int*,*expr1* [,*expr2* ...] ...

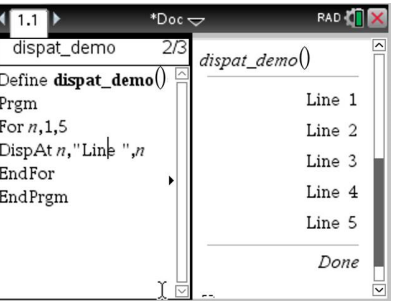
DispAt tillader dig at angive den linje, hvor det specificerede udtryk eller streng vil blive vist på skærmen.

Linjeantallet kan angives som et udtryk.

Vær opmærksom på, at linjenummeret ikke gælder hele skærmen, men for området umiddelbart efter kommando/program.

```
DispAt
```

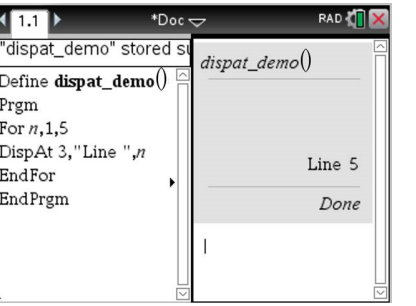
Eksempel



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

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

EndFor EndPrgm	

Fejlbetingelser:

Fejlmeddelelse DispAt linjeantal skal være mellem 1 og 8	**Beskrivelse** Udtryk evaluerer linjeantallet udenfor rækken 1-8 (inklusive)
For få argumenter	Funktionen eller kommandoen mangler et eller flere argumenter.
Ingen argumenter	Samme som aktuel 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.

▸DMS **Katalog >**

Udtr **▸DMS**

List **▸DMS**

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 236.

I vinkeltilstanden Grader:

$(45.371)\blacktriangleright\text{DMS}$ $45°22'15.6"$

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

▸DMS

Katalog >

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

domain()

Katalog >

domain(*udtryk1***,** *Var***)⇒***udtryk*

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

domain() kan bruges til at undersøge definitionsmængden 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 ∫() ikke er en tilladt funktion.

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

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

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

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

$$\text{domain}\left(\int_1^x \frac{1}{t}\,dt,x\right) \blacktriangleright \text{domain}\left(\int_1^x \frac{1}{t}\,dt,x\right)$$

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 −∞, 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 = ∞ eller −∞.

Hvis rækken eller en af dens afledte afbrydes af et spring ved *Punkt*, indeholder resultatet sandsynligvis deludtryk på formen sign(…) eller abs(…) for en reel udviklingsvariabel eller $(-1)^{\text{floor}(\ldots\text{angle}(\ldots)\ldots)}$ 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 "| *Var* > *Punkt*", "| *Var* < *Punkt*", "| "*Var* ≥ *Punkt*", eller "*Var* ≤ *Punkt*" for at opnå et enklere resultat.

$\text{dominantTerm}\left(\tan\left(\sin\left(x\right)\right)-\sin\left(\tan\left(x\right)\right),x\right)$ $\frac{x^7}{30}$

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

$\text{dominantTerm}\left(x^{-2}\cdot\tan\left(\frac{1}{3}\right),x\right)$ $\frac{1}{x^{\frac{5}{3}}}$

$\text{dominantTerm}\left(\ln\left(x^x-1\right)\cdot x^{-2},x\right)$ $\frac{\ln\left(x\cdot\ln\left(x\right)\right)}{x^2}$

$\text{dominantTerm}\left(e^{\frac{-1}{z}},z\right)$

$\text{dominantTerm}\left(e^{\frac{-1}{z}},z,0\right)$

$\text{dominantTerm}\left(\left(1+\frac{1}{n}\right)^n,n,\infty\right)$ e

$\text{dominantTerm}\left(\tan^{-1}\left(\frac{1}{x}\right),x,0\right)$ $\frac{\pi\cdot\text{sign}\left(x\right)}{2}$

$\text{dominantTerm}\left(\tan^{-1}\left(\frac{1}{x}\right),x\right)\left|x>0\right.$ $\frac{\pi}{2}$

dominantTerm() Katalog >

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 $Var \rightarrow Punkt$. **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 166.

dotP() Katalog >

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^() [e^x]-tast

e^(*Udtr1***)⇒*udtryk***

Returnerer **e** opløftet til potensen *Udtr1*.

e^1	e
$e^{1.}$	2.71828
e^{3^2}	e^9

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 $re^{i\theta}$ polær form. Anvend dog kun denne form i vinkeltilstanden Radian. Den forårsager en domænefejl i vinkeltilstandene Grader eller Nygrader.

e^(*Liste1***)⇒*liste***

$e^{\{1,1.,0.5\}}$	$\{e,2.71828,1.64872\}$

e^()

e-tast

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.

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

eff()

Katalog >

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 128.

$$\text{eff}(5.75,12) \quad 5.90398$$

eigVc()

Katalog >

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, ..., x_n], så:

$x_1^2 + x_2^2 + ... + x_n^2 = 1$

I rektangulært komplekst format:

$$\begin{bmatrix}-1 & 2 & 5\\ 3 & -6 & 9\\ 2 & -5 & 7\end{bmatrix} \rightarrow m1 \quad \begin{bmatrix}-1 & 2 & 5\\ 3 & -6 & 9\\ 2 & -5 & 7\end{bmatrix}$$

$$\text{eigVc}(m1)$$

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

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

eigVc() Katalog >

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.

eigVl() Katalog >

eigVl(*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:

$$\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix} \rightarrow m1 \qquad \begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix}$$

$$\text{eigVl}(m1)$$

$$\{-4.40941, 2.20471+0.763006\cdot i, 2.20471-0.$$

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

Else Se If, side 91.

ElseIf Katalog >

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≤-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
```

EndFor **Se For, side 76.**

EndFunc **Se Func, side 80.**

EndIf **Se If, side 91.**

EndLoop **Se Loop, side 114.**

EndPrgm **Se Prgm, side 142.**

EndTry **Se Try, side 197.**

EndWhile **Se While, side 207.**

euler ()

Katalog >

euler(*Expr*, *Var*, *depVar*, {*Var0, VarMax*}, *depVar0*, *VarStep* [, *eulerStep*]**)** ⇒*matrix*

euler(*SystemOfExpr*, *Var*, *ListOfDepVars*, {*Var0*, *VarMax*}, *ListOfDepVars0*, *VarStep* [, *eulerStep*]**)** ⇒*matrix*

euler(*ListOfExpr*, *Var*, *ListOfDepVars*, {*Var0*, *VarMax*}, *ListOfDepVars0*, *VarStep* [, *eulerStep*]**)** ⇒*matrix*

Anvender Euler-metoden til at løse systemet

Differentialligning:

y'=0.001*y*(100-y) og y(0)=10

$$\text{euler}\left(0.001\cdot y\cdot\left(100-y\right),t,y,\left\{0,100\right\},10,1\right)$$
$$\begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 10. & 10.9 & 11.8712 & 12.9174 & 14.042 \end{bmatrix}$$

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

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

med $depVar(Var0)=depVar0$ i intervallet $[Var0, VarMax]$. 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.

{$Var0$, $VarMax$} 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 **sign(**$VarStep$**)** = **sign(**$VarMax$-$Var0$**)** og løsninger returneres i $Var0$+i·$VarStep$ for alle i=0,1,2,... således at $Var0$+i·$VarStep$ 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 $VarStep/eulerStep$.

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 $y2(0)=5$

$$\text{euler}\left(\begin{cases} -y1+0.1\cdot y1\cdot y2 \\ 3\cdot y2-y1\cdot y2 \end{cases}, t, \{y1, y2\}, \{0,5\}, \{2,5\}, 1\right)$$

$$\begin{bmatrix} 0. & 1. & 2. & 3. & 4. & 5. \\ 2. & 1. & 1. & 3. & 27. & 243. \\ 5. & 10. & 30. & 90. & 90. & -2070. \end{bmatrix}$$

eval(*Expr***)** ⇒ *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.

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 81), **GetStr** (side 88), og **Send** (side 163).

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.

```
fadein()                                   Done
```

```
n:=0.25                                    0.25
m:=8                                       8
n·m                                        2.
Send "SET COLOR.BLUE ON TIME eval(n·m)"
                                           Done
iostr.SendAns    "SET COLOR.BLUE ON TIME 2"
```

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).

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

Exit

Katalog >

Exit

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

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.

Funktionsliste:

Define $g()$=Func
Local *temp*,*i*
$0\rightarrow$*temp*
For *i*,1,100,1
temp+*i*$\rightarrow$*temp*
If *temp*>20 Then
Exit
EndIf
EndFor
EndFunc
	Done

$g()$	21

▸exp

Katalog >

Udtr **▸exp**

Repræsenterer *Udtr* i termer af den naturlige eksponentialfunktion *e*. Det er en konverteringsoperator til visning. Denne operator kan kun anvendes ved slutningen af indtastningslinjen.

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

$\frac{d}{dx}\left(e^{x}+e^{-x}\right)$	$2\cdot\sinh(x)$
$2\cdot\sinh(x)$▸exp	$e^{x}-e^{-x}$

exp()

ex-tast

exp(*Udtr1***)⇒***udtryk*

Returnerer **e** opløftet til potensen *Udtr1*.

Returnerer **e** opløftet til potensen *Værdi1*.

Bemærk: Se også **e** eksponentskabelon, side 2.

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^{1} → e

$e^{1.}$ → 2.71828

e^{3^2} → e^{9}

exp(*Liste1***)⇒***liste*

Returnerer **e** opløftet til potensen af hvert element i *Liste1*.

$e^{\{1,1.,0.5\}}$ → $\{e, 2.71828, 1.64872\}$

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()**.

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

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

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()**.

$\text{solve}(x^2-x-2=0,x)$ → $x=-1 \text{ or } x=2$

$\text{exp▸list}(\text{solve}(x^2-x-2=0,x),x)$ → $\{-1,2\}$

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(…)**.

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.

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 skærmplads, 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.

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

$x^2+2\cdot x\cdot y+2\cdot x+y^2+2\cdot y+1$

$\text{expand}\left(\frac{x^2-x+y^2-y}{x^2\cdot y^2-x^2\cdot y-x\cdot y^2+x\cdot y}\right)$

$\frac{1}{x-1}-\frac{1}{x}+\frac{1}{y-1}-\frac{1}{y}$

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

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

$\text{expand}\left(\frac{x^2-x+y^2-y}{x^2\cdot y^2-x^2\cdot y-x\cdot y^2+x\cdot y},y\right)$

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

$\text{expand}(Ans,x)$ $\frac{1}{x-1}-\frac{1}{x}+\frac{1}{y\cdot(y-1)}$

$\text{expand}\left(\frac{x^3+x^2-2}{x^2-2}\right)$ $\frac{2\cdot x}{x^2-2}+x+1$

$\text{expand}(Ans,x)$ $\frac{1}{x-\sqrt{2}}+\frac{1}{x+\sqrt{2}}+x+1$

expand()

Katalog >

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.

$\ln(2\cdot x\cdot y)+\sqrt{2\cdot x\cdot y}$ $\quad \ln(2\cdot x\cdot y)+\sqrt{2\cdot x\cdot y}$

$\text{expand}(Ans)$ $\quad \ln(x\cdot y)+\sqrt{2}\cdot\sqrt{x\cdot y}+\ln(2)$

$\text{expand}(Ans)\,|\,y\geq 0$ $\quad \ln(x)+\sqrt{2}\cdot\sqrt{x}\cdot\sqrt{y}+\ln(y)+\ln(2)$

$\text{sign}(x\cdot y)+|x\cdot y|+e^{2\cdot x+y}$ $\quad e^{2\cdot x+y}+\text{sign}(x\cdot y)+|x\cdot y|$

$\text{expand}(Ans)$ $\quad \text{sign}(x)\cdot\text{sign}(y)+|x|\cdot|y|+\left(e^{x}\right)^2\cdot e^{y}$

expr()

Katalog >

expr(*Streng***)⇒***udtryk*

Returnerer den tegnstreng, der er indeholdt i *Streng* som et udtryk og og eksekverer den straks.

$\text{expr}("1+2+x\text{^}2+x")$ $\quad x^2+x+3$

$\text{expr}("\text{expand}((1+x)\text{^}2)")$ $\quad x^2+2\cdot x+1$

"Define cube(x)=x^3" → *funcstr* $\quad$ "Define cube(x)=x^3"

$\text{expr}(funcstr)$ $\quad$ *Done*

$cube(2)$ $\quad 8$

ExpReg

Katalog >

ExpReg *X*, *Y* [, [*Frekv*][, *Kategori*, *Medtag*]]

Beregner polynomielle tredjegradsregression y = a ·(b)ˣpå listerne *X* og *Y* med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 182.)

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 259.

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
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 *X-liste*, der faktisk bruges i regressionen ud fra begrænsninger af *Hyppighed*, *Kategoriliste* og *Medtag kategorier*
stat.YReg	Liste af datapunkter i den modificerede *Y-liste*, der faktisk bruges i regressionen ud fra begrænsninger af *Hyppighed*, *Kategoriliste* og *Medtag kategorier*
stat.FreqReg	Liste med hyppigheder, der svarer til *stat.XReg* og *stat.YReg*

F

factor() **Katalog >**

factor(*Udtr1*[**,** *Var*]**)**⇒*udtryk*

factor(*Liste1*[**,***Var*]**)**⇒*liste*

factor(*Matrix1*[**,***Var*]**)**⇒*matrix*

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

$\text{factor}\left(a^3 \cdot x^2 - a \cdot x^2 - a^3 + a\right)$	$a \cdot (a-1) \cdot (a+1) \cdot (x-1) \cdot (x+1)$
$\text{factor}\left(x^2+1\right)$	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

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 variablen *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.

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.

$\text{factor}\left(a^3\cdot x^2-a\cdot x^2-a^3+a,x\right)$

$a\cdot\left(a^2-1\right)\cdot(x-1)\cdot(x+1)$

$\text{factor}\left(x^2-3,x\right)$ $\left(x+\sqrt{3}\right)\cdot\left(x-\sqrt{3}\right)$

$\text{factor}\left(x^2-a,x\right)$ $\left(x+\sqrt{a}\right)\cdot\left(x-\sqrt{a}\right)$

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

$x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3$

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

$(x-0.964673)\cdot(x+0.611649)\cdot(x+2.12543)\cdot(x^$

factor() Katalog >

factor(*rationaltTal***)** 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 vare længere end en dag, og opløsning af et 100-cifret tal kan vare længere end et århundrede.

factor(152417172689)	123457·1234577
isPrime(152417172689)	false

Sådan stopper du en beregning manuelt,

- **Håndholdt:** Hold tasten [on] nede, mens du gentagne gange trykker på [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 *RationaltTal* ikke er et primtal, og den næststørste faktor har mere end fem cifre.

FCdf() Katalog >

FCdf
(
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*.

FCdf() Katalog >

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

Fill Katalog >

Fill *Udtr, matrixVar⇒matrix*

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

matrixVar skal eksistere i forvejen.

Fill *Udtr, listeVar⇒liste*

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

Listevar skal eksistere i forvejen.

$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} \rightarrow amatrix$ $\quad \begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}$

Fill 1.01,*amatrix* $\quad$ *Done*

amatrix $\quad \begin{bmatrix}1.01 & 1.01\\1.01 & 1.01\end{bmatrix}$

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

Fill 1.01,*alist* $\quad$ *Done*

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

FiveNumSammendrag Katalog >

FiveNumSummary *X*[**,** [*Frekvv*]**[** *Kategori*,*Medtag*]]

Frembringer en forkortet version af 1-variabelstatistikken på listen *X*.
En sammenfatning af resultaterne lagres i *stat.results* variable. (side 182.)

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.

FiveNumSammendrag

Katalog >

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 259

Output-variabel	Beskrivelse
stat.MinX	Minimum af x-værdier
stat.Q_1X	1. kvartil af x
stat.MedianX	Median af x
stat.Q_3X	3. kvartil af x
stat.MaxX	Maksimum af x-værdier

floor()

Katalog >

floor(*Udtr1)*⇒*heltal*

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

Argumentet kan være et reelt eller komplekst tal.

floor(−2.14) → −3.

floor(*Liste1***)**⇒*liste*

floor(*Matrix1***)**⇒*matrix*

Returnerer en liste eller matrix med nedrunding af hvert element.

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

floor({3/2, 0, −5.3}) → {1, 0, −6.}

floor([1.2 3.4; 2.5 4.8]) → [1. 3.; 2. 4.]

fMax()

Katalog >

fMax(*Udtr, Var***)**⇒*Boolsk udtryk*

fMax(*Udtr***,** *Var***,***nedreGrænse***)**

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

fMax(*Udtr***,** *Var***) |** *nedreGrænse*≤*Var* ≤*øvreGrænse*

fMax(1−(x−a)²−(x−b)², x) → x=(a+b)/2

fMax(.5·x³−x−2, x) → x=∞

fMax() Katalog >

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.

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()**.

$\text{fMax}(0.5 \cdot x^3 - x - 2, x) \mid x \le 1 \qquad x = -0.816497$

fMin() Katalog >

fMin(*Udtr, Var***)⇒***Boolsk udtryk*

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

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

fMin(*Udtr***,** *Var***) |** *nedreGrænse*≤*Var* ≤*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()**.

$\text{fMin}(1-(x-a)^2-(x-b)^2, x) \qquad x = -\infty \text{ or } x = \infty$

$\text{fMin}(0.5 \cdot x^3 - x - 2, x) \mid x \ge 1 \qquad x = 1.$

For

Katalog >

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.

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 Local *tempsum*,*step*,*i* $0 \rightarrow$ *tempsum* $1 \rightarrow$ *step* For *i*,1,100,*step* *tempsum*+*i* $\rightarrow$ *tempsum* EndFor EndFunc	*Done*
$g()$	5050

format()

Katalog >

format(*Udtr*[, *formatStreng*]**)**⇒*streng*

Returnerer *Udtr* som en tegnstreng 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.

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"

format()

Katalog >

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.

fPart()

Katalog >

fPart(*Udtr1***)⇒***udtryk*

fPart(*Liste1***)⇒***liste*

fPart(*Matrix1***)⇒***matrix*

Returnerer decimaldelen af argumentet.

For en liste eller matrix returneres decimaldelen af elementerne.

Argumentet kan være et reelt eller komplekst tal.

fPart(-1.234) → -0.234

fPart({1,-2.3,7.003}) → {0,-0.3,0.003}

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***)⇒***liste*

freqTable▸list({1,2,3,4},{1,4,3,1}) → {1,2,2,2,2,3,3,3,4}

freqTable▸list({1,2,3,4},{1,4,0,1}) → {1,2,2,2,2,4}

freqTable▸list() Katalog >

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- & Statistikapplikationerne.

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 259.

frequency() Katalog >

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), ..., b(n)}, er de specificerede områder {**?**≤b(1), b(1)<**?**≤b(2),...,b(n-1)<**?**≤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 **countIf()**-funktionen er resultatet { countIf(liste, **?**≤b(1)), countIf(liste, b(1)<**?**≤b(2)), ..., countIf(liste, b(n-1)<**?**≤b(n)), countIf(liste, b(n)>**?**)}.

datalist:={1,2,**e**,3,π,4,5,6,"hello",7}
{1,2,2.71828,3,3.14159,4,5,6,"hello",7}

frequency(*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.

frequency()

Katalog >

Elementer i *Liste1*, der ikke kan "placeres i en størrelse" ignoreres. Tomme (ugyldige) elementer ignoreres også. Yderligere oplysninger om tomme elementer findes på side 259.

I applikationen Lister og regneark kan du anvende et celleområde i stedet for begge argumenter.

Bemærk: Se også **countIf()**, side 37.

FTest_2Samp

Katalog >

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 182.)

eller H_a: σ1 > σ2, sæt *Hypot*>0

Til H_a: σ1 ≠ σ2 (standard), sæt *Hypot* =0

Til H_a: σ1 < σ2, sæt *Hypot*<0

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

Output-variabel	Beskrivelse
stat.F	Beregnet Û statistik for datasekvensen
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.dfTæller	frihedsgrader for tæller = n1-1

Output-variabel	Beskrivelse
stat.dfNævner	tæller, frihedsgrader = n2-1
stat.sx1, stat.sx2	Stikprøve standardafvigelse for datasekvenserne i *Liste 1* og *Liste 2*
stat.x1_bar stat.x2_bar	Middelværdi af stikprøver for datasekvenserne i *Liste 1* og *Liste 2*
stat.n1, stat.n2	Størrelse på stikprøverne

Func

Katalog >

Func
Blok
EndFunc

Skabelon til oprettelse af en brugerdefineret funktion.

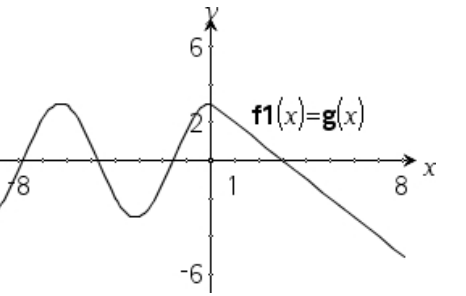
Blok kan være en enkelt sætning, en serie sætninger adskilt med kolon eller en serie sætninger på separate linjer. Funktionen kan anvende **Return**-instruktionen til at returnere et specifikt resultat.

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

Definition af en stykvis funktion:

```
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()

Katalog >

gcd(*Værdi1, Værdi2***)⇒***udtryk*

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(18,33)                          3
```

gcd() Katalog >

gcd(*Liste1, Liste2***)⇒***liste*

Returnerer de største fælles divisorer af de tilsvarende elementer i *Liste1* og *Liste2*.

$\gcd(\{12,14,16\},\{9,7,5\})$ $\{3,7,1\}$

gcd(*Matrix1, Matrix2***)⇒***matrix*

Returnerer de største fælles divisorer af de tilsvarende elementer i Matrix1 og *Matrix2*.

$\gcd\left(\begin{bmatrix}2&4\\6&8\end{bmatrix},\begin{bmatrix}4&8\\12&16\end{bmatrix}\right)$ $\begin{bmatrix}2&4\\6&8\end{bmatrix}$

geomCdf() Katalog >

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

geomCdf(*p***,***øvreGrænse***)** for P(1≤X ≤*øvreGrænse*)**⇒***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 ≤ *øvreGrænse*), sæt *nedreGrænse*= 1.

geomPdf() Katalog >

geomPdf(*p***,***XVærdi***)⇒***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*]

Eksempel: Anmod om den nuværende værdi for hubbens indbyggede lysniveausensor. Brug **Get** for at modtage værdien, og tildel den til den variable *lightval*.

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 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 88 og **Send**, side 163.

Send "READ BRIGHTNESS"	*Done*
Get *lightval*	*Done*
lightval	0.347922

Indlejr READ-anmodningen med **Get**-kommandoen.

Get "READ BRIGHTNESS",*lightval*	*Done*
lightval	0.378441

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

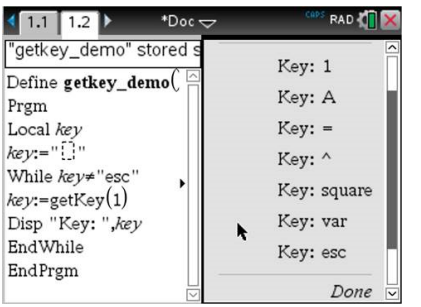
Beskrivelse:getKey() - tillader et TI-Basic-program at få tastaturinput - håndholdt, stationær og emuleret på stationær.

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.

getKey()

Eksempel:



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”

Håndholdte enheder/emulatortast	Skrivebord	Returværdi
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"
=	=	"="
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)

Håndholdte enheder/emulatortast	Skrivebord	Returværdi
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”
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 skrivebordskontroltaster	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
Hurtigsvar	Afslut program, håndter hændelse	Samme som den håndholdte (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC

Hændelse	Enhed	Skrivebord - TI-Nspire™ Student Software
		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*

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.

getLangInfo() "en"

getLangInfo() Katalog >

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() Catalog >

getLockInfo(*Var*)⇒*værdi*

Returnerer den aktuelle låste/oplåste tilstand på variablen *Var*.

værdi =**0**: *Var* er ulåst eller findes ikke.

værdi =**1**: *Var* er låst og kan ikke ændres eller slettes.

Se **Lock**, side 110, og**unLock**, side 204.

a:=65	65
Lock *a*	*Done*
getLockInfo(*a*)	1
a:=75	"Error: Variable is locked."
DelVar *a*	"Error: Variable is locked."
Unlock *a*	*Done*
a:=75	75
DelVar *a*	*Done*

getMode() Katalog >

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.

getMode(0)	{1,7,2,1,3,1,4,1,5,1,6,1,7,1,8,1}
getMode(1)	7
getMode(8)	1

getMode() Katalog >

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 167.

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(*Udtr1***)⇒***udtryk*

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[*promptStreng*,] *var*[, *statusVar*]

GetStr[*promptStreng*,] *func*(*arg1*, *...argn*) [, *statusVar*]

Se **Get** for eksempler.

GetStr

Hub-menu

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 81 og **Send**, side 163.

getType()

Katalog >

getType(*var***)⇒***streng*

Returnerer en streng, som angiver datatypen for variablen *var*.

Hvis *var* ikke er defineret, returneres strengen "NONE".

$\{1,2,3\} \rightarrow temp$	$\{1,2,3\}$
getType(*temp*)	"LIST"
$3 \cdot i \rightarrow temp$	$3 \cdot i$
getType(*temp*)	"EXPR"
DelVar *temp*	*Done*
getType(*temp*)	"NONE"

getVarInfo()

Katalog >

getVarInfo()⇒*matrix* eller *streng*

getVarInfo(*BibNavnStreng***)** ⇒*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 (*Biblioteksnavnestreng***)**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.

getVarInfo()	"NONE"
Define $x=5$	*Done*
Lock x	*Done*
Define LibPriv $y=\{1,2,3\}$	*Done*
Define LibPub $z(x)=3 \cdot x^2 - x$	*Done*
getVarInfo()	$\begin{bmatrix} x & \text{"NUM"} & \text{"}\square\text{"} & 1 \\ y & \text{"LIST"} & \text{"LibPriv "} & 0 \\ z & \text{"FUNC"} & \text{"LibPub "} & 0 \end{bmatrix}$
getVarInfo(*tmp3*)	"Error: Argument must be a string"
getVarInfo("tmp3")	$\begin{bmatrix} volcyl2 & \text{"NONE"} & \text{"LibPub "} & 0 \end{bmatrix}$

getVarInfo() Katalog >

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.

$a:=1$	1
$b:=[1\ \ 2]$	$[1\ \ 2]$
$c:=[1\ \ 3\ \ 7]$	$[1\ \ 3\ \ 7]$
$vs:=\text{getVarInfo}()$	$\begin{bmatrix} a & \text{"NUM"} & \text{"}\square\text{"} & 0 \\ b & \text{"MAT"} & \text{"}\square\text{"} & 0 \\ c & \text{"MAT"} & \text{"}\square\text{"} & 0 \end{bmatrix}$
$vs[1]$	$[1\ \ \text{"NUM"}\ \ \text{"}\square\text{"}\ \ 0]$
$vs[1,1]$	1
$vs[2]$	"Error: Invalid list or matrix"
$vs[2,1]$	$[1\ \ 2]$

Goto Katalog >

Goto *etiketnavn*

Overfører kontrol til etiketten *etiketnavn*.

Etiketnavn skal defineres i den samme funktion med en **Lbl**-kommando.

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→temp
    1→i
    Lbl top
    temp+i→temp
    If i<10 Then
    i+1→i
    Goto top
    EndIf
    Return temp
    EndFunc
g()                                55
```

▸Grad Katalog >

Udtr1 ▸ **Grad**⇒*udtryk*

Konverterer *Udtr1* til vinkelmål i nygrader.

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

I vinkeltilstanden Grader:

$(1.5)\blacktriangleright\text{Grad}$	$(1.66667)^g$

I vinkeltilstanden Radian:

$(1.5)\blacktriangleright\text{Grad}$	$(95.493)^g$

identitet() Katalog >

identitet(*heltal***)** ⇒ *matrix*

Returnerer identitetsmatrixen med en dimension af *heltal*.

Heltal skal være et positivt heltal.

identity(4)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

If Katalog >

If *BooleanExpr*
Statement

If *BooleanExpr* **Then**
Block
EndIf

Hvis *BooleanExpr* evalueres som sand, eksekveres enkeltsætningen *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.

Define $g(x)$=Func	*Done*
If $x<0$ Then	
Return x^2	
EndIf	
EndFunc	
$g(-2)$	4

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.

Define $g(x)$=Func	*Done*
If $x<0$ Then	
Return $-x$	
Else	
Return x	
EndIf	
EndFunc	
$g(12)$	12
$g(-12)$	12

If

Katalog >

Block1 og *Block2* kan være en enkelt sætning.

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)=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
g(-4)                                      4
g(10)                                      3
```

ifFn()

Katalog >

ifFn(*BooleanExpr*,*Value_If_true* [,*Value_If_false* [,*Value_If_unknown*]]**)** ⇒ *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()

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.

$\text{ifFn}(\{1,2,3\}<2.5,\{5,6,7\})$ $\{5,6,\text{undef}\}$

Value_If_False er ikke specificeret. Undef anvendes.

$\text{ifFn}(\{2,\text{"a"}\}<2.5,\{6,7\},\{9,10\},\text{"err"})$ $\{6,\text{"err"}\}$

Ét element valgt fra *Value_If_True*. Ét element valgt fra *Value_If_unknown*.

imag()

Katalog >

imag(*Expr1***)** ⇒ *udtryk*

Returnerer imaginærdelen af argumentet.

Bemærk: Alle udefinerede variable behandles som reelle variable. Se også reel (), page 150

$\text{imag}(1+2\cdot i)$ 2

$\text{imag}(z)$ 0

$\text{imag}(x+i\cdot y)$ y

imag(*List1***)** ⇒ *liste*

Returnerer en liste med imaginærdelen af elementerne.

$\text{imag}(\{-3,4-i,i\})$ $\{0,-1,1\}$

imag(*Matrix1***)** ⇒ *matrix*

Returnerer en matrix med imaginærdelene af elementerne.

$\text{imag}\begin{bmatrix} a & b \\ i\cdot c & i\cdot d \end{bmatrix}$ $\begin{bmatrix} 0 & 0 \\ c & d \end{bmatrix}$

impDif()

Katalog >

impDif(*Equation***,** *Var***,** *dependVar*[**,***Ord*]**)** ⇒ *udtryk*

hvor ordenen *Ord* som standard er 1.

Beregner den implicitte differentialkvotient til ligninger, hvor en variabel defineres implicit ud fra en anden.

$\text{impDif}(x^2+y^2=100,x,y)$ $\frac{-x}{y}$

Henvisning

Se #(), side 234.

inString()

Katalog >

inString(*srcString***,** *subString***[,** *Start***])** ⇒ *heltal*

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.

inString("Hello there","the")	7
inString("ABCEFG","D")	0

int()

Katalog >

int(*Expr***)** ⇒ *heltal*

int(*List1***)** ⇒ *liste*
int(*Matrix1***)** ⇒ *matrix*

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.

int(-2.5)	-3.
int([-1.234 0 0.37])	[-2. 0 0.]

intDiv()

Katalog >

intDiv(*Number1***,** *Number2***)** ⇒ *heltal*
intDiv(*List1***,** *List2***)** ⇒ *liste*
intDiv(*Matrix1***,** *Matrix2***)** ⇒ *matrix*

Returnerer den heltalsdel med fortegn, der er en del af (*Number1* ÷ *Number2*).

Returnerer for lister og matricer heltalsdelen med fortegn, der er en del af (argument 1 ÷ argument 2), for hvert elementpar.

intDiv(-7,2)	-3
intDiv(4,5)	0
intDiv({12,-14,-16},{5,4,-3})	{2,-3,5}

interpoler () Katalog >

interpoler(*xValue*, *xList*, *yList*, *yPrimeList***)** ⇒ *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:
$y'=-3\cdot y+6\cdot t+5$ og $y(0)=5$

$rk:=\text{rk23}(-3\cdot y+6\cdot t+5,t,y,\{0,10\},5,1)$

$\begin{bmatrix}0. & 1. & 2. & 3. & 4. \\ 5. & 3.19499 & 5.00394 & 6.99957 & 9.00593 & 10\end{bmatrix}$

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

Brug funktionen interpolate() til at beregne funktionsværdierne for listen med x-værdier:

$xvaluelist:=\text{seq}(i,i,0,10,0.5)$

$\{0,0.5,1.,1.5,2.,2.5,3.,3.5,4.,4.5,5.,5.5,6.,6.5,$

$xlist:=\text{mat}\blacktriangleright\text{list}(rk[1])$

$\{0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.\}$

$ylist:=\text{mat}\blacktriangleright\text{list}(rk[2])$

$\{5.,3.19499,5.00394,6.99957,9.00593,10.9978$

$yprimelist:=-3\cdot y+6\cdot t+5|y=ylist \text{ and } t=xlist$

$\{-10.,1.41503,1.98819,2.00129,1.98221,2.006$

$\text{interpolate}(xvaluelist,xlist,ylist,yprimelist)$

$\{5.,2.67062,3.19499,4.02782,5.00394,6.00011$

invχ²() Katalog >

invχ²(*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***)**

invF()

Katalog >

Beregner den inverse kumulerede F fordelingsfunktion angivet ved *dfNumer* og *dfDenom* for et givet *Areal* under kurven.

invBinom()

Katalog >

invBinom (*CumulativeProb***,***NumTrials***,***Prob***,** *OutputForm***)⇒** *skalar* eller *matrix*

Givet antallet af forsøg (*NumTrials*) og sandsynligheden for succes i hvert forsøg (*Prob*) vil denne funktion returnere det minimale antal successer, *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 %?

$$\text{invBinom}\left(0.77,30,\frac{1}{6}\right) \qquad 6$$

$$\text{invBinom}\left(0.77,30,\frac{1}{6},1\right) \qquad \begin{bmatrix} 5 & 0.616447 \\ 6 & 0.776537 \end{bmatrix}$$

invBinomN()

Katalog >

invBinomN(*CumulativeProb***,***Prob***,** *NumSuccess,OutputForm***)⇒** *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?

$$\text{invBinomN}(0.01,0.7,49) \qquad 86$$

$$\text{invBinomN}(0.01,0.7,49,1) \qquad \begin{bmatrix} 85 & 0.010451 \\ 86 & 0.00709 \end{bmatrix}$$

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***)** ⇒ *heltal*
iPart(*List1***)** ⇒ *liste*
iPart(*Matrix1***)** ⇒ *matrix*

Returnerer heltalsdelen af argumentet.

Returnerer heltalsdelen af hvert element for lister og matrixer.

Argumentet kan være et reelt eller komplekst tal.

$\text{iPart}(-1.234)$ → $-1.$

$\text{iPart}\left(\left\{\frac{3}{2},-2.3,7.003\right\}\right)$ → $\{1,-2.,7.\}$

irr() Katalog >

irr(*CF0*,*CFList* [,*CFFreq*]**)** ⇒ *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.

list1:={6000,−8000,2000,−3000} → {6000,−8000,2000,−3000}

list2:={2,2,2,1} → {2,2,2,1}

irr(5000,*list1*,*list2*) → −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 119.

isPrime() Katalog >

isPrime(*Number***)** ⇒ *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 *nextprim*(*n*)=Func		*Done*
	Loop	
	n+1→*n*	
	If isPrime(*n*)	
	Return *n*	
	EndLoop	
	EndFunc	
nextprim(7)		11

isVoid() Katalog >

isVoid(*Var***)** ⇒ *boolsk konstantudtryk*
isVoid(*Expr***)** ⇒ *boolsk konstantudtryk*
isVoid(*List***)** ⇒ *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 259.

a:=_	_
isVoid(*a*)	true
isVoid({1,_,3})	{false,true,false}

Lbl

Katalog >

Lbl *etiketNavn*

Definerer en etiket med navnet *etiketNavn* i en funktion.

Du kan anvende en **Goto** *etiketNavn*-kommando til at videregive kontrollen til kommandoen lige efter etiketten.

EtiketNavn 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.

```
Define g()=Func                    Done
    Local temp,i
    0→temp
    1→i
    Lbl top
    temp+i→temp
    If i<10 Then
    i+1→i
    Goto top
    EndIf
    Return temp
    EndFunc
g()                                55
```

lcm()

Katalog >

lcm(*Værdi1***,** *Værdi2***)⇒***udtryk*

lcm(*Liste1***,** *Liste2***)⇒***liste*

lcm(*Matrix1***,** *Matrix2***)⇒***matrix*

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.

```
lcm(6,9)                                   18
lcm({1/3,-14,16},{2/15,7,5})    {2/3,14,80}
```

left()

Katalog >

left(*kildeStreng***[,** *Tal***])⇒***streng*

Returnerer *Antal*-tegn fra venstre i tegnstrengen *kildeStreng*.

Hvis du udelader *Antal*, returneres alle *kildeStreng*-variable.

```
left("Hello",2)                            "He"
```

left() Katalog >

left(*Liste1*[**,** *Antal*]**)⇒***liste*

Returnerer *Antal*-elementer til venstre i *Liste1*.

Hvis du udelader *Antal*, returneres hele *Liste1*.

left(*Sammenligning***)⇒***udtryk*

Returnerer venstre af en ligning eller ulighed.

left({1,3,-2,4},3) {1,3,-2}

left(x<3) x

libShortcut() Katalog >

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 30.

For at slette en variabelgruppe, se **DelVar** på side 50.

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*])⇒*udtryk*

limit(*Liste1*, *Var*, *Punkt* [, *Retning*])⇒*liste*

limit(*Matrix1*, *Var*, *Punkt* [, *Retning*])⇒*matrix*

Returnerer den ønskede grænseværdi.

Bemærk: Se også **Grænseværdi skabelon**, side 7.

Retning: negativ=fra venstre, positiv=fra højre, ellers =begge. (Ved udeladelse er *Retning* som standard begge.)

Grænseværdier i + ∞ og - ∞ 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.

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.

$\lim_{x\to 5}(2\cdot x+3)$ → 13

$\lim_{x\to 0^+}\left(\frac{1}{x}\right)$ → ∞

$\lim_{x\to 0}\left(\frac{\sin(x)}{x}\right)$ → 1

$\lim_{h\to 0}\left(\frac{\sin(x+h)-\sin(x)}{h}\right)$ → $\cos(x)$

$\lim_{n\to\infty}\left(\left(1+\frac{1}{n}\right)^n\right)$ → *e*

$\lim_{x\to\infty}(a^x)$ → undef

$\lim_{x\to\infty}(a^x)\,|\,a>1$ → ∞

$\lim_{x\to\infty}(a^x)\,|\,a>0 \text{ and } a<1$ → 0

LinRegBx *X*, *Y*[, [*Frekv*][*Kategori*, *Medtag*]]

Beregner den lineære regressiony = a+b ·xpå listerne *X* og *Y* med hyppigheder *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 182.)

Alle lister skal have ens dimensioner med 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.

Kategory 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 259.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: a+b ·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 *X-liste* der faktisk bruges i regressionen ud fra begrænsninger i *Frekv*, *kategoriliste*,og *Medtag kategorier*
stat.YReg	Liste af datapunkter i den modificerede *Y-liste* der faktisk bruges i regressionen ud fra begrænsninger i *Frekv*, *kategoriliste*,og *Medtag kategorier*
stat.FreqReg	Liste med hyppigheder, der svarer til *stat.XReg* og *stat.YReg*

LinRegMx *X*, *Y*[, [*Frekv*][*Kategori*, *Medtag*]]

Beregner den lineære regression y = m · x + b på listerne *X* og *Y* med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 182.)

Alle lister skal have ens dimensioner med 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.

Kategory 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 259.

Output-variabel	**Beskrivelse**
stat.RegEqn	Regressionsligning: m · 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 *X-liste* der faktisk bruges i regressionen ud fra begrænsninger i *Frekv*, *kategoriliste*,og *Medtag kategorier*
stat.YReg	Liste af datapunkter i den modificerede *Y-liste* der faktisk bruges i regressionen ud fra begrænsninger i *Frekv*, *kategoriliste*,og *Medtag kategorier*
stat.FreqReg	Liste med hyppigheder, der svarer til *stat.XReg* og *stat.YReg*

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 182)

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 259.

Output-variabel	**Beskrivelse**
stat.RegEqn	Regressionsligning: a+b·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-variabel	**Beskrivelse**
[stat.CLower, stat.CUpper]	Konfidensinterval for hældningen
stat.ME	Konfidensinterval, fejlmargen

Output-variabel	Beskrivelse
stat.SESlope	Standarfejl 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	Prædiktionsintervalsmargin for fejl
stat.SEPred	standardfejl for prædiktion
Statistik.$\hat{y}$	a + b · XVærdi

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=α+βx. Den tester nulhypotesen H_0:β=0 (ækvivalent, ρ=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 nul-hypotesen (H_0:β=ρ=0) vil blive testet.

Til H_a: β≠0 og ρ≠0 (standard), sæt $Hypot$=0

LinRegtTest

katalog >

Til H_a: $\beta<0$ og $\rho<0$, sæt *Hypot*<=0

Til H_a: $\beta>0$ og $\rho>0$, sæt *Hypot*> 0

En sammenfatning af resultaterne lagres i *stat.results* variable. (side 182)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: a + b · x
stat.t	*t*-Statistik for signifikanstest
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	Standarfejl for hældning
stat.r^2	Forklaringsgraden
stat.r	Korrelationskoefficient
stat.Resid	Residualer fra regressionen

linSolve()

Katalog >

linSolve(*SystemafLineæreLigninger*, *Var1*, *Var2*, ...**)⇒***liste*

linSolve(*LineærLign1* **og** *LineærLign2* **og** ..., *Var1*, *Var2*, ...**)** ⇒ *liste*

linSolve({*LineærLign1*, *LineærLign2*, ...**},** *Var1*, *Var2*, ...**)** ⇒ *liste*

linSolve(*SystemafLineæreLigninger*, **{***Var1*, *Var2*, ...**})** ⇒ *liste*

linSolve(*LineærLign1* **og** *LineærLign2* **og** ..., **{***Var1*, *Var2*, ...**})** ⇒ *liste*

linSolve({*LineærLign1*, *LineærLign12*, ...**},** **{***Var1*, *Var2*, ...**})** ⇒

$\text{linSolve}\left(\left\{2\cdot x+4\cdot y=3,\ 5\cdot x-3\cdot y=7\right\},\{x,y\}\right)\quad \left\{\frac{37}{26},\frac{1}{26}\right\}$

$\text{linSolve}\left(\left\{2\cdot x=3,\ 5\cdot x-3\cdot y=7\right\},\{x,y\}\right)\quad \left\{\frac{3}{2},\frac{1}{6}\right\}$

$\text{linSolve}\left(\left\{apple+4\cdot pear=23,\ 5\cdot apple-pear=17\right\},\{apple,pear\}\right)\quad \left\{\frac{13}{3},\frac{14}{3}\right\}$

$\text{linSolve}\left(\left\{apple\cdot 4+\frac{pear}{3}=14,\ -apple+pear=6\right\},\{apple,pear\}\right)\quad \left\{\frac{36}{13},\frac{114}{13}\right\}$

linSolve()

Katalog >

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()

Katalog >

Δlist(*Liste1***)⇒***liste*

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*.

ΔList({20,30,45,70}) {10,15,25}

list▸mat()

Katalog >

list▸mat(*Liste* **[,** *elementerPrRække***])⇒***matrix*

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(...)`.

list▸mat({1,2,3}) [1 2 3]

list▸mat({1,2,3,4,5},2)

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 0 \end{bmatrix}$$

▸ln

Katalog >

Udtr ▸**ln**⇒*udtryk*

Konverterer inputtet *Udtr* til et udtryk, der kun indeholder naturlige logaritmer (ln).

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

$\left(\log_{10}(x)\right)\blacktriangleright \ln \quad \frac{\ln(x)}{\ln(10)}$

ln()

ctrl ex -taster

ln(*Udtr1***)**⇒*udtryk*

ln(*Liste1***)**⇒*liste*

Returnerer den naturlige logaritme til argumentet.

Til en liste returneres de naturlige logaritmer af elementerne.

$\ln(2.) \quad 0.693147$

Hvis kompleks formattilstand er reel:

$\ln(\{-3,1.2,5\})$

"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()**.

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

I vinkeltilstanden radian og rektangulært komplekst format:

$$\ln\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix}$$

$$\begin{bmatrix}1.83145+1.73485\cdot i & 0.009193-1.49086\\ 0.448761-0.725533\cdot i & 1.06491+0.623491\\ -0.266891-2.08316\cdot i & 1.12436+1.79018\cdot\end{bmatrix}$$

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*]]

LnReg **Katalog >**

Beregner den lineære regression y = a+b · ln (x)på liste X og Y med hyppighed $Frekv$. En sammenfatning af resultaterne lagres i $stat.results$ variable. (side 182.)

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.

$Kategory$ 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 259.

Output-variabel	**Beskrivelse**
stat.RegEqn	Regressionsligning: a+b · 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 X-$liste$ der faktisk bruges i regressionen ud fra begrænsninger i $Frekv$, $kategoriliste$,og $Medtag\ kategorier$
stat.YReg	Liste af datapunkter i den modificerede Y-$liste$ der faktisk bruges i regressionen ud fra begrænsninger i $Frekv$, $kategoriliste$,og $Medtag\ kategorier$
stat.FreqReg	Liste med hyppigheder, der svarer til $stat.XReg$ og $stat.YReg$

Local

Katalog >

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
rollcount()                                   16
rollcount()                                   3
```

Lock

Catalog >

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 systemvariablen *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 204 og **getLockInfo()**, side 87.

```
a:=65                                         65
Lock a                                        Done
getLockInfo(a)                                1
a:=75                    "Error: Variable is locked."
DelVar a                 "Error: Variable is locked."
Unlock a                                      Done
a:=75                                         75
DelVar a                                      Done
```

log()

ctrl 10^x -taster

log(*Udtr1*[**,***Udtr2*]**)⇒***udtryk*

log(*Liste1*[**,***Udtr2*]**)⇒***liste*

Returner -*Udtr2*-talslogaritmen til argumentet.

Bemærk: Se også **Log-skabelon**, side 2.

Ved en liste returneres *Udtr2*-talslogaritmen til elementerne.

Hvis *Udtr2* udelades, anvendes 10-talslogaritmen.

$\log_{10}(2.)$ → 0.30103

$\log_{4}(2.)$ → 0.5

$\log_{3}(10)-\log_{3}(5)$ → $\log_{3}(2)$

Hvis kompleks formattilstand er reel:

$\log_{10}(\{-3,1.2,5\})$ → Error: *Non*–real *result*

Hvis kompleks formattilstand er rektangulær:

$\log_{10}(\{-3,1.2,5\})$

$\{\log_{10}(3)+1.36438\cdot i, 0.079181, \log_{10}(5)\}$

log(*kvadratMatrix1*
[**,***Udtr2*]**)⇒***kvadratMatrix*

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.

Hvis tal-argumentet udelades, anvendes 10-talslogaritmen.

I vinkeltilstanden radian og rektangulært komplekst format:

$\log_{10}\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix}$

$\begin{bmatrix}0.795387+0.753438\cdot i & 0.003993-0.6474\\ 0.194895-0.315095\cdot i & 0.462485+0.2707\\ -0.115909-0.904706\cdot i & 0.488304+0.7774\end{bmatrix}$

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

▸logbase

Katalog >

Udtr1 **▸logbase(***Udtr2***)⇒***udtryk*

Omformer inputudtrykket til et udtryk, der anvender *Udtr2*-talslogaritmen.

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive `@>logbase(...)`.

$\log_{3}(10)-\log_{5}(5)\blacktriangleright\text{logbase}(5)$ → $\dfrac{\log_{5}\left(\frac{10}{3}\right)}{\log_{5}(3)}$

Logistik *X*, *Y*[, [*Frekv*] [, *Kategori*, *Medtag*]]

Beregner den logistiske regressiony = (c/(1+a ·e^{-bx}))på listerne *X* og *Y* med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 182.)

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.

Kategory 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 259.

Output-variabel	**Beskrivelse**
stat.RegEqn	Regressionsligning: c/(1+a · e^{-bx})
stat.a, stat.b, stat.c	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 *Frekv, kategoriliste,*og *Medtag kategorier*
stat.YReg	Liste af datapunkter i den modificerede *Y-liste* der faktisk bruges i regressionen ud fra begrænsninger i *Frekv, kategoriliste,*og *Medtag kategorier*
stat.FreqReg	Liste med hyppigheder, der svarer til *stat.XReg* og *stat.YReg*

LogisticD *X*, *Y* [, [*Iterationer*], [*Frekv*] [, *Kategori*, *Medtag*]]

Beregner den logistiske regression y = (c/(1+a ·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 182.)

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.

Kategory 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 259.

Output-variabel	**Beskrivelse**
stat.RegEqn	Regressionsligning: c/(1+a ·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 *X-liste* der faktisk bruges i regressionen ud fra begrænsninger i *Frekv*, *Kategori liste* og *Medtag Kategorier*
stat.YReg	Liste af datapunkter i den modificerede *Y-liste* der faktisk bruges i regressionen ud fra begrænsninger i *Frekv*, *Kategori liste* og *Medtag Kategorier*
stat.FreqReg	Liste med hyppigheder, der svarer til *stat.XReg* og *stat.YReg*

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
rollcount()                                         16
rollcount()                                         3
```

LU

Katalog >

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 · *uMatrix* = *pMatrix* · *matrix*

$$\begin{bmatrix}6 & 12 & 18\\5 & 14 & 31\\3 & 8 & 18\end{bmatrix}\rightarrow m1 \qquad \begin{bmatrix}6 & 12 & 18\\5 & 14 & 31\\3 & 8 & 18\end{bmatrix}$$

LU *m1*,*lower*,*upper*,*perm* — *Done*

lower
$$\begin{bmatrix}1 & 0 & 0\\\frac{5}{6} & 1 & 0\\\frac{1}{2} & \frac{1}{2} & 1\end{bmatrix}$$

upper
$$\begin{bmatrix}6 & 12 & 18\\0 & 4 & 16\\0 & 0 & 1\end{bmatrix}$$

perm
$$\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$$

LU

Katalog >

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 ctrl enter 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 ·max(dim(*Matrix*)) ·rækkeNorm (*Matrix*)

LU-algoritmen til faktoropløsning anvender partiel pivotering med rækkeombytninger.

$$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1 \qquad \begin{bmatrix} m & n \\ o & p \end{bmatrix}$$

$$\text{LU } m1, lower, upper, perm \qquad Done$$

$$lower \qquad \begin{bmatrix} 1 & 0 \\ \frac{m}{o} & 1 \end{bmatrix}$$

$$upper \qquad \begin{bmatrix} o & p \\ 0 & n-\frac{m \cdot p}{o} \end{bmatrix}$$

$$perm \qquad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

M

mat▸list()

Katalog >

mat▸list(*Matrix***)⇒***liste*

Returner en liste bestående af elementerne i *Matrix*. Elementerne kopieres fra *Matrix* række for række.

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

$$\text{mat}\blacktriangleright\text{list}\left(\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}\right) \qquad \{1,2,3\}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow m1 \qquad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

$$\text{mat}\blacktriangleright\text{list}(m1) \qquad \{1,2,3,4,5,6\}$$

max()

Katalog >

max(*Udtr1***, *Udtr2***)⇒***udtryk*

max(*Liste1***, *Liste2***)⇒***liste*

max(*Matrix1***, *Matrix2***)⇒***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.

$$\max(2.3, 1.4) \qquad 2.3$$

$$\max(\{1,2\}, \{-4,3\}) \qquad \{1,3\}$$

max() Katalog >

max(*List***)⇒***udtryk*

Returnerer det største element i *liste*.

max(*Matrix1***)⇒***matrix*

Returnerer en rækkevektor med det største element i hver kolonne i *Matrix1*.

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 259

Bemærk: Se også **fMax()** og **min()**.

$\max(\{0,1,-7,1.3,0.5\})$ → 1.3

$\max\begin{bmatrix} 1 & -3 & 7 \\ -4 & 0 & 0.3 \end{bmatrix}$ → $\begin{bmatrix} 1 & 0 & 7 \end{bmatrix}$

mean() Katalog >

mean(*Liste***[, *hyppighedsliste*])⇒***udtryk*

Returnerer middelværdien for elementerne i *Liste*.

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

mean(*Matrix1***[, *Hyppighedsmatrix*])**
⇒*matrix*

Returnerer en rækkevektor af middelværdierne af alle kolonner i *Matrix1*.

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 259.

$\text{mean}(\{0.2,0,1,-0.3,0.4\})$ → 0.26

$\text{mean}(\{1,2,3\},\{3,2,1\})$ → $\frac{5}{3}$

I rektangulært vektorformat:

$\text{mean}\begin{bmatrix} 0.2 & 0 \\ -1 & 3 \\ 0.4 & -0.5 \end{bmatrix}$ → $\begin{bmatrix} -0.133333 & 0.833333 \end{bmatrix}$

$\text{mean}\begin{bmatrix} \frac{1}{5} & 0 \\ -1 & 3 \\ \frac{2}{5} & \frac{-1}{2} \end{bmatrix}$ → $\begin{bmatrix} \frac{-2}{15} & \frac{5}{6} \end{bmatrix}$

$\text{mean}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix},\begin{bmatrix} 5 & 3 \\ 4 & 1 \\ 6 & 2 \end{bmatrix}\right)$ → $\begin{bmatrix} \frac{47}{15} & \frac{11}{3} \end{bmatrix}$

median() Katalog >

median(*Liste***[, *Hyppighedsliste*])⇒***udtryk*

Returnerer medianen af elementerne i *Liste*.

$\text{median}(\{0.2,0,1,-0.3,0.4\})$ → 0.2

median() Katalog >

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*.

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 259.

$$\text{median}\left(\begin{bmatrix} 0.2 & 0 \\ 1 & -0.3 \\ 0.4 & -0.5 \end{bmatrix}\right) \qquad \begin{bmatrix} 0.4 & -0.3 \end{bmatrix}$$

MedMed Katalog >

MedMed *X*, *Y* [, *Frekv*] [, *Kategori*, *Medtag*]]

Beregner median-median linjey = (m ·x+b)på listerne *X* og *Y* med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 182.)

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.

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

MedMed

Katalog >

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 259.

Output-variabel	Beskrivelse
stat.RegEqn	Median-median-linjeligning: m ·x+b
stat.m, stat.b	Modelkoeifficienter
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 *Frekv*, *kategoriliste*,og *Medtag Kategorier*
stat.YReg	Liste af datapunkter i den modificerede *Y-liste* der faktisk bruges i regressionen ud fra begrænsninger i *Frekv*, *kategoriliste*,og *Medtag Kategorier*
stat.FreqReg	Liste med hyppigheder, der svarer til *stat.XReg* og *stat.YReg*

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("Hello there",2)      "ello there"
mid("Hello there",7,3)    "the"
mid("Hello there",1,5)    "Hello"
mid("Hello there",1,0)    ""
```

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*.

```
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() Katalog >

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({"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(2.3,1.4)                    1.4
min({1,2},{-4,3})            {-4,2}
```

min(*Liste***)⇒***udtryk*

Returnerer det mindste element af *Liste*.

```
min({0,1,-7,1.3,0.5})           -7
```

min(*Matrix1***)⇒***matrix*

Returnerer en rækkevektor med det mindste element i hver kolonne i *Matrix1*.

```
min([ 1  -3  7 ])          [-4  -3  0.3]
   [-4   0  0.3]
```

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

mirr() Katalog >

mirr
(
*finansRente***,***geninvestRente***,***CF0***,***CFListe*
[,*CFFrekv***])**

Finansfunktion, der returnerer den modificerede interne rente af en investering.

finansRente er rentesatsen, du betaler for pengestrømsbeløbene.

geninvestRente er rentesatsen, som pengestrømmen geninvesteres til.

```
list1:={6000,-8000,2000,-3000}
                {6000,-8000,2000,-3000}
list2:={2,2,2,1}                {2,2,2,1}
mirr(4.65,12,5000,list1,list2)  13.41608607
```

mirr()

Katalog >

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 97.

mod()

Katalog >

mod(*Udtr1***,** *Udtr2***)⇒***udtryk*

mod(*Liste1***,** *Liste2***)⇒***liste*

mod(*Matrix1***,** *Matrix2***)⇒***matrix*

Returnerer det første argument modulo andet argument som defineret efter definitionen:

mod(x,0) = x

mod(x,y) = x − y 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 154

$\text{mod}(7,0)$	7
$\text{mod}(7,3)$	1
$\text{mod}(-7,3)$	2
$\text{mod}(7,-3)$	-2
$\text{mod}(-7,-3)$	-1
$\text{mod}(\{12,-14,16\},\{9,7,-5\})$	$\{3,0,-4\}$

mRow()

Katalog >

mRow(*Udtr***,** *Matrix1***,** *Indeks***)⇒***matrix*

$$\text{mRow}\left(\frac{-1}{3},\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix},2\right) \qquad \begin{bmatrix}1 & 2\\ -1 & \frac{-4}{3}\end{bmatrix}$$

mRow() Katalog >

Returnerer en kopi af *Matrix1* med hvert element i rækken *Indeks* af *Matrix1* ganget med *Udtr*.

mRowAdd() Katalog >

mRowAdd(*Udtr*, *Matrix1*, *Indeks1*, *Indeks2***)** ⇒*matrix*

Returnerer en kopi af *Matrix1* med hvert element i rækken *Indeks2* af *Matrix1* erstattet med:

Udtr × *række Indeks1* + *række Indeks2*

Indeks2

$$\text{mRowAdd}\left(-3,\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix},1,2\right) \qquad \begin{bmatrix}1 & 2\\ 0 & -2\end{bmatrix}$$

$$\text{mRowAdd}\left(n,\begin{bmatrix}a & b\\ c & d\end{bmatrix},1,2\right) \qquad \begin{bmatrix}a & b\\ a\cdot n+c & b\cdot n+d\end{bmatrix}$$

MultReg Katalog >

MultReg *Y*, *X1*[,*X2*[,*X3*,...[,*X10*]]]

Beregner multiple lineære regressioner af listen *Y* på listerne *X1*, *X2*, ..., *X10*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 182.)

Alle lister skal have samme dimension.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: b0+b1 · x1+b2 · x2+ ...
stat.b0, stat.b1, ...	Regressionskoefficienter
stat.r^2	Koefficient af multipel forklaringsgrad
stati.$\hat{y}$List	$\hat{y}$ Liste = b0+b1 · x1+ ...
stat.Resid	Residualer fra regressionen

MultRegIntervals Katalog >

MultRegIntervals *Y*, *X1*[,*X2*[,*X3*,...[,*X10*]]],*XValListe*[,*CNiveau*]

MultRegIntervals

Katalog >

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 182.)

Alle lister skal have samme dimension.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

Output-variabel	**Beskrivelse**
stat.RegEqn	Regressionsligning: b0+b1 · x1+b2 · x2+ ...
Statistik.$\hat{y}$	Et punktestimat: $\hat{y}$ = b0 + b1 · xl + ... for *XValListe*
stat.dfFejl (stat.dfError)	Frihedsgrader for fejl
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 182.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

`Output`

Output-variabel	**Beskrivelse**
stat.RegEqn	Regressionsligning: b0+b1 · x1+b2 · x2+ …
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
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.$\hat{y}$ List	$\hat{y}$ Liste = b0+b1 · x1+ . . .
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

BoolskUdtryk1 **nand** *BoolskUdtryk2*
returnerer *boolsk udtryk*

BoolskListe1 **nand** *BoolskListe2* returnerer *Boolsk liste*

BoolskMatrix1 **nand** *BoolskMatrix2*
returnerer *Boolsk matrix*

$x\geq 3$ and $x\geq 4$	$x\geq 4$
$x\geq 3$ nand $x\geq 4$	$x<4$

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* ⇒ *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 hexadecimale indtastninger skal du som præfiks benytte henholdsvis 0b eller 0h. Uden præfiks behandles heltallene som decimaltal (10 talssystem)

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}

nCr()

Katalog >

nCr(*Udtr1*, *Udtr2***)⇒***udtryk*

For heltal *Udtr1* og *Udtr2* med *Udtr2* ≥ *Udtr2* ≥ 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.

$\mathrm{nCr}(z,3)$	$\frac{z\cdot(z-2)\cdot(z-1)}{6}$
$Ans \mid z=5$	10
$\mathrm{nCr}(z,c)$	$\frac{z!}{c!\cdot(z-c)!}$
$\frac{Ans}{\mathrm{nPr}(z,c)}$	$\frac{1}{c!}$

nCr(*Udtr*, **0)⇒1**

nCr(*Udtr*, *negativtHeltal***)⇒0**

nCr(*Udtr*, *positivtHeltal***)⇒** *Udtr* · **(***Udtr*−1**)... (***Udtr*−*positivtHeltal***+1)/** *positivtHeltal***!**

nCr(*Udtr*, *ikkeHeltal***)⇒***udtryk***!/ ((***Udtr*−*ikkeHeltal***)!** ·*ikkeHeltal***!)**

nCr(*Liste1*, *Liste2***)⇒***liste*

Returnerer en liste med kombinationer baseret på de sammenhørende elementpar i de to lister. Argumenterne skal være lister af samme størrelse.

$\mathrm{nCr}(\{5,4,3\},\{2,4,2\})$	$\{10,1,3\}$

nCr(*Matrix1*, *Matrix2***)⇒***matrix*

Returnerer en matrix af kombinationer baseret på de sammenhørende elementpar i de to matricer. Argumenterne skal være matricer af samme størrelse.

$\mathrm{nCr}\left(\begin{bmatrix}6 & 5\\ 4 & 3\end{bmatrix},\begin{bmatrix}2 & 2\\ 2 & 2\end{bmatrix}\right)$	$\begin{bmatrix}15 & 10\\ 6 & 3\end{bmatrix}$

nDerivative()

Katalog >

nDerivative(*Udtr1*,*Var*=*Værdi* [,*Orden*]**)⇒***værdi*

nDerivative(*Udtr1*,*Var*[,*Orden*]**) |** *Var*=*Værdi***⇒***værdi*

Returnerer den numeriske differentialkvotient udregnet med en automatisk differentiationsmetode.

$\mathrm{nDerivative}(\lvert x\rvert,x=1)$	1
$\mathrm{nDerivative}(\lvert x\rvert,x) \mid x=0$	undef
$\mathrm{nDerivative}(\sqrt{x-1},x) \mid x=1$	undef

nDerivative()

Katalog >

Når *Værdi* er angivet, tilsidesætter den alle forudgående variabeltildelinger og alle nuværende "|" substitutioner for variablen.

Differentialkvotientens orden skal være **1** eller **2**.

newList()

Katalog >

newList(*antalElementer***)⇒***liste*

Returnerer en liste med en dimension af *antalElementer*. Hvert element er nul.

$\text{newList}(4)$ $\{0,0,0,0\}$

newMat()

Katalog >

newMat(*antalRækker***, *antalKolonner***)⇒***matrix*

Returnerer en matrix med nulpunkter af dimensionen *antalRækker* gange *antalKolonner*.

$\text{newMat}(2,3)$ $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

nfMax()

Katalog >

nfMax(*Udtr***, *Var***)⇒***værdi*

nfMax(*Udtr***, *Var*, *nedreGrænse***)⇒***værdi*

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()**.

$\text{nfMax}(-x^2-2\cdot x-1,x)$ $-1.$

$\text{nfMax}(0.5\cdot x^3-x-2,x,-5,5)$ $5.$

nfMin() Katalog >

nfMin(*Udtr*, *Var*)⇒*værdi*

nfMin(*Udtr*, *Var*, *nedreGrænse*)⇒*værdi*

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()**.

$\text{nfMin}(x^2+2\cdot x+5,x)$ $-1.$

$\text{nfMin}(0.5\cdot x^3-x-2,x,-5,5)$ $-5.$

nInt() Katalog >

nInt(*Udtr1*, *Var*, *nedre*, *øvre*)⇒*udtryk*

Hvis integranden *Udtr1* ikke indeholder andre variable end *Var*, og hvis *Nedre* og *Øvre* er konstante, + ∞ eller - ∞, så returnerer **nInt()** en tilnærmet værdi af ∫**(***Udtr1*, *Var*, *Nedre*, *Øvre***)**. Denne tilnærmede værdi er et vægtet gennemsnit af nogle eksempelværdier af integranden i intervallet *Nedre*<*Var*<*Øvre*.

$\text{nInt}(e^{-x^2},x,-1,1)$ 1.49365

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.

$\text{nInt}(\cos(x),x,-\pi,\pi+1.\text{E}{-12})$ $-1.04144\text{E}{-12}$

$\int_{-\pi}^{\pi+10^{-12}} \cos(x)\,dx$ $-\sin\left(\frac{1}{1000000000000}\right)$

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.

$\text{nInt}\left(\text{nInt}\left(\frac{e^{-x\cdot y}}{\sqrt{x^2-y^2}},y,-x,x\right),x,0,1\right)$ 3.30423

Bemærk: Se også ∫**()**, side 217.

nom()

Katalog >

nom(*effektivRente*,*CpY*)⇒*værdi*

Finansfunktion, der omregner den effektive årlige rente *effektivRente* til en nominel 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 61.

nom(5.90398,12)	5.75

nor

ctrl **=** **-taster**

BoolskUdtryk1 **nor** *BoolskUdtryk2* returnerer *boolsk udtryk*

BoolskListe1 **nor** *BoolskListe2* 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.

$x\geq 3$ or $x\geq 4$	$x\geq 3$
$x\geq 3$ nor $x\geq 4$	$x<3$

*Heltal1***nor***Heltal2*⇒*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***)⇒***udtryk*

norm(*Vektor***)⇒***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."

$\text{normalLine}\left(x^2,x,1\right)$	$\frac{3}{2}-\frac{x}{2}$
$\text{normalLine}\left((x-3)^2-4,x,3\right)$	$x=3$
$\text{normalLine}\left(\begin{matrix}\frac{1}{3},x=0\end{matrix}\right)$	0
$\text{normalLine}\left(\sqrt{\lvert x\rvert},x=0\right)$	undef

normCdf() Katalog >

normCdf(*nedreGrænse***,***øvreGrænse*[,μ[,σ]]**)⇒***tal* hvis *nedreGrænse* og *øvreGrænse* er tal, *liste* hvis *nedreGrænse* og *øvreGrænse* er lister

Beregner normalfordelingenssandsynligheden mellem *nedreGrænse* og *øvreGrænse* for de angivne μ (standard=0) og σ (standard=1).

For P(X ≤ *øvreGrænse*), sæt *nedreGrænse*= −∞.

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.

not

Katalog >

not *Boolsk udtr1*⇒*Boolsk udtryk*

Returnerer true eller false eller en forenklet form af argumentet.

not *Heltal1*⇒*heltal*

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 hexadecimale 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(2≥3)	true
not(*x*<2)	*x*≥2
not not *innocent*	*innocent*

I hexadecimal tilstand:

Vigtigt: Tallet nul, ikke bogstavet O.

not 0h7AC36	0hFFFFFFFFFF853C9

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()

Katalog >

nPr(*Udtr1***,** *Udtr2***)**⇒*udtryk*

For heltal er *Udtr1* og *Udtr2* med *Udtr1* ≥ *Udtr2* ≥ 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)** · **(***Udtr***+2)**... **(***Udtr*−*negHeltal***))**

nPr(*Udtr***,** *posHeltal***)** ⇒ *Udtr* · **(***Udtr***−1)...** **(***Udtr*−*posHeltal***+1)**

nPr(*Udtr***,** *ikkeHetal***)** ⇒ *Udtr***! /** **(***Udtr*−*ikkeHeltal***)!**

nPr(*z*,3)	$z\cdot(z-2)\cdot(z-1)$
Ans|*z*=5	60
nPr(*z*,−3)	$\frac{1}{(z+1)\cdot(z+2)\cdot(z+3)}$
nPr(*z*,*c*)	$\frac{z!}{(z-c)!}$
Ans·nPr(*z*−*c*,−*c*)	1

nPr() Katalog >

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.

$\text{nPr}(\{5,4,3\},\{2,4,2\})$ → $\{20,24,6\}$

nPr(*Matrix1*, *Matrix2*)⇒*matrix*

Returnerer en matrix med permutationer baseret på de sammenhørende elementpar i de to matricer. Argumenterne skal være matricer af samme størrelse.

$\text{nPr}\left(\begin{bmatrix}6 & 5\\ 4 & 3\end{bmatrix},\begin{bmatrix}2 & 2\\ 2 & 2\end{bmatrix}\right)$ → $\begin{bmatrix}30 & 20\\ 12 & 6\end{bmatrix}$

npv() Katalog >

npv(*RenteSats*,*CFO*,*CFListe*[,*CFFrekv*]**)**

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).

CF0 er startpengestrømmen på tidspunkt 0. Den skal være et reelt tal.

CFListe er en liste over pengestrømsbeløb efter startpengestrømmen *CF0*.

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.

list1:={6000,⁻8000,2000,⁻3000} → {6000,⁻8000,2000,⁻3000}

list2:={2,2,2,1} → {2,2,2,1}

npv(10,5000,*list1*,*list2*) → 4769.91

nSolve() Katalog >

nSolve(*Ligning*,*Var*[=*Gæt*]**)**⇒*tal eller fejlstreng*

nSolve(*Ligning*,*Var*[=*Gæt*]**,***nedreGrænse***)** ⇒*tal eller fejlstreng*

$\text{nSolve}(x^2+5\cdot x-25=9,x)$ → 3.84429

$\text{nSolve}(x^2=4,x=-1)$ → -2.

$\text{nSolve}(x^2=4,x=1)$ → 2.

nSolve(*Ligning*,*Var* **[=***Guess***]**,*nedreGrænse*,*øvreGrænse***)** ⇒*tal eller fejlstreng*

nSolve(*Ligning*,*Var***[=***Guess***]) |** *nedreGrænse*≤*Var*≤*øvreGrænse* ⇒ *tal eller fejlstreng*

Søger iterativt efter en approksimeret reel numerisk løsning af *Ligning* for dens ene variabel. Angiv variablen 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.

$\text{nSolve}\left(x^2+5\cdot x-25=9,x\right)\mid x<0$	-8.84429
$\text{nSolve}\left(\frac{(1+r)^{24}-1}{r}=26,r\right)\mid r>0 \text{ and } r<0.25$	0.006886
$\text{nSolve}\left(x^2=-1,x\right)$	"No solution found"

O

OneVar [**1,**]*X*[**,**[*Hyppighed*] [**,***Kategori*,*Medtag*]]

OneVar [*n***,**]*X1*,*X2*[*X3*[**,**...[**,***X20*]]]

Beregner statistik med en variabel på op til 20 lister. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 182.)

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 259

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. σx	Populations standardafvigelse for x
stat.n	Antal datapunkter
stat.MinX	Minimum af x-værdier
stat.Q_1X	1. kvartil af x
stat.MedianX	Median af x.
stat.Q_3X	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*

BoolskListe1 **or** *BoolskListe2* returnerer *Boolsk liste*

BoolskMatrix1 **or** *BoolskMatrix2* returnerer *Boolsk matrix*

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

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*⇒*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.

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).

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.

$x\geq 3$ or $x\geq 4$	$x\geq 3$

Define $g(x)$=	Func	*Done*
	If $x\leq 0$ or $x\geq 5$	
	Goto *end*	
	Return $x\cdot 3$	
	Lbl *end*	
	EndFunc	
$g(3)$		9
$g(0)$		*A function did* not return *a value*

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.

or

Katalog >

Bemærk: Se **xor.**

ord()

Katalog >

ord(*Streng***)⇒***heltal*

ord(*Liste1***)⇒***liste*

Returnerer den numeriske kode til første tegn i tegnstrengen *Streng*, eller en liste med de første tegn i hvert listeelement.

ord("hello")	104
char(104)	"h"
ord(char(24))	24
ord({"alpha","beta"})	{97,98}

P

P▸Rx()

Katalog >

P▸Rx(*rUdtr***, θ***Udtr***)⇒***udtryk*

P▸Rx(*rListe***, θ***Liste***)⇒***liste*

P▸Rx(*rMatrix***, θ***Matrix***)⇒***matrix*

Returnerer den ækvivalente x-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 °, G eller r til midlertidigt at tilsidesætte den indstillede vinkeltilstand.

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

I vinkeltilstanden Radian:

P▸Rx(r,θ)	cos(θ)·r
P▸Rx(4,60°)	2
P▸Rx({-3,10,1.3},{π/3,-π/4,0})	{-3/2,5·√2,1.3}

P▸Ry()

Katalog >

P▸Ry(*rUdtr***, θ***Udtr***)⇒***udtryk*

P▸Ry(*rListe***, θ***Liste***)⇒***liste*

P▸Ry(*rMatrix***, θ***Matrix***)⇒***matrix*

Returnerer den ækvivalente y-koordinat til parret (r, θ).

I vinkeltilstanden Radian:

P▸Ry(r,θ)	sin(θ)·r
P▸Ry(4,60°)	2·√3
P▸Ry({-3,10,1.3},{π/3,-π/4,0})	{-3·√3/2,-5·√2,0.}

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@>Ry(...)**.

PassErr

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 26, og **Try**, side 197.

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

Se et eksempel på **PassErr** i Eksempel2 under **Try**-kommandoen, side 197.

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 $p(x)=\begin{cases} x, & x>0 \\ \text{undef}, & x\le 0 \end{cases}$	*Done*
$p(1)$	1
$p(-1)$	undef

poissCdf() Katalog >

poissCdf(λ**,***nedreGrænse*,*øvreGrænse***)**⇒*tal* hvis *nedreGrænse* og *øvreGrænse* er tal, *liste* hvis *nedreGrænse* og *øvregrænse* er lister

poissCdf(λ**,***øvreGrænse***)**(for P(0≤X ≤*øvreGrænse*)⇒*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 ≤ *øvreGrænse*), sæt *nedreGrænse*=0

poissPdf() Katalog >

poissPdf(λ**,***XVærdi***)**⇒*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∠θ]. Vektoren skal være af dimensionen 2 og kan være en række eller kolonne.

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 151.

$[1 \quad 3.] \blacktriangleright \text{Polar} \qquad [3.16228 \quad \angle 1.24905]$

$[x \quad y] \blacktriangleright \text{Polar} \qquad \left[\sqrt{x^2+y^2} \quad \angle \frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1}\left(\frac{x}{y}\right)\right]$

kompleksVærdi **▸Polar**

Viser *kompleksVærdi* i polær form.

- Vinkeltilstanden Grader returnerer (r∠θ).

I vinkeltilstanden Radian:

▸Polar

Katalog >

- Vinkeltilstanden Radian returnerer $re^{i\theta}$.

complekseVærdi 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.

$(3+4\cdot i)\blacktriangleright\text{Polar}$ $\quad e^{i\cdot\left(\frac{\pi}{2}-\tan^{-1}\left(\frac{3}{4}\right)\right)}\cdot 5$

$\left(\left(4\angle\frac{\pi}{3}\right)\right)\blacktriangleright\text{Polar}$ $\quad e^{\frac{i\cdot\pi}{3}}\cdot 4$

I vinkeltilstanden Nygrader:

$(4\cdot i)\blacktriangleright\text{Polar}$ $\quad (4\angle\ 100.)$

I vinkeltilstanden Grader:

$(3+4\cdot i)\blacktriangleright\text{Polar}$ $\quad \left(5\angle\ 90-\tan^{-1}\left(\frac{3}{4}\right)\right)$

polyCoeffs()

Katalog >

polyCoeffs(*Poly* [,*Var*]**)⇒***liste*

Returnerer en liste over koefficienterne i polynomiet *Poly* vedrørende variablen *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*.

$\text{polyCoeffs}((x+y+z)^2,x)$ $\quad \{1,2\cdot(y+z),(y+z)^2\}$

$\text{polyCoeffs}((x+y+z)^2,y)$ $\quad \{1,2\cdot(x+z),(x+z)^2\}$

$\text{polyCoeffs}((x+y+z)^2,z)$ $\quad \{1,2\cdot(x+y),(x+y)^2\}$

polyDegree() Katalog >

polyDegree(*Poly* [,*Var*])⇒*værdi*

Returnerer graden af det polynomielle udtryk *Poly* med hensyn til variablen *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 koefficenterne ikke kan. Det skyldes, at graden kan udtrækkes uden at ekspandere polynomiet.

polyEval() Katalog >

polyEval(*Liste1*, *Udtr1*)⇒*udtryk*

polyEval(*Liste1*, *Liste2*)⇒*udtryk*

Fortolker første argument som koefficienten til et polynomium i faldende grad og returnerer polynomiet beregnet for værdien af det andet argument.

$\text{polyEval}(\{a,b,c\},x)$	$a\cdot x^2+b\cdot x+c$
$\text{polyEval}(\{1,2,3,4\},2)$	26
$\text{polyEval}(\{1,2,3,4\},\{2,-7\})$	$\{26,-262\}$

polyGcd() Katalog >

polyGcd(*Udtryk1*,*Udtryk2*)⇒*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*]**)⇒***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*]**)⇒***udtryk*

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.

$\text{polyRemainder}(x-1,x-3)$ → 2

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

$\text{polyRemainder}(x^2-1,x-1)$ → 0

$\text{polyRemainder}((x-y)\cdot(y-z),x+y+z,x)$ → $-(y-z)\cdot(2\cdot y+z)$

$\text{polyRemainder}((x-y)\cdot(y-z),x+y+z,y)$ → $-2\cdot x^2-5\cdot x\cdot z-2\cdot z^2$

$\text{polyRemainder}((x-y)\cdot(y-z),x+y+z,z)$ → $(x-y)\cdot(x+2\cdot y)$

polyRoots()

Katalog >

polyRoots(*Poly*,*Var*) ⇒

polyRoots(*ListeAfKoeff*) ⇒

Den første syntaks, **polyRoots(***Poly*,*Var*), returnerer en liste af reelle rødder af polynomiet *Poly* med hensyn til variablen *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.

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

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

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

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

PowerReg

Katalog >

PowerReg *X*, *Y* [, *Frekv*] [, *Kategori*, *Medtag*]]

Beregner den lineære regressiony = (a · (x)^b)på liste *X* og *Y* med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 182.)

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.

Kategory 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 259.

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 *X-liste* der faktisk bruges i regressionen ud fra begrænsninger i *Frekv*, *kategoriliste*,og *Medtag Kategorier*
stat.YReg	Liste af datapunkter i den modificerede *Y-liste* der faktisk bruges i regressionen ud fra begrænsninger i *Frekv*, *kategoriliste*,og *Medtag Kategorier*
stat.FreqReg	Liste med hyppigheder, der svarer til *stat.XReg* og *stat.YReg*

Prgm

Katalog >

Prgm
Blok
EndPrgm

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.

Beregn GCD og vis mellemresultater.

```
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 Π(), side 231.

Product (PI)

Se Π(), side 231.

product()

Katalog >

product(*Liste*[**,** *Start*[**,** *slut*]]**)⇒***udtryk*

Returnerer produktet af elementerne indeholdt i *Liste*. *Start* og *Slut* er valgfri De angiver en serie af elementer.

product(*Matrix1*[**,** *Start*[**,** *slut*]]**)⇒***matrix*

Returnerer en rækkevektor med produkterne af elementerne i kolonnerne i *Matrix1*. *Start* og *slut* er valgfri. De angiver en serie af rækker.

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 259.

$\text{product}(\{1,2,3,4\})$ → 24

$\text{product}(\{2,x,y\})$ → $2\cdot x\cdot y$

$\text{product}(\{4,5,8,9\},2,3)$ → 40

$\text{product}\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}$ → $[28\ \ 80\ \ 162]$

$\text{product}\left(\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix},1,2\right)$ → $[4\ \ 10\ \ 18]$

propFrac()

Katalog >

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.

$\text{propFrac}\left(\frac{4}{3}\right)$ → $1+\frac{1}{3}$

$\text{propFrac}\left(\frac{-4}{3}\right)$ → $-1-\frac{1}{3}$

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

$\text{propFrac}(Ans)$ → $\frac{1}{x+1}+x+\frac{1}{y+1}+y$

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()**.

Q

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-triangulær.

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 ctrl enter 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 ·max(dim(*Matrix*)) ·rowNorm (*Matrix*)

Tallet med flydende komma (9). i m1 får resultaterne beregnet med flydende komma.

$$\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9.\end{bmatrix} \rightarrow m1 \qquad \begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9.\end{bmatrix}$$

$$\text{QR } m1,qm,rm \qquad \textit{Done}$$

$$qm = \begin{bmatrix}0.123091 & 0.904534 & 0.408248\\ 0.492366 & 0.301511 & -0.816497\\ 0.86164 & -0.301511 & 0.408248\end{bmatrix}$$

$$rm = \begin{bmatrix}8.12404 & 9.60114 & 11.0782\\ 0. & 0.904534 & 1.80907\\ 0. & 0. & 0.\end{bmatrix}$$

QR

Katalog >

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*.

$$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1 \qquad \begin{bmatrix} m & n \\ o & p \end{bmatrix}$$

QR *m1*,*qm*,*rm* — *Done*

$$qm = \begin{bmatrix} \frac{m}{\sqrt{m^2+o^2}} & \frac{-\text{sign}(m\cdot p-n\cdot o)\cdot o}{\sqrt{m^2+o^2}} \\ \frac{o}{\sqrt{m^2+o^2}} & \frac{m\cdot\text{sign}(m\cdot p-n\cdot o)}{\sqrt{m^2+o^2}} \end{bmatrix}$$

$$rm = \begin{bmatrix} \sqrt{m^2+o^2} & \frac{m\cdot n+o\cdot p}{\sqrt{m^2+o^2}} \\ 0 & \frac{|m\cdot p-n\cdot o|}{\sqrt{m^2+o^2}} \end{bmatrix}$$

QuadReg

Katalog >

QuadReg X**,** Y [**,** *Frekv*] [**,** *Kategori*, *Medtag*]]

Beregner andengrads polynomiel regression y = a · x² +b ·x+cpå listerne X og Y med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 182.)

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*angiverr frekvensen af hændelse for hver tilsvarende X og Y datapunkt. Standardværdien er 1. Alle elementer skal være heltal ≥ 0.

Kategory 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 259.

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 *Frekv*, *kategoriliste*,og *Medtag Kategorier*
stat.YReg	Liste af datapunkter i den modificerede *Y-liste* der faktisk bruges i regressionen ud fra begrænsninger i *Frekv*, *kategoriliste*,og *Medtag Kategorier*
stat.FreqReg	Liste med hyppigheder, der svarer til *stat.XReg* og *stat.YReg*

QuartReg

Katalog >

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 182.)

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*angiverr frekvensen af hændelse for hver tilsvarende *X* og *Y* datapunkt. Standardværdien er 1. Alle elementer skal være heltal ≥ 0.

Kategory 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.

QuartReg

Katalog >

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

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 *X-liste* der faktisk bruges i regressionen ud fra begrænsninger i *Frekv*, *kategoriliste*,og *Medtag Kategorier*
stat.YReg	Liste af datapunkter i den modificerede *Y-liste* der faktisk bruges i regressionen ud fra begrænsninger i *Frekv*, *kategoriliste*,og *Medtag Kategorier*
stat.FreqReg	Liste med hyppigheder, der svarer til *stat.XReg* og *stat.YReg*

R

R▶Pθ()

Katalog >

R▶Pθ (*xExpr*, *yExpr*) ⇒ *udtryk*

R▶Pθ (*xList*, *yList*) ⇒ *liste*
R▶Pθ (*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:

$$R\blacktriangleright P\theta(x,y) \qquad 90\cdot\text{sign}(y)-\tan^{-1}\left(\frac{x}{y}\right)$$

I vinkeltilstanden Nygrader:

$$R\blacktriangleright P\theta(x,y) \qquad 100\cdot\text{sign}(y)-\tan^{-1}\left(\frac{x}{y}\right)$$

I vinkeltilstanden Radian:

$$R\blacktriangleright P\theta(3,2) \qquad \tan^{-1}\left(\frac{2}{3}\right)$$

$$R\blacktriangleright P\theta\left(\begin{bmatrix}3 & -4 & 2\end{bmatrix},\begin{bmatrix}0 & \frac{\pi}{4} & 1.5\end{bmatrix}\right) \qquad \begin{bmatrix}0 & \tan^{-1}\left(\frac{16}{\pi}\right)+\frac{\pi}{2} & 0.643501\end{bmatrix}$$

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:

$\mathrm{R}\blacktriangleright\mathrm{Pr}(3,2)$ → $\sqrt{13}$

$\mathrm{R}\blacktriangleright\mathrm{Pr}(x,y)$ → $\sqrt{x^2+y^2}$

$\mathrm{R}\blacktriangleright\mathrm{Pr}\left(\begin{bmatrix}3 & -4 & 2\end{bmatrix},\begin{bmatrix}0 & \frac{\pi}{4} & 1.5\end{bmatrix}\right)$ → $\begin{bmatrix}3 & \frac{\sqrt{\pi^2+256}}{4} & 2.5\end{bmatrix}$

▶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:

$(1.5)\blacktriangleright\mathrm{Rad}$ → $(0.02618)^r$

I vinkeltilstanden Nygrader:

$(1.5)\blacktriangleright\mathrm{Rad}$ → $(0.023562)^r$

rand() Katalog >

rand() ⇒ *udtryk*
rand(*#Trials***)** ⇒ *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***)** ⇒ *udtryk*
randBin(*n***,** *p***,** *#Trials***)** ⇒ *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*) ⇒ *udtryk*
randInt(*lowBound*,*upBound*,*#Trials*) ⇒ *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 .

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 [μ−3•σ, μ+3•σ].

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(x,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 *list3*={1,2,3,4,5}	*Done*
Define *list4*=randSamp(*list3*,6)	*Done*
list4	{2,3,4,3,1,2}

RandSeed Katalog >

RandSeed *Number*

Hvis *Number* = 0, indstilles startværdien til fabriksindstillingerne for vilkårlig-tal-generatoren. Hvis *Number* ≠ 0, anvendes det til at generere to startværdier, der lagres i systemvariablerne seed1 og seed2.

RandSeed 1147	*Done*
rand()	0.158206

real() Katalog >

real(*Expr1***)** ⇒ *udtryk*

Returnerer den reelle del af argumentet.

Bemærk: Alle udefinerede variable behandles som reelle variable. Se også **imag()**, page 93.

real(2+3·*i*)	2
real(*z*)	*z*
real(*x*+*i*·*y*)	*x*

real(*List1***)** ⇒ *liste*

Returnerer realdelen af alle elementer.

real({*a*+*i*·*b*,3,*i*})	{*a*,3,0}

real(*Matrix1***)** ⇒ *matrix*

Returnerer realdelen af alle elementer.

$$\text{real}\left(\begin{bmatrix} a+i\cdot b & 3 \\ c & i \end{bmatrix}\right) \qquad \begin{bmatrix} a & 3 \\ c & 0 \end{bmatrix}$$

▶Rect

Katalog >

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 137.

$\left[3 \quad \angle\frac{\pi}{4} \quad \angle\frac{\pi}{6}\right]$▶Rect

$\left[\frac{3\cdot\sqrt{2}}{4} \quad \frac{3\cdot\sqrt{2}}{4} \quad \frac{3\cdot\sqrt{3}}{2}\right]$

$[a \quad \angle b \quad \angle c]$

$[a\cdot\cos(b)\cdot\sin(c) \quad a\cdot\sin(b)\cdot\sin(c) \quad a\cdot\cos(c)]$

complexValue ▶**Rect**

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∠θ) polær indtastning.

I vinkeltilstanden Radian:

$4\cdot e^{\frac{\pi}{3}}$▶Rect $\quad 4\cdot e^{\frac{\pi}{3}}$

$(4\angle\frac{\pi}{3})$▶Rect $\quad 2+2\cdot\sqrt{3}\cdot i$

I vinkeltilstanden Nygrader:

$(1\angle 100)$▶Rect $\quad i$

I vinkeltilstanden grader:

$(4\angle 60)$▶Rect $\quad 2+2\cdot\sqrt{3}\cdot i$

Bemærk: Du kan skrive ∠ ved at vælge det i symbollisten i kataloget.

ref()

Katalog >

ref(*Matrix1***[, *Tol*]) ⇒** *matrix*

Returnerer række-echelonformen af *Matrix1*.

$\text{ref}\begin{bmatrix}-2 & -2 & 0 & -6\\ 1 & -1 & 9 & -9\\ -5 & 2 & 4 & -4\end{bmatrix}$

$\begin{bmatrix}1 & \frac{-2}{5} & \frac{-4}{5} & \frac{4}{5}\\ 0 & 1 & \frac{4}{7} & \frac{11}{7}\\ 0 & 0 & 1 & \frac{-62}{71}\end{bmatrix}$

Ethvert matrixelement 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 ctrl enter eller indstiller **Auto- eller Approximate** - tilstanden til Approximate, foretages beregningerne med flydende komma.
- Hvis Tol udelades eller ikke anvendes, beregnes standardtolerancen som: 5E−14 •max(dim($Matrix1$)) •rowNorm ($Matrix1$)

Undgå udefinerede elementer i $Matrix1$. De kan føre til uventede resultater.

Hvis for eksempel a er udefineret i følgende udtryk, vil en advarsel blive vist, og resultatet vises som:

$$\text{ref}\left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) \qquad \begin{bmatrix} 1 & \frac{1}{a} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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.

$$\text{ref}\left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) | a=0 \qquad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Bemærk: Se også **rref()**, page 161.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow m1 \qquad \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\text{ref}(m1) \qquad \begin{bmatrix} 1 & \frac{d}{c} \\ 0 & 1 \end{bmatrix}$$

RefreshProbeVars

Giver dig adgang til sensordata fra alle tilkoblede sensorprober i dit TI-Basic-program.

StatusVar-værdi	Status
statusVar =0	Normal (fortsæt med programmet)
statusVar =1	Vernier DataQuest™-applikation er i dataindsamlingstilstand. **Bemærk:** Vernier DataQuest™-applikation skal være i målertilstand for at denne kommando virker.
statusVar =2	Vernier DataQuest™-applikation blev ikke startet.
statusVar =3	Vernier DataQuest™-applikationen blev startet, men du har ikke tilkoblet nogen prober.

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

Disp "For varmt"

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()

Katalog >

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•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 120.

$\text{remain}(7,0)$ → 7

$\text{remain}(7,3)$ → 1

$\text{remain}(-7,3)$ → -1

$\text{remain}(7,-3)$ → 1

$\text{remain}(-7,-3)$ → -1

$\text{remain}(\{12,-14,16\},\{9,7,-5\})$ → $\{3,0,1\}$

$$\text{remain}\left(\begin{bmatrix}9 & -7\\ 6 & 4\end{bmatrix},\begin{bmatrix}4 & 3\\ 4 & -3\end{bmatrix}\right) \rightarrow \begin{bmatrix}1 & -1\\ 2 & 1\end{bmatrix}$$

Request

Katalog >

Request *promptString***,** *var*[**,** *DispFlag* [**,** *statusVar*]]

Request *promptString***,** *func***(***arg1***,** *...argn***)** [**,** *DispFlag* [**,** *statusVar*]]

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 variablen *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.

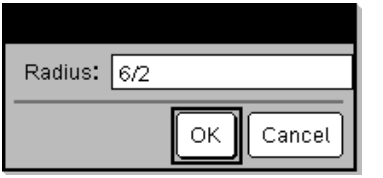
- Hvis *DispFlag* er udelades eller evalueres til **1**, vil klarmeddelelsen og brugerens svar blive vist i beregnerens historik.
- Hvis *DispFlag* evalueres til **0**, vil klarmeddelelsen og svaret ikke blive vist historikken.

Definer et program:

```
Define request_demo()=Prgm
    Request "Radius: ",r
    Disp "Areal = ",pi*r²
EndPrgm
```

Start programmet, og indtast et svar:

```
request_demo()
```



Resultat efter valg af **OK**:

```
Radius: 6/2
Areal = 28,2743
```

Request

Katalog >

Det valgfri 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 variablen *statusVar* til en værdi på **1**.
- I modsat fald indstilles variablen *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*

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 [on] nede, mens du gentagne gange trykker på [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.

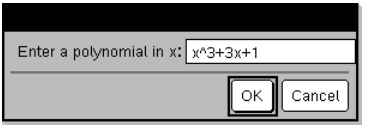
Bemærk: Se også **RequestStr**, page 155.

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:

```
polynomium()
```



Resultat efter indtastning af x^3+3x+1 og valg af **OK**:

```
Reelle rødder er: {-0.322185}
```

RequestStr

Katalog >

RequestStr *promptString*, *var*[, *DispFlag*]

Definer et program:

RequestStr

Katalog >

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:

- **Håndholdt:** Hold tasten [on] nede, mens du gentagne gange trykker på [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.

Bemærk: Se også **Request**, page 154.

```
Define requestStr_demo()=Prgm
    RequestStr "Dit navn:",name,0
    Disp "Svaret har ",dim(name),"
tegn."
EndPrgm
```

Start programmet, og indtast et svar:

```
requestStr_demo()
```

Resultat efter valg af **OK** (Bemærk, at argumentet *DispFlag* fra **0** udelader klarmeddelelsen og svaret fra historikken):

```
requestStr_demo()
                    Svaret har fem tegn.
```

Return

Katalog >

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() Katalog >

right(*List1*[, *Num*]**)** ⇒ *liste*

Returnerer *Num*-elementer længst til højre i *List1*.

Hvis du udelader *Num*, returneres hele *List1*.

right({1,3,⁻2,4},3)	{3,⁻2,4}

right(*sourceString*[, *Num*]**)** ⇒ *string*

Returnerer *Num*-tegn længst til højre i tegnstrengen *sourceString*.

Hvis du udelader *Num*, returneres hele *sourceString*.

right("Hello",2)	"lo"

right(*Comparison***)** ⇒ *udtryk*

Returnerer højre side af en ligning eller ulighed.

right(x<3)	3

rk23 () Katalog >

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

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

med *depVar*(*Var0*)=*depVar0* i intervallet [*Var0*,*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).

Differentialligning:

y'=0.001*y*(100-y) og y(0)=10

$$\text{rk23}(0.001\cdot y\cdot(100-y),t,y,\{0,100\},10,1)$$

$$\begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 10. & 10.9367 & 11.9493 & 13.042 & 14.2 \end{bmatrix}$$

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

$$\text{rk23}(0.001\cdot y\cdot(100-y),t,y,\{0,100\},10,1,1.\text{E}{-6})$$

$$\begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 10. & 10.9367 & 11.9495 & 13.0423 & 14.2189 \end{bmatrix}$$

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

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

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

rk23 ()

Katalog >

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.

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: sign(*VarStep*) = sign(*VarMax*-*Var0*) og løsninger returneres ved *Var0*+i**VarStep* for alle i=0,1,2,... således at *Var0*+i**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.

diftol er fejltolerancen (som standard 0.001).

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

$$\{10.,10.9367,11.9494,13.0423,14.2189,15.48$$

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 $y2(0)=5$

$$\text{rk23}\left(\begin{cases} -y1+0.1\cdot y1\cdot y2 \\ 3\cdot y2-y1\cdot y2 \end{cases},t,\{y1,y2\},\{0,5\},\{2,5\},1\right)$$

$$\begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 2. & 1.94103 & 4.78694 & 3.25253 & 1.82848 \\ 5. & 16.8311 & 12.3133 & 3.51112 & 6.27245 \end{bmatrix}$$

root()

Katalog >

root(*Expr***)** ⇒ *rod*
root(*Expr1***, *Expr2***)** ⇒ *rod*

root(*Expr*) returnerer kvadratroden af *Expr*.

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.

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

$$\sqrt[3]{3} \qquad 3^{\frac{1}{3}}$$

$$\sqrt[3]{3.} \qquad 1.44225$$

rotate(*Integer1*[**,***#ofRotations*]**)** ⇒ *heltal*

Roterer bittene i et binært heltal. Du kan indtaste *Integer1* i ethvert talsystem; det konverteres automatisk til en 64-bit binær form med fortegn. Hvis *Integer1* er for stort til denne form, bringer en symmetrisk modulo-operation værdien ind i det korrekte område. Der er flere oplysninger under ►**Base2**, side 18.

Hvis *#ofRotations* er positivt, kører rotationen mod venstre. Hvis *#ofRotations* er negativt, kører rotationen mod højre. Standardindstillingen er −1 (rotér en bit til højre).

For eksempel i en højrerotation:

Hver bit roterer til højre.

0b00000000000000011110101100000110101

Bitten længst til højre roterer længst mod venstre.

giver:

0b10000000000000011110101100000110101

Resultatet vises i den valgte tilstand for talsystem.

I binær tilstand:

rotate(0b11111111111111111111111111111111)	0b10000000000000000000000000000001
rotate(256,1)	0b1000000000

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

I hexadecimal tilstand:

rotate(0h78E)	0h3C7
rotate(0h78E,-2)	0h800000000000001E3
rotate(0h78E,2)	0h1E38

Vigtigt: Til binære eller hexadecimale indtastninger skal du som præfiks altid benytte henholdsvis 0b eller 0h (nul, ikke bogstavet O).

rotate(*List1*[**,***#ofRotations*]**)** ⇒ *liste*

Returnerer en kopi af *List1* roteret til højre eller venstre med *#of Rotations*-elementer. Ændrer ikke *List1*.

Hvis *#ofRotations* er positivt, kører rotationen mod venstre. Hvis *#of Rotations* er negativt, kører rotationen mod højre. Standardindstillingen er −1 (roter en bit til højre).

I decimal tilstand:

rotate({1,2,3,4})	{4,1,2,3}
rotate({1,2,3,4},-2)	{3,4,1,2}
rotate({1,2,3,4},1)	{2,3,4,1}

rotate(*String1*[**,***#ofRotations*]**)** ⇒ *string*

Returnerer en kopi af *String1* roteret til højre eller venstre med *#ofRotations*-tegn. Ændrer ikke *String1*.

rotate("abcd")	"dabc"
rotate("abcd",-2)	"cdab"
rotate("abcd",1)	"bcda"

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*]) ⇒** *udtryk*

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*]) ⇒** *liste*

Returnerer en liste med elementerne afrundet til det angivne antal cifre.

round(*Matrix1***[, *digits*]) ⇒** *matrix*

Returnerer en matrix med elementerne afrundet til det angivne antal cifre.

round(1.234567,3) 1.235

round({π,√2,ln(2)},4) {3.1416,1.4142,0.6931}

round([ln(5) ln(3); π e^1],1) [1.6 1.1; 3.1 2.7]

rowAdd() Katalog >

rowAdd(*Matrix1***, *rIndex1*, *rIndex2*) ⇒** *matrix*

Returnerer en kopi af *Matrix1* med rækken *rIndex2* udskiftet med summen af rækker *rIndex1* og *rIndex2*.

rowAdd([3 4; −3 −2],1,2) [3 4; 0 2]

rowAdd([a b; c d],1,2) [a b; a+c b+d]

rowDim() Katalog >

rowDim(*Matrix***) ⇒** *udtryk*

Returnerer antallet af rækker i *Matrix*.

Bemærk: Se også **colDim()**, side 27.

[1 2; 3 4; 5 6] → *m1* [1 2; 3 4; 5 6]

rowDim(*m1*) 3

rowNorm() Katalog >

rowNorm(*Matrix***)** ⇒ *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 27.

$$\text{rowNorm}\begin{bmatrix} -5 & 6 & -7 \\ 3 & 4 & 9 \\ 9 & -9 & -7 \end{bmatrix} \quad 25$$

rowSwap() Katalog >

rowSwap(*Matrix1***, *rIndex1*, *rIndex2*)** ⇒ *matrix*

Returnerer *Matrix1* med rækkerne *rIndex1* og *rIndex2* byttet.

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

$$\text{rowSwap}(mat,1,3) \quad \begin{bmatrix} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{bmatrix}$$

rref() Katalog >

rref(*Matrix1*[**,** *Tol*]**)** ⇒ *matrix*

Returnerer den reducerede række-echelonform af *Matrix1*.

$$\text{rref}\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & \frac{66}{71} \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$$

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*.

$$\text{rref}\begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Hvis du anvender ctrl eller indstiller **Auto- eller Approximate** -tilstanden til Approximate, foretages beregningerne med flydende komma.
- Hvis *Tol* udelades eller ikke anvendes, beregnes standardtolerancen som: 5E−14 •max(dim(*Matrix1*)) •rowNorm (*Matrix1*)

Bemærk: Se også **ref()**, page 151.

sec()

trig-tast

sec(*Udtr1***)** ⇒ *udtryk*

sec(*Liste1***)** ⇒ *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.

I vinkeltilstanden Grader:

$$\sec(45) \qquad \sqrt{2}$$

$$\sec(\{1,2.3,4\}) \qquad \left\{\frac{1}{\cos(1)}, 1.00081, \frac{1}{\cos(4)}\right\}$$

sec⁻¹()

trig-tast

sec⁻¹(*Udtr1***)** ⇒ *udtryk*

sec⁻¹(*Liste1***)** ⇒ *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(...)**.

I vinkeltilstanden Grader:

$$\sec^{-1}(1) \qquad 0$$

I vinkeltilstanden Nygrader:

$$\sec^{-1}(\sqrt{2}) \qquad 50$$

I vinkeltilstanden Radian:

$$\sec^{-1}(\{1,2,5\}) \qquad \left\{0, \frac{\pi}{3}, \cos^{-1}\left(\frac{1}{5}\right)\right\}$$

sech()

Katalog >

sech(*Udtr1***)** ⇒ *udtryk*

sech(*Liste1***)** ⇒ *liste*

Returnerer den hyperbolske sekans til *Udtr1* eller returnerer en liste med den hyperbolske sekans til elementerne i *Liste1*.

$$\operatorname{sech}(3) \qquad \frac{1}{\cosh(3)}$$

$$\operatorname{sech}(\{1,2.3,4\}) \qquad \left\{\frac{1}{\cosh(1)}, 0.198522, \frac{1}{\cosh(4)}\right\}$$

sech⁻¹() Katalog >

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:

$\text{sech}^{-1}(1)$ → 0

$\text{sech}^{-1}(\{1,-2,2.1\})$ → $\left\{0,\frac{2\cdot\pi}{3}\cdot i,8.\text{E}{-15}+1.07448\cdot i\right\}$

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 81), **GetStr** (side 88), og **eval()** (side 65).

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 *lightval*	*Done*
lightval	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.

n:=50	50
m:=4	4
Send "SET SOUND eval(m· n)"	*Done*
iostr.SendAns	"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.

$\text{seq}\left(n^2,n,1,6\right)$ $\quad \{1,4,9,16,25,36\}$

$\text{seq}\left(\frac{1}{n},n,1,10,2\right)$ $\quad \left\{1,\frac{1}{3},\frac{1}{5},\frac{1}{7},\frac{1}{9}\right\}$

$\text{sum}\left(\text{seq}\left(\frac{1}{n^2},n,1,10,1\right)\right)$ $\quad \frac{1968329}{1270080}$

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 ≈.

$\text{sum}\left(\text{seq}\left(\frac{1}{n^2},n,1,10,1\right)\right)$ $\quad 1.54977$

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 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.

Genererer de første fem led i sekvensen $u(n) = u(n\text{-}1)^2/2$ med $u(1)$=**2** og *VarStep*=**1**.

$\text{seqGen}\left(\frac{\left(u\left(n-1\right)\right)^2}{n},n,u,\{1,5\},\{2\}\right)$
$\left\{2,2,\frac{4}{3},\frac{4}{9},\frac{16}{405}\right\}$

Eksempel, hvor Var0=2:

$\text{seqGen}\left(\frac{u\left(n-1\right)+1}{n},n,u,\{2,5\},\{3\}\right)$
$\left\{3,\frac{4}{3},\frac{7}{12},\frac{19}{60}\right\}$

Eksempel, hvor den startværdien er symbolsk:

$\text{seqGen}\left(u\left(n-1\right)+2,n,u,\{1,5\},\{a\}\right)$
$\{a,a+2,a+4,a+6,a+8\}$

seqGen() Katalog >

Det oprindelige indhold af *Var* er uændret efter **seqGen()** er gennemført .

Standardværdien for *VarStep* = **1**.

System med to sekvenser:

$$\text{seqGen}\left\{\left\{\frac{1}{n},\frac{u2(n-1)}{2}+u1(n-1)\right\},n,\{u1,u2\},\{1,5\},\begin{bmatrix}\_\\2\end{bmatrix}\right\}$$

$$\begin{bmatrix}1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5}\\ 2 & 2 & \frac{3}{2} & \frac{13}{12} & \frac{19}{24}\end{bmatrix}$$

Bemærk: Void (_) i den oprindelige ledmatrix ovenfor bruges for at angive, at det oprindelige led for u1(n) er beregnet vha. den eksplicitte sekvensformel u1 (n)=1/n.

seqn() Katalog >

seqn(*Expr***(***u***,** *n* [**,** *ListOfInitTerms*[**,** *nMax* [**,** *CeilingValue*]]]**)**⇒*liste*

Genererer en liste med led for sekvensen *u* (*n*)=*Expr*(*u*, *n*) som følger: Øger *n* fra 1 til *nMax* med 1, beregner *u*(*n*) for de tilsvarende værdier af *n* vha. udtrykket *Expr* (*u*, *n*) og *ListOfInitTerms*, og returnerer resultaterne som en liste.

seqn(*Expr***(***n* [**,** *nMax* [**,** *CeilingValue*]]**)**⇒*liste*

Genererer en liste med led for sekvens *u* (*n*)=*Expr*(*n*) som følger: Øger *n* fra 1 til *nMax* med 1, beregner *u*(*n*) for de tilsvarende værdier af *n* vha. udtrykket *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**.

$$\text{seqn}\left\{\frac{u(n-1)}{n},\{2\},6\right\}$$

$$\left\{2,1,\frac{1}{3},\frac{1}{12},\frac{1}{60},\frac{1}{360}\right\}$$

$$\text{seqn}\left\{\frac{1}{n^2},6\right\} \qquad \left\{1,\frac{1}{4},\frac{1}{9},\frac{1}{16},\frac{1}{25},\frac{1}{36}\right\}$$

series(*Udtr1*, *Var*, *Orden* [, *Punkt*])⇒*udtryk*

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

series(*Udtr1*, *Var*, *Orden* [, *Punkt*]) | *Var*<*Punkt*⇒*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 −∞, 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 = ∞ eller −∞.

Hvis rækken eller en af dens afledte afbrydes af et spring ved *Punkt*, indeholder resultatet sandsynligvis deludtryk på formen sign(...) eller abs(...) for en reel udviklingsvariabel eller $(-1)^{\text{floor}(\ldots\text{angle}(\ldots)\ldots)}$ 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 "| *Var* > *Punkt*", "| *Var* < *Punkt*", "| "*Var* ≥ *Punkt*", eller "*Var* ≤ *Punkt*" for at opnå et enklere resultat.

$$\text{series}\left(\frac{1-\cos(x-1)}{(x-1)^2},x,4,1\right) \qquad \frac{1}{2}-\frac{(x-1)^2}{24}+\frac{(x-1)^4}{720}$$

$$\text{series}\left(\frac{-1}{e^{z_-}},z_-,1\right) \qquad z_-1$$

$$\text{series}\left(\left(1+\frac{1}{n}\right)^n,n,2,\infty\right) \qquad e-\frac{e}{2\cdot n}+\frac{11\cdot e}{24\cdot n^2}$$

$$\text{series}\left(\tan^{-1}\left(\frac{1}{x}\right),x,5\right)\Big|x>0 \qquad \frac{\pi}{2}-x+\frac{x^3}{3}-\frac{x^5}{5}$$

$$\text{series}\left(\int\frac{\sin(x)}{x}\,dx,x,6\right) \qquad x-\frac{x^3}{18}+\frac{x^5}{600}$$

$$\text{series}\left(\int_0^x \sin(x\cdot\sin(t))\,dt,x,7\right) \qquad \frac{x^3}{2}-\frac{x^5}{24}-\frac{29\cdot x^7}{720}$$

$$\text{series}\left(\left(1+e^x\right)^2,x,2,1\right)$$
$$(e+1)^2+2\cdot e\cdot(e+1)\cdot(x-1)+e\cdot(2\cdot e+1)\cdot(x-1)^2$$

series() Katalog >

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 taylor(...) giver,, så det dominerende led ikke er det længst til venstre.

Bemærk: Se også **dominantTerm()**, side 59.

setMode() Katalog >

setMode(*tilstandNavnHeltal***,** *indstilHeltal***)** ⇒*heltal* **setMode(***liste***)** ⇒*heltalsliste*

Kun gyldig i en funktion eller et program.

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.

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.

```
Define prog1()=Prgm                    Done
  Disp approx(π)
  setMode(1,16)
  Disp approx(π)
  EndPrgm
```

```
prog1()
                                       3.14159
                                       3.14
                                       Done
```

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 87.

Bemærk: De aktuelle tilstandsindstillinger videresendes til kaldte underrutiner. Hvis en underrutine 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.

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(*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 bitten længst mod højre, og 0 eller 1 indsættes for at matche bitten længst til venstre. I en flytning til venstre droppes bitten længst mod venstre, og 0 indsættes som bitten længst til højre.

For eksempel i en højreflytning:

Hver bit flytter til højre.

0b00000000000000111101011000011010

Indsætter 0, hvis bitten længst til venstre er 0,

eller 1, hvis bitten længst til venstre er 1.

giver:

00b00000000000000111101011000011101

Resultatet vises i den valgte tilstand for talsystem. Der vises ikke foranstillede nuller.

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 hexadecimale indtastninger skal du som præfiks altid benytte henholdsvis 0b eller 0h (nul, ikke bogstavet O).

shift(*Liste1* [**,***antalFlytninger*]**)⇒***liste*

Returnerer en kopi af *Liste1* flyttet til højre eller venstre med *antalflytninger* elementer. Ændrer ikke *Liste1*.

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() Katalog >

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.

shift("abcd")	" abc"
shift("abcd",-2)	" ab"
shift("abcd",1)	"bcd "

sign() Katalog >

sign(*Udtr1***)⇒***udtryk*

sign(*Liste1***)⇒***liste*

sign(*Matrix1***)⇒***matrix*

For reelle og komplekse *Udtr1*, returneres *Udtr1*/**abs(***Udtr1***)**, når *Udtr1*≠0.

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.

sign(-3.2)	-1.
sign({2,3,4,-5})	{1,1,1,-1}
sign(1+|x|)	1

Hvis kompleks formattilstand er real:

sign([-3 0 3])	[-1 ±1 1]

simult(*koeffMatrix***,** *konstVektor***[,** *Tol***])⇒***matrix*

Returnerer en kolonnevektor, der indeholder løsningerne til et system af lineære ligninger.

Bemærk: Se også **linSolve()**, side 106.

koeffMatrix skal være en kvadratisk matrix, der indeholder kofficienterne til ligningerne.

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*.

- 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 ·max(dim(*koeffMatrix*)) ·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 for x og y:

x + 2y = 1

3x + 4y = ⁻1

$$\text{simult}\left(\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix},\begin{bmatrix}1\\ -1\end{bmatrix}\right) \qquad \begin{bmatrix}-3\\ 2\end{bmatrix}$$

Løsningen er x=⁻3 og y=2.

Løs:

ax + by = 1

cx + dy = 2

$$\begin{bmatrix}a & b\\ c & d\end{bmatrix}\rightarrow matx1 \qquad \begin{bmatrix}a & b\\ c & d\end{bmatrix}$$

$$\text{simult}\left(matx1,\begin{bmatrix}1\\ 2\end{bmatrix}\right) \qquad \begin{bmatrix}\frac{-(2\cdot b-d)}{a\cdot d-b\cdot c}\\ \frac{2\cdot a-c}{a\cdot d-b\cdot c}\end{bmatrix}$$

Løs:

x + 2y = 1

3x + 4y = ⁻1

x + 2y = 2

3x + 4y = ⁻3

$$\text{simult}\left(\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix},\begin{bmatrix}1 & 2\\ -1 & -3\end{bmatrix}\right) \qquad \begin{bmatrix}-3 & -7\\ 2 & \frac{9}{2}\end{bmatrix}$$

For det første system er x=⁻3 og y=2. For det andet system er x=⁻7 og y=9/2.

▸sin

Katalog > 📖

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(...) modulo 1−sin(...)^2

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

Bemærk: Konverteringsoperatorer 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.

$(\cos(x))^2 \blacktriangleright \sin \qquad 1-(\sin(x))^2$

sin()

trig-tast

sin(*Udtr1***)**⇒*udtryk*

sin(*Liste1***)**⇒*liste*

sin(*Udtr1***)** returnerer sinus af argumentet som et udtryk.

sin(*Liste1***)** returnerer en liste med sinus til alle elementer i *Liste1*.

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 Grader:

$\sin\left(\frac{\pi}{4}r\right) \qquad \frac{\sqrt{2}}{2}$

$\sin(45) \qquad \frac{\sqrt{2}}{2}$

$\sin(\{0,60,90\}) \qquad \left\{0,\frac{\sqrt{3}}{2},1\right\}$

I vinkeltilstanden Nygrader:

$\sin(50) \qquad \frac{\sqrt{2}}{2}$

I vinkeltilstanden Radian:

sin()

trig-tast

$\sin\left(\frac{\pi}{4}\right)$ $\frac{\sqrt{2}}{2}$

$\sin(45°)$ $\frac{\sqrt{2}}{2}$

sin(*kvadratMatrix1***)⇒***kvadratMatrix***

Returnerer matrixsinus af *kvadratMatrix1*. Dette er ikke det samme som at beregne sinus for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

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

I vinkeltilstanden Radian:

$$\sin\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix}$$

$$\begin{bmatrix}0.9424 & -0.04542 & -0.031999\\ -0.045492 & 0.949254 & -0.020274\\ -0.048739 & -0.00523 & 0.961051\end{bmatrix}$$

$\sin^{-1}()$

trig-tast

sin⁻¹(*Udtr1***)⇒***udtryk*

sin⁻¹(*Liste1***)⇒***liste*

sin⁻¹(*Udtr1***)** returnerer den vinkel, hvis sinus er *Udtr1* som et udtryk.

sin⁻¹(*Liste1***)** returnerer en liste med de inverse sinusværdier 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 **`arcsin(…)`**.

sin⁻¹(*kvadratMatrix1***)⇒***kvadratMatrix*

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()**.

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

I vinkeltilstanden Grader:

$\sin^{-1}(1)$ 90

I vinkeltilstanden Nygrader:

$\sin^{-1}(1)$ 100

I vinkeltilstanden Radian:

$\sin^{-1}(\{0,0.2,0.5\})$ $\{0,0.201358,0.523599\}$

I vinkeltilstanden radian og tilstanden rektangulært komplekst format:

$$\sin^{-1}\begin{bmatrix}1 & 5\\ 4 & 2\end{bmatrix}$$

$$\begin{bmatrix}-0.174533-0.12198\cdot i & 1.74533-2.35591\cdot i\\ 1.39626-1.88473\cdot i & 0.174533-0.593162\cdot i\end{bmatrix}$$

sinh() Katalog >

sinh(*Udtr1***)⇒***udtryk*

sinh(*Liste1***)⇒***liste*

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*.

$\sinh(1.2)$ → 1.50946

$\sinh(\{0,1.2,3.\})$ → {0,1.50946,10.0179}

sinh(*kvadratMatrix1***)⇒***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()**.

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

I vinkeltilstanden Radian:

$$\sinh\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix} \rightarrow \begin{bmatrix}360.954 & 305.708 & 239.604\\ 352.912 & 233.495 & 193.564\\ 298.632 & 154.599 & 140.251\end{bmatrix}$$

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

sinh⁻¹(*Udtr1***)⇒***udtryk*

sinh⁻¹(*Liste1***)⇒***liste*

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(...)`**.

$\sinh^{-1}(0)$ → 0

$\sinh^{-1}(\{0,2.1,3\})$ → $\{0,1.48748,\sinh^{-1}(3)\}$

sinh⁻¹(*kvadratMatrix1***)⇒***kvadratMatrix*

Returnerer den matrixinverse 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()**.

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

I vinkeltilstanden Radian:

$$\sinh^{-1}\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix} \rightarrow \begin{bmatrix}0.041751 & 2.15557 & 1.1582\\ 1.46382 & 0.926568 & 0.112557\\ 2.75079 & -1.5283 & 0.57268\end{bmatrix}$$

SinReg *X*, *Y* [, [*Iterationer*],[*Periode*] [, *Kategori*, *Medtag*]]

Beregner sinusregressionen på listerne *X* og *Y*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 182.)

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.

Kategory 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 259.

Output-variabel	**Beskrivelse**
stat.RegEqn	Regressionsligning: a · sin(bx+c)+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 *X-liste* der faktisk bruges i regressionen ud fra begrænsninger i *Frekv*, *kategoriliste*,og *Medtag Kategorier*
stat.YReg	Liste af datapunkter i den modificerede *Y-liste* der faktisk bruges i regressionen ud fra begrænsninger i *Frekv*, *kategoriliste*,og *Medtag Kategorier*
stat.FreqReg	Liste med hyppigheder, der svarer til *stat.XReg* og *stat.YReg*

solve() Katalog >

solvee(*Ligning*, *Var***)⇒***Boolesk udtryk***

solve(*Ligning*, *Var*=*Gæt***)⇒***Boolesk udtryk***

solve(*Ulighed*, *Var***)⇒***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.

$\text{solve}(a\cdot x^2+b\cdot x+c=0,x)$

$x=\frac{\sqrt{b^2-4\cdot a\cdot c}-b}{2\cdot a}$ or $x=\frac{-\left(\sqrt{b^2-4\cdot a\cdot c}+b\right)}{2\cdot a}$

Løsninger kan være ikke-reelle for visse kombinationer af værdier for udefinerede variable.

$Ans|a=1$ and $b=1$ and $c=1$

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

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

$\text{solve}\left((x-a)\cdot e^x=-x\cdot(x-a),x\right)$

$x=a$ or $x=-0.567143$

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.

$(x+1)\cdot\frac{x-1}{x-1}+x-3$ $2\cdot x-2$

For uligheder af typerne ≥, ≤, <, eller >, er eksplicitte løsninger usandsynlige, medmindre uligheden er lineær og kun indeholder *Var*.

$\text{solve}(5\cdot x-2\geq 2\cdot x,x)$

$x\geq\frac{2}{3}$

For Eksakt-indstillingen af **Auto eller tilnærmet**-tilstanden returneres dele, der ikke kan løses som en implicit ligning eller ulighed.

$\text{exact}\left(\text{solve}\left((x-a)\cdot e^x=-x\cdot(x-a),x\right)\right)$

$e^x+x=0$ or $x=a$

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 ulighedsoperatorerne til at udelukke dette interval fra efterfølgende søgninger.

I vinkeltilstanden Radian:

$\text{solve}\left(\tan(x)=\frac{1}{x},x\right)\mid x>0 \text{ and } x<1$ $\quad x=0.860334$

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.

$\text{solve}(x=x+1,x)$ false

$\text{solve}(x=x,x)$ true

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.

$2\cdot x-1\le 1 \text{ and } \text{solve}(x^2\ne 9,x)$ $\quad x\ne -3 \text{ and } x\le 1$

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 vinkeltilstanden Radian:

$\text{solve}(\sin(x)=0,x)$ $\quad x=n1\cdot\pi$

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.

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

$\text{solve}(\sqrt{x}=-2,x)$ false

$\text{solve}(-\sqrt{x}=-2,x)$ $\quad x=4$

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

solve(*Lign1* **and** *Lign2* [**and**...], *VarEllerGæt1*, *VarEllerGæt2* [**,** ...]**)**⇒*Boolesk - udtryk*

$\text{solve}(y=x^2-2 \text{ and } x+2\cdot y=-1,\{x,y\})$

$x=\frac{-3}{2} \text{ and } y=\frac{1}{4} \text{ or } x=1 \text{ and } y=-1$

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.

Du kan adskille ligningerne med **and**-operatoren, eller du kan indtaste et *SystemAfLign* 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

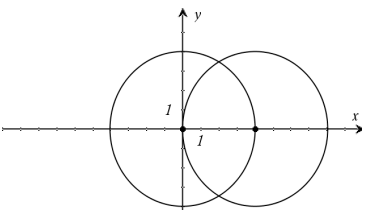
- eller -

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.

$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y\}\right)$$
$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ or } x=\frac{r}{2} \text{ and } y=\frac{-\sqrt{3}\cdot r}{2}$$

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.

$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y,z\}\right)$$
$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ and } z=c1 \text{ or } x=\frac{r}{2} \text{ and } y\text{▸}$$

Cylinderløsningerne illustrerer, hvordan familier af løsninger kan indeholde arbitrære konstanter af formen ck, hvor k 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 hukomvarEllerGæt 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-polynomiel 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.

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

$$\text{solve}\left(x+e^{z}\cdot y=1 \text{ and } x-y=\sin(z),\{x,y\}\right)$$
$$x=\frac{e^{z}\cdot\sin(z)+1}{e^{z}+1} \text{ and } y=\frac{-(\sin(z)-1)}{e^{z}+1}$$

$$\text{solve}\left(e^{z}\cdot y=1 \text{ and } -y=\sin(z),\{y,z\}\right)$$
$$y=2.812\text{E}{-10} \text{ and } z=21.9911 \text{ or } y=0.001871\ldots$$

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.

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.

$\{2,1,4,3\}\rightarrow list1$	$\{2,1,4,3\}$
SortA *list1*	*Done*
list1	$\{1,2,3,4\}$
$\{4,3,2,1\}\rightarrow list2$	$\{4,3,2,1\}$
SortA *list2*,*list1*	*Done*
list2	$\{1,2,3,4\}$
list1	$\{4,3,2,1\}$

SortA

Katalog >

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 259.

SortD

Katalog >

SortD *Liste1*[**,** *Liste2*] [**,** *Liste3*] ...

SortD *Vektor1*[**,***Vektor2*] [**,***Vektor3*] ...

Identisk med **SortA**, med den undtagelse, at **SortD** sorterer elementerne i faldende rækkefølge.

tomme (ugyldige) elementer i det første argument flyttes til bunden. Yderligere oplysninger om tomme elementer findes på side 259.

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
$\{1,2,3,4\} \rightarrow list2$	$\{1,2,3,4\}$
SortD *list1*,*list2*	*Done*
list1	$\{4,3,2,1\}$
list2	$\{3,4,1,2\}$

▸Sphere

Katalog >

Vektor **▸Sphere**

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

Viser række- eller kolonnevektoren i sfærisk form [ρ ∠θ ∠ϕ].

Vektor skal have dimensionen 3 og kan være enten en række- eller kolonnevektor.

Bemærk: ▸Sphere er en displayformat-kommando, ikke en konverteringsfunktion. Du kan kun anvende den i slutningen af en indtastningslinje.

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 ≈.

$$\begin{bmatrix}1 & 2 & 3\end{bmatrix} \blacktriangleright \text{Sphere}$$
$$\begin{bmatrix}3.74166 & \angle 1.10715 & \angle 0.640522\end{bmatrix}$$

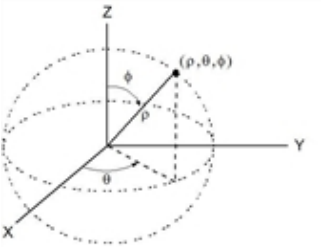
$$\left(\begin{bmatrix}2 & \angle \frac{\pi}{4} & 3\end{bmatrix}\right) \blacktriangleright \text{Sphere}$$
$$\begin{bmatrix}3.60555 & \angle 0.785398 & \angle 0.588003\end{bmatrix}$$

Tryk enter

▸Sphere

Katalog >

$$\left\{\left[2 \quad \angle\frac{\pi}{4} \quad 3\right]\right\}\blacktriangleright\text{Sphere}$$
$$\left[\sqrt{13} \quad \angle\frac{\pi}{4} \quad \angle\sin^{-1}\left(\frac{2\cdot\sqrt{13}}{13}\right)\right]$$



sqrt()

Katalog >

sqrt(*Udtr1***)⇒***udtryk*

sqrt(*Liste1***)⇒***liste*

Returnerer kvadratroden af argumentet.

For en liste returneres kvadratrødderne af alle elementer i *Liste1*.

Bemærk: Se også **Kvadratrodsskabelon**, side 1.

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

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
"r²"	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.AdjR²	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.r²	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.σx	stat.SSRow
stat.b5	stat.DW	stat.MSInteract	stat.σy	stat.tList
stat.b6	stat.e	stat.MSReg	stat.σx1	stat.UpperPred
stat.b7	stat.ExpMatrix	stat.MSRow	stat.σx2	stat.UpperVal
stat.b8	stat.$\bar{F}$	stat.n	stat.Σx	stat.$\bar{x}$
stat.b9	stat.$\bar{F}$Block	stat.$\hat{p}$	stat.Σx²	stat.$\bar{x}$1
stat.b10	stat.$\bar{F}$col	stat.$\hat{p}$1	stat.Σxy	stat.$\bar{x}$2
stat.bList	stat.$\bar{F}$Interact	stat.$\hat{p}$2	stat.Σy	stat.$\bar{x}$Diff
stat.χ^2	stat.FreqReg	stat.$\hat{p}$Diff	stat.Σy²	stat.$\bar{x}$List
stat.c	stat.$\bar{F}$row	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

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.

```
Se eksemplet med stat.results.
```

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*.

Bemærk: *Liste* skal have mindst to elementer. Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 259

I vinkeltilstanden Radian og tilstanden Auto:

$$\text{stDevPop}(\{a,b,c\}) \qquad \frac{\sqrt{2\cdot(a^2-a\cdot(b+c)+b^2-b\cdot c+c^2)}}{3}$$

$$\text{stDevPop}(\{1,2,5,-6,3,-2\}) \qquad \frac{\sqrt{465}}{6}$$

$$\text{stDevPop}(\{1.3,2.5,-6.4\},\{3,2,5\}) \qquad 4.11107$$

stDevPop() Katalog >

stDevPop(*Matrix1***[,** *Hyppighedsmatrix***])⇒***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 259.

$$\text{stDevPop}\left(\begin{bmatrix}1 & 2 & 5\\ -3 & 0 & 1\\ 5 & 7 & 3\end{bmatrix}\right) \quad \left[\frac{4\cdot\sqrt{6}}{3} \quad \frac{\sqrt{78}}{3} \quad \frac{2\cdot\sqrt{6}}{3}\right]$$

$$\text{stDevPop}\left(\begin{bmatrix}-1.2 & 5.3\\ 2.5 & 7.3\\ 6 & -4\end{bmatrix},\begin{bmatrix}4 & 2\\ 3 & 3\\ 1 & 7\end{bmatrix}\right) \quad \begin{bmatrix}2.52608 & 5.21506\end{bmatrix}$$

stDevSamp() Katalog >

stDevSamp(*Liste***[,** *Hyppighedsliste***])⇒***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 259

$$\text{stDevSamp}(\{a,b,c\}) \quad \frac{\sqrt{3\cdot\left(a^2-a\cdot(b+c)+b^2-b\cdot c+c^2\right)}}{3}$$

$$\text{stDevSamp}(\{1,2,5,-6,3,-2\}) \quad \frac{\sqrt{62}}{2}$$

$$\text{stDevSamp}(\{1.3,2.5,-6.4\},\{3,2,5\}) \quad 4.33345$$

stDevSamp(*Matrix1***[,** *Hyppighedsmatrix***])⇒***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*.

Bemærk: *Matrix1* skal have mindst to rækker. Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 259.

$$\text{stDevSamp}\left(\begin{bmatrix}1 & 2 & 5\\ -3 & 0 & 1\\ 5 & 7 & 3\end{bmatrix}\right) \quad \begin{bmatrix}4 & \sqrt{13} & 2\end{bmatrix}$$

$$\text{stDevSamp}\left(\begin{bmatrix}-1.2 & 5.3\\ 2.5 & 7.3\\ 6 & -4\end{bmatrix},\begin{bmatrix}4 & 2\\ 3 & 3\\ 1 & 7\end{bmatrix}\right) \quad \begin{bmatrix}2.7005 & 5.44695\end{bmatrix}$$

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.

$i:=0$	0
Define $prog1()$=Prgm For i,1,10,1 If i=5 Stop EndFor EndPrgm	*Done*
$prog1()$	*Done*
i	5

Store

Se → (store), side 241.

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($m1$,2,1,3,2)	$\begin{bmatrix}4&5\\7&8\end{bmatrix}$
subMat($m1$,2,2)	$\begin{bmatrix}5&6\\8&9\end{bmatrix}$

Sum (Sigma)

Se G(), side 232.

sum() Katalog >

sum(*Liste*[**,** *Start*[**,** *Slut*]]**)⇒***udtryk*

Returnerer summen af elementer i *Liste*.

Start og *Slut* er valgfri. De angiver en serie af elementer.

Alle ugyldige argumenter giver et ugyldigt resultat. Tomme (ugyldige) elementer i *List* ignoreres. Yderligere oplysninger om tomme elementer findes på side 259.

$\text{sum}(\{1,2,3,4,5\})$ 15

$\text{sum}(\{a,2\cdot a,3\cdot a\})$ $6\cdot a$

$\text{sum}(\text{seq}(n,n,1,10))$ 55

$\text{sum}(\{1,3,5,7,9\},3)$ 21

sum(*Matrix1*[**,** *Start*[**,** *Slut*]]**)⇒***matrix*

Returnerer en rækkevektor med summerne af elementerne i kolonnerne i *Matrix1*.

Start og *Slut* er valgfri. De angiver en serie af rækker.

Alle ugyldige argumenter giver et ugyldigt resultat. Tomme (ugyldige) elementer i *Matrix1* ignoreres. Yderligere oplysninger om tomme elementer findes på side 259.

$\text{sum}\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix}$ $\begin{bmatrix}5&7&9\end{bmatrix}$

$\text{sum}\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}$ $\begin{bmatrix}12&15&18\end{bmatrix}$

$\text{sum}\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix},2,3)$ $\begin{bmatrix}11&13&15\end{bmatrix}$

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:

- 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.

$\text{sumIf}(\{1,2,\mathbf{e},3,\pi,4,5,6\},2.5<?<4.5)$

$\mathbf{e}+\pi+7$

$\text{sumIf}(\{1,2,3,4\},2<?<5,\{10,20,30,40\})$

70

sumIf() Katalog >

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 259.

Bemærk: Se også **countIf()**, side 37.

sumSeq() Se Σ(), side 232.

system() Katalog >

system(*Udtr1* **[, ***Udtr2* **[, ***Udtr3* **[, ...]]])**

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.

$\text{solve}\left(\begin{cases} x+y=0 \\ x-y=8 \end{cases}, x, y\right)$ $x=4$ and $y=-4$

T

T (transponere) Katalog >

Matrix1^{T}⇒*matrix*

Returnerer den kompleks konjugerede transponerede *Matrix1*.

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

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}^{\mathsf{T}}$ $\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$

$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{\mathsf{T}}$ $\begin{bmatrix} a & c \\ b & d \end{bmatrix}$

$\begin{bmatrix} 1+i & 2+i \\ 3+i & 4+i \end{bmatrix}^{\mathsf{T}}$ $\begin{bmatrix} 1-i & 3-i \\ 2-i & 4-i \end{bmatrix}$

tan()

trig-tast

tan(*Udtr1***)⇒***udtryk*

tan(*Liste1***)⇒***liste*

tan(*Udtr1***)** returnerer tangens af argumentet som et udtryk.

tan(*Liste1***)** returnerer en liste med tangens til alle elementer i *Liste1*.

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 Grader:

$\tan\left(\frac{\pi}{4}_r\right)$ 1

$\tan(45)$ 1

$\tan(\{0,60,90\})$ $\{0,\sqrt{3},\text{undef}\}$

I vinkeltilstanden Nygrader:

$\tan\left(\frac{\pi}{4}_r\right)$ 1

$\tan(50)$ 1

$\tan(\{0,50,100\})$ $\{0,1,\text{undef}\}$

I vinkeltilstanden Radian:

$\tan\left(\frac{\pi}{4}\right)$ 1

$\tan(45°)$ 1

$\tan\left(\left\{\pi,\frac{\pi}{3},-\pi,\frac{\pi}{4}\right\}\right)$ $\{0,\sqrt{3},0,1\}$

tan(*kvadratMatrix1***)⇒***kvadratMatrix*

Returnerer matrixtangens til *kvadratMatrix1*. Dette er ikke det samme som at beregne tangens for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

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

I vinkeltilstanden Radian:

$$\tan\left(\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix}\right)$$

$$\begin{bmatrix}-28.2912 & 26.0887 & 11.1142\\ 12.1171 & -7.83536 & -5.48138\\ 36.8181 & -32.8063 & -10.4594\end{bmatrix}$$

tan⁻¹()

trig-tast

tan⁻¹(*Udtr1***)⇒***udtryk*

tan⁻¹(*Liste1***)⇒***list*

tan⁻¹(*Udtr1***)** returnerer vinklen, hvis tangens er *Udtr1*, som et udtryk.

tan⁻¹(*Liste1***)** returnerer en liste med den inverse tangens til hvert element af *Liste1*.

I vinkeltilstanden Grader:

$\tan^{-1}(1)$ 45

I vinkeltilstanden Nygrader:

tan⁻¹() trig-tast

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 **arctan(…)**.

tan⁻¹(*kvadratMatrix1***)⇒***kvadratMatrix***

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()**.

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

$\tan^{-1}(1)$ → 50

I vinkeltilstanden Radian:

$\tan^{-1}(\{0,0.2,0.5\})$ → $\{0,0.197396,0.463648\}$

I vinkeltilstanden Radian:

$$\tan^{-1}\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix} \rightarrow \begin{bmatrix}-0.083658 & 1.26629 & 0.62263\\ 0.748539 & 0.630015 & -0.070012\\ 1.68608 & -1.18244 & 0.455126\end{bmatrix}$$

tangentLine() Katalog >

tangentLine(*Udtr1***,***Var***,***Punkt***)⇒***udtryk***

tangentLine(*Udtr1***,***Var***,***Punkt***)⇒***udtryk***

Returnerer tangentlinjen 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 **tangentLine(**f1(x),x,2**)** "false."

$\text{tangentLine}(x^2,x,1)$ → $2\cdot x-1$

$\text{tangentLine}((x-3)^2-4,x=3)$ → -4

$\text{tangentLine}\left(x^{\frac{1}{3}},x=0\right)$ → $x=0$

$\text{tangentLine}(\sqrt{x^2-4},x=2)$ → undef

$x:=3:\ \text{tangentLine}(x^2,x,1)$ → 5

tanh() Katalog >

tanh(*Udtr1***)⇒***udtryk***

tanh(*Liste1***)⇒***liste***

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***

$\tanh(1.2)$ → 0.833655

$\tanh(\{0,1\})$ → $\{0,\tanh(1)\}$

I vinkeltilstanden Radian:

tanh() Katalog >

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()**.

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

$$\tanh\left(\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix}\right)$$

$$\begin{bmatrix}-0.097966 & 0.933436 & 0.425972\\ 0.488147 & 0.538881 & -0.129382\\ 1.28295 & -1.03425 & 0.428817\end{bmatrix}$$

$\tanh^{-1}()$ Katalog >

$\tanh^{-1}$(*Udtr1***)⇒***udtryk***

$\tanh^{-1}$(*Liste1***)⇒***liste***

$\tanh^{-1}$(*Udtr1***)** returnerer den inverse hyperbolske tangens af argumentet som et udtryk.

$\tanh^{-1}$(*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(...)`**.

I rektangulært komplekst format:

$$\tanh^{-1}(0) \qquad 0$$

$$\tanh^{-1}(\{1,2.1,3\})$$

$$\left\{\text{undef}, 0.518046-1.5708\cdot i, \frac{\ln(2)}{2}-\frac{\pi}{2}\cdot i\right\}$$

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

$\tanh^{-1}$(*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.

I vinkeltilstanden Radian og rektangulært komplekst format:

$$\tanh^{-1}\left(\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix}\right)$$

$$\begin{bmatrix}-0.099353+0.164058\cdot i & 0.267834-1.4908\\ -0.087596-0.725533\cdot i & 0.479679-0.9473\\ 0.511463-2.08316\cdot i & -0.878563+1.7901\end{bmatrix}$$

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.

$\text{taylor}\left(e^{\sqrt{x}},x,2\right)$ $\text{taylor}\left(e^{\sqrt{x}},x,2,0\right)$

$\text{taylor}\left(e^{t},t,4\right)\left|t=\sqrt{x}\right.$ $\frac{x^2}{24}+\frac{x^{\frac{3}{2}}}{6}+\frac{x}{2}+\sqrt{x}+1$

$\text{taylor}\left(\frac{1}{x\cdot(x-1)},x,3\right)$ $\text{taylor}\left(\frac{1}{x\cdot(x-1)},x,3,0\right)$

$\text{expand}\left(\frac{\text{taylor}\left(\frac{x}{x\cdot(x-1)},x,4\right)}{x},x\right)$ $-x^3-x^2-x-\frac{1}{x}-1$

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 ≤ *øvreGrænse*), sæt *nedreGrænse*= −∞.

tCollect() Katalog >

tCollect(*Udtr1)⇒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.

$\text{tCollect}\left(\left(\cos(\alpha)\right)^2\right)$ $\frac{\cos(2\cdot\alpha)+1}{2}$

$\text{tCollect}\left(\sin(\alpha)\cdot\cos(\beta)\right)$ $\frac{\sin(\alpha-\beta)+\sin(\alpha+\beta)}{2}$

tCollect() Katalog >

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.

tExpand Katalog >

tExpand(*Udtr1***)⇒***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.

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 π/180 forstyrrer funktionen i **tExpand()**, der genkender ekspanderbare former. For at få de bedste resultater skal **tExpand()** anvendes i Radian-tilstand.

$\text{tExpand}(\sin(3\cdot\varphi))$ $\quad 4\cdot\sin(\varphi)\cdot(\cos(\varphi))^2-\sin(\varphi)$

$\text{tExpand}(\cos(\alpha-\beta))$ $\quad \cos(\alpha)\cdot\cos(\beta)+\sin(\alpha)\cdot\sin(\beta)$

Text Katalog >

Text*promptStreng*[**,** *DispFlag*]

Programmeringskommando: Standser programmet midlertidigt og viser tegnstrengen *promptStreng* i en dialogboks.

Når en bruger vælger **OK**, fortsætter programudførelsen.

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**.

Text

Katalog >

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 155 eller **RequestStr**, side 155.

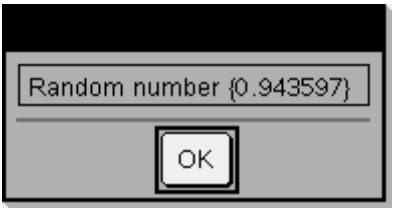
Bemærk: Du kan bruge denne kommando i et brugerdefineret program men ikke i en funktion.

```
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:



Then

Se If, side 91.

tInterval

Katalog >

tInterval *Liste*[**,***Frekv*[**,***CNiveau*]]

(Datalisteinput)

tInterval $\bar{x}$**,***sx***,***n***[,***CNiveau***]**

(Sammenfatning, stat input)

Beregner et *t* konfidensinterval En sammenfatning af resultaterne lagres i *stat.results* variable. (side 182.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

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.σx	Stikprøve standardafvigelse
stat.n	Længde på datasekvensen med målingsgennemsnit

tInterval_2Samp

Katalog >

tInterval_2Samp *Listet1*,*Liste2*
[,*Hyppighed1*[,*Hyppighed12*[,*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 182.)

Puljet=**1** puljer varianser. *Puljet*=**0** puljer ikke varianser.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

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.σx1, stat.σx2	Stikprøve standardafvigelser for *Liste 1* og *Liste 2*

Output-variabel	Beskrivelse
stat.n1, stat.n2	Antal målinger i datasekvenserne
stat.sp	Den puljede standardafvigelse. Beregnes, når *Puljet* = 1.

tmpCnv() Katalog >

tmpCnv(*Udtr_°tempEnhed***, _ °***tempEnhed2***) ⇒***udtryk _°tempEnhed2***

Konverterer en temperaturværdi med *Udtr* mellem enhederne. De gyldige enheder for temperatur er:

_°C Celsius

_°F Fahrenheit

_°K Kelvin

_°R Rankine

Skriv ° ved at vælge den blandt Katalogsymboler.

Skriv _ ved at trykke på ctrl ␣.

For eksempel konverteres 100_°C til 212_ °F.

Et temperaturområde konverteres i stedet med Δ**tmpCnv()**.

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.

ΔtmpCnv() Katalog >

Δ**tmpCnv(***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

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.

ΔtmpCnv() Katalog >

_°F Fahrenheit

_°K Kelvin

_°R Rankine

Du kan skrive ° ved at vælge det på symbolpaletten eller skrive @d.

Skriv _ ved at trykke på ctrl ␣.

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.

For at konvertere en bestemt temperaturværdi i stedet for et område skal du anvende **tmpCnv()**.

tPdf() Katalog >

tPdf(*XVærdi*,*fg*)⇒*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***)⇒***udtryk*

Returner sporet (sum af alle elementer på hoveddiagonalen) af *kvadratMatrix*.

$\text{trace}\left(\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}\right)$ 15

$\text{trace}\left(\begin{bmatrix}a & 0\\ 1 & a\end{bmatrix}\right)$ $2\cdot a$

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 269.

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.

```
Define prog1()=Prgm
                Try
                z:=z+1
                Disp "z incremented."
                Else
                Disp "Sorry, z undefined."
                EndTry
                EndPrgm
                                    Done
z:=1:prog1()
                        z incremented.
                                    Done
DelVar z:prog1()
                        Sorry, z undefined.
                                    Done
```

Eksempel 2

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.

$$eigenvals\left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix}\right)$$

$$eigenvals\left(\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$$

Bemærk: Se også **ClrErr**, side 26, og **PassErr**, side 136.

```
Define eigenvals(a,b)=Prgm

© Program eigenvals(A,B) viser eigenværdier for A·B

Try

  Disp "A= ",a

  Disp "B= ",b

  Disp " "

  Disp "Eigenværdier 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
```

Try

Katalog >

EndTry

EndPrgm

tTest

Katalog >

tTest μ*0*,*Liste*[,*Hyppighed*[,*Hypot*]]

(Datalisteinput)

tTest μ*0*,$\bar{x}$,*sx*,*n*,[*Hypot*]

(Sammenfatning, stat input)

Udfører en hypotesetest for et enkelt ubekendt populationsgennemsnit μ, når populationens standardafvigelseσ er ubekendt. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 182.)

Test H_0: μ = μ0, mod en af de følgende:

Til H_a: μ < μ0, sæt *Hypot*<0

til H_a: μ ≠ μ0 (standard), sæt *Hypot*=0

til H_a: μ > μ0, sæt *Hypot*>0

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

Output-variabel	Beskrivelse
stat.t	($\bar{x}$ − μ0) / (stdafvigelse / kvr(n))
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.fg	Frihedsgrader
stat.$\bar{x}$	Middelværdi af stikprøver af datasekvensen i *Liste*
stat.sx	Standardmåleafvigelse for datasekvensen
stat.n	Stikprøvens størrelse

tTest_2Samp

Katalog >

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 182.)

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 259.

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 *Liste 1* og *Liste 2*
stat.sx1, stat.sx2	Stikprøve standardafvigelse for datasekvenserne i *Liste 1* og *Liste 2*
stat.n1, stat.n2	Størrelse på stikprøverne
stat.sp	Den puljede standardafvigelse. Beregnes, når *Puljet*=1.

tvmFV() Katalog >

tvmFV(*N*,*I*,*PV*,*Pmt*,[*PpY*],[*CpY*],[*PmtAt*]**)**⇒*værdi*

tvmFV(120,5,0,-500,12,12) 77641.1

Finansfunktion, der beregner penges fremtidige værdi.

tvmFV()

Katalog >

Bemærk: Argumenter, der bruges i TVM-funktionerne er beskrevet i tabellen over TVM-argumenter, side 201. Se også **amortTbl()**, side 8.

tvmI()

Katalog >

tvmI(*N*,*PV*,*Pmt*,*FV*,[*PpY*],[*CpY*],[*PmtAt*]**)**⇒*værdi*

Finansfunktion, der beregner den årlige rente.

Bemærk: Argumenter, der bruges i TVM-funktionerne er beskrevet i tabellen over TVM-argumenter, side 201. Se også **amortTbl()**, side 8.

```
tvmI(240,100000,-1000,0,12,12)        10.5241
```

tvmN()

Katalog >

tvmN(*I*,*PV*,*Pmt*,*FV*,[*PpY*],[*CpY*],[*PmtAt*]**)**⇒*værdi*

Finansfunktion, der beregner antallet af betalingsperioder.

Bemærk: Argumenter, der bruges i TVM-funktionerne er beskrevet i tabellen over TVM-argumenter, side 201. Se også **amortTbl()**, side 8.

```
tvmN(5,0,-500,77641,12,12)            120.
```

tvmPmt()

Katalog >

tvmPmt(*N*,*I*,*PV*,*FV*,[*PpY*],[*CpY*],[*PmtAt*]**)**⇒*værdi*

Finansfunktion, der beregner beløbet for hver betaling.

Bemærk: Argumenter, der bruges i TVM-funktionerne er beskrevet i tabellen over TVM-argumenter, side 201. Se også **amortTbl()**, side 8.

```
tvmPmt(60,4,30000,0,12,12)           -552.496
```

tvmPV() Katalog >

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 201. 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
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 182.)

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

Kategory 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 259.

Output-variabel	**Beskrivelse**
stat.$\bar{x}$	Gennemsnit af x-værdier
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

Output-variabel	Beskrivelse
stat.Q_1X	1. kvartil af x
stat.MedianX	Median af x
stat.Q_3X	3. kvartil af x
stat.MaxX	Maksimum af x-værdier
stat.MinY	Minimum af y-værdier
stat.Q_1Y	1. kvartil af y
stat.MedY	Median af y
stat.Q_3Y	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

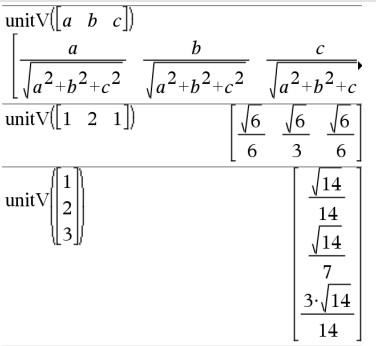
U

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.



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

unLock

Catalog >

unLock *Var1* [**,** *Var2*] [**,** *Var3*] ...

unLock *Var***.**

Oplåser de angivne variable eller variabelgrupper. Låste variable kan ikke redigeres eller slettes.

Se **Lock**, side 110 og **getLockInfo()**, side 87.

$a:=65$	65
Lock a	*Done*
getLockInfo(a)	1
$a:=75$	"Error: Variable is locked."
DelVar a	"Error: Variable is locked."
Unlock a	*Done*
$a:=75$	75
DelVar a	*Done*

V

varPop()

Katalog >

varPop(*Liste*[**,** *hyppighedsliste*]**)**⇒*udtryk*

Returnerer populationsvariansen af *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 259.

varPop$\{\{5,10,15,20,25,30\}\}$	$\frac{875}{12}$
$Ans \cdot 1.$	72.9167

varSamp()

Katalog >

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.

varSamp$\{\{a,b,c\}\}$	$\frac{a^2-a\cdot(b+c)+b^2-b\cdot c+c^2}{3}$
varSamp$\{\{1,2,5,-6,3,-2\}\}$	$\frac{31}{2}$
varSamp$\{\{1,3,5\},\{4,6,2\}\}$	$\frac{68}{33}$

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 259.

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 259.

$$\text{varSamp}\begin{bmatrix}1 & 2 & 5\\ -3 & 0 & 1\\ .5 & .7 & 3\end{bmatrix} \quad \begin{bmatrix}4.75 & 1.03 & 4\end{bmatrix}$$

$$\text{varSamp}\left(\begin{bmatrix}-1.1 & 2.2\\ 3.4 & 5.1\\ -2.3 & 4.3\end{bmatrix},\begin{bmatrix}6 & 3\\ 2 & 4\\ 5 & 1\end{bmatrix}\right) \quad \begin{bmatrix}3.91731 & 2.08411\end{bmatrix}$$

W

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 [on] nede, mens du gentagne gange trykker på [enter].
- **Windows®:** Hold tasten **F12** nede, mens du

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"

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.

warnCodes () Katalog >

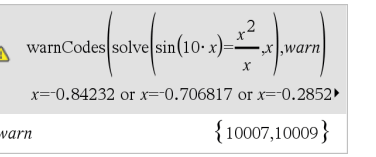
warnCodes(*Expr1***,** *StatusVar***)⇒***udtryk*

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*.

StatusVar skal være et gyldigt variabelnavn.

Se side 277 for at få en oversigt over fejlkoder og tilhørende meddelelser.



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

when() Katalog >

when(*Betingelse***,** *sandtResultat* **[,** *falskResultat***][,** *ubekendtResultat***])** **⇒***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.

when()

Katalog >

Anvend en **undef** *falskResultat* til at definere et udtryk, der kun tegner grafen til et interval.

when() er nyttig til definition af rekursive funktioner.

$\text{when}(n>0, n\cdot factoral(n-1), 1) \rightarrow factoral(n)$	*Done*
$factoral(3)$	6
$3!$	6

While

Katalog >

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 $i \le n$	
$tempsum + \frac{1}{i} \rightarrow tempsum$	
$i+1 \rightarrow i$	
EndWhile	
Return $tempsum$	
EndFunc	
	Done
$sum\_of\_recip(3)$	$\frac{11}{6}$

X

xor

Katalog >

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.

true xor true	false
5>3 xor 3>5	true

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**, side 134.

Heltal1 **xor** *Heltal2* ⇒ *heltal*

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 hexadecimale 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.

Bemærk: Se **or**, side 134.

I hexadecimal tilstand:

Vigtigt: Tallet nul, ikke bogstavet O.

0h7AC36 xor 0h3D5F	0h79169

I binær tilstand:

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() **Katalog >**

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***)**.

$\text{zeros}\left(a\cdot x^2+b\cdot x+c,x\right)$

$\left\{\frac{\sqrt{b^2-4\cdot a\cdot c}-b}{2\cdot a},\frac{-\left(\sqrt{b^2-4\cdot a\cdot c}+b\right)}{2\cdot a}\right\}$

$a\cdot x^2+b\cdot x+c|x=Ans[2]$ 0

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()**.

zeros({*Udtr1***,** *Udtr2***}, {***VarEllerGæt1***,** *VarEllerGæt2* **[,** ... **]})⇒***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.

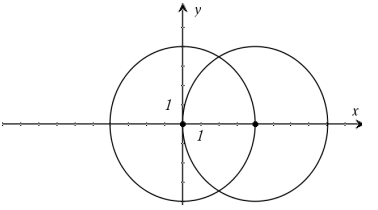
Antag for eksempel, at du har en cirkel med radius r og centrum i origo og en anden cirkel med radius r og centrum, hvor den første cirkel skærer den positive x-akse. Anvend **zeros()** til at finde skæringspunkterte.

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.

$\text{exact}\left(\text{zeros}\left(a\cdot\left(e^x+x\right)\cdot\left(\text{sign}\left(x\right)-1\right),x\right)\right)$ $\{\ \}$

$\text{exact}\left(\text{solve}\left(a\cdot\left(e^x+x\right)\cdot\left(\text{sign}\left(x\right)-1\right)=0,x\right)\right)$

$e^x+x=0$ or $x>0$ or $a=0$



$\text{zeros}\left(\left\{x^2+y^2-r^2,\left(x-r\right)^2+y^2-r^2\right\},\left\{x,y\right\}\right)$

$\begin{bmatrix} \frac{r}{2} & \frac{-\sqrt{3}\cdot r}{2} \\ \frac{r}{2} & \frac{\sqrt{3}\cdot r}{2} \end{bmatrix}$

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

Udtræk række 2:

$$Ans[2] \qquad \left[\frac{r}{2} \quad \frac{\sqrt{3}\cdot r}{2}\right]$$

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.

$$\text{zeros}\left(\left\{x^2+y^2-r^2,(x-r)^2+y^2-r^2\right\},\{x,y,z\}\right)$$
$$\begin{bmatrix} \frac{r}{2} & \frac{-\sqrt{3}\cdot r}{2} & c1 \\ \frac{r}{2} & \frac{\sqrt{3}\cdot r}{2} & c1 \end{bmatrix}$$

For polynomielle systemer afhænger beregningstiden eller hukommelsesforbruget stærkt af den rækkefølge, de ubekendte angives i. Hvis det initielle 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 **zeros()** en Gauss-eliminering i et forsøg på at bestemme alle nulpunkter.

$$\text{zeros}\left(\left\{x+e^z\cdot y-1,x-y-\sin(z)\right\},\{x,y\}\right)$$
$$\begin{bmatrix} \frac{e^z\cdot\sin(z)+1}{e^z+1} & \frac{-(\sin(z)-1)}{e^z+1} \end{bmatrix}$$

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.

$$\text{zeros}\left(\left\{e^z\cdot y-1,-y-\sin(z)\right\},\{y,z\}\right)$$
$$\begin{bmatrix} 0.041458 & 3.18306 \\ 0.001871 & 6.28131 \\ 4.76\text{E}{-11} & 1796.99 \\ 2.\text{E}{-13} & 254.469 \end{bmatrix}$$

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.

$$\text{zeros}\left(\left\{e^z\cdot y-1,-y-\sin(z)\right\},\{y,z=2\cdot\pi\}\right)$$
$$\begin{bmatrix} 0.001871 & 6.28131 \end{bmatrix}$$

zInterval

Katalog >

zInterval σ,*Liste*[,*Hyppighed*[,*CNiveau*]]

(Datalisteinput)

zInterval σ,$\bar{x}$,*n* [,*CNiveau*]

(Sammenfatning, stat input)

Beregner et *z* konfidensinterval. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 182.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

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 *Liste*

zInterval_1Prop

Katalog >

zInterval_1Prop *x*,*n* [,*CNiveau*]

Beregner én-proportion *z* konfidensinterval. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 182.)

x er et ikke-negativt heltal.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelingssandsynlighed
stat.$\hat{p}$	Den beregnede brøkdel af successer
stat.ME	Fejlmargen
stat.n	Antal stikprøver i datasekvens

zInterval_2Prop

Katalog >

zInterval_2Prop *x1*,*n1*,*x2*,*n2*[,*CNiveau*]

Beregner et to-proportion *z* konfidensinterval. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 182.)

x1 og *x2* er ikke-negative heltal.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelingssandsynlighed
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

Katalog >

zInterval_2Samp σ_1,σ_2 ,*Liste1*,*Liste2* [,*Hyppighed1*[,*Hyppighed2*,[*CNiveau*]]]

(Datalisteinput)

zInterval_2Samp σ_1,σ_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 182.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelingssandsynlighed
stat.$\bar{x}$1-$\bar{x}$2	Middelværdi stikprøver i i datasekvenserne fra den vilkårlige normalfordeling
stat.ME	Fejlmargen
stat.$\bar{x}$1, stat.$\bar{x}$2	Middelværdi stikprøver i datasekvenserne fra den vilkårlige normalfordeling
stat.σx1, stat.σx2	Stikprøve standardafvigelser for *Liste 1* og *Liste 2*
stat.n1, stat.n2	Antal stikprøver i datasekvenserne
stat.r1, stat.r2	Kendt populationsstandardafvigelse for datasekvenserne *Liste 1* og *Liste 2*

zTest

Katalog >

zTest μ*0*,σ,*Liste*,[*Hyppighed*[,*Hypot*]]

(Datalisteinput)

zTest μ*0*,σ,$\bar{x}$,*n*[,*Hypot*]

(Sammenfatning, stat input)

Udfører en *z* test med hyppigheden *Hyppighedsliste*. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 182.)

Test H_0: μ = μ0, mod en af de følgende:

Til H_a: μ < μ0, sæt *Hypot*<0

Til H_a: μ ≠ μ0 (standard), sæt *Hypot*=0

Til H_a: μ > μ0, sæt *Hypot*>0

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

Output-variabel	Beskrivelse
stat.z	($\bar{x}$ − μ0) / (σ / 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 *Liste*
stat.sx	Standardmåleafvigelse for datasekvensen. Returneres kun for *Data*input.
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 182.)

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 259.

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 182.)

x1 og *x2* er ikke-negative heltal.

zTest_2Prop

Katalog >

Test H_0: $p1 = p2$ mod en af følgende:

Til H_a: $p1 > p2$, sæt *Hypot*>0

til H_a: $p1 \neq p2$ *(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 259.

Output-variabel	Beskrivelse
stat.z	Standardnormalværdi beregnet som differens af proportioner
stat.PVal	Mindste signifikansniveau, ved hvilket 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

Katalog >

zTest_2Samp σ_1,σ_2 ,*Liste1*,*Liste2* [,*Hyppighed1*[,*Hyppighed2*[,*Hypot*]]]

(Datalisteinput)

zTest_2Samp σ_1,σ_2,$\bar{x}1$,*n1*,$\bar{x}2$,*n2*[,*Hypot*]

(Sammenfatning, stat input)

Beregner en to-prøvers *z* test. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 182.)

Test H_0: $\mu1 = \mu2$, mod af de følgende:

Til H_a: $\mu1 < \mu2$, sæt *Hypot*<0

til H_a: $\mu1 \neq \mu2$ (standard), sæt *Hypot*=0

til H_a: $\mu1 > \mu2$, *Hypot*>0

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 259.

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 *Liste 1* og *Liste 2*
stat.sx1, stat.sx2	Stikprøve standardafvigelse for datasekvenserne i *Liste 1* og *Liste 2*
stat.n1, stat.n2	Størrelse på stikprøverne

Symboler

+ (adder)

+-tast

Udtr1 **+** *Udtr2*⇒*udtryk*

Returnerer summen af de to argumenter.

56	56
56+4	60
60+4	64
64+4	68
68+4	72

Liste1 **+** *Liste2*⇒*liste*

Matrix1 **+** *Matrix2*⇒*matrix*

Returnerer en liste (eller matrix), der indeholder summerne af tilsvarende elementer i *Liste1* og *Liste2* (eller *Matrix1* og *Matrix2*).

Argumenternes dimensioner må være ens.

$\left\{22,\pi,\frac{\pi}{2}\right\}\rightarrow l1$	$\left\{22,\pi,\frac{\pi}{2}\right\}$
$\left\{10{,}5,\frac{\pi}{2}\right\}\rightarrow l2$	$\left\{10{,}5,\frac{\pi}{2}\right\}$
$l1+l2$	$\{32,\pi+5,\pi\}$
$Ans+\{\pi,-5,-\pi\}$	$\{\pi+32,\pi,0\}$
$\begin{bmatrix}a & b\\ c & d\end{bmatrix}+\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$	$\begin{bmatrix}a+1 & b\\ c & d+1\end{bmatrix}$

Udtr **+** *Liste1*⇒*liste*

Liste1 **+** *Udtr*⇒*liste*

Returnerer en liste med summerne af *Udtr* og hvert element i *Liste1*.

$15+\{10,15,20\}$	$\{25,30,35\}$
$\{10,15,20\}+15$	$\{25,30,35\}$

Udtr + *Matrix1*matrix *Matrix1* + *Udtr*⇒*matrix*

Returnerer en matrix med *Udtr* føjet til hvert element på diagonalen af *Matrix1*. *Matrix1* skal være kvadratisk.

$20+\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$	$\begin{bmatrix}21 & 2\\ 3 & 24\end{bmatrix}$

Bemærk: Anvend **.+** (punktum plus) til at addere et udtryk til hvert element.

−(subtraher)

−-tast

Udtr1−*Udtr2*⇒*udtryk*

Returnerer *Udtr1* minus *Udtr2*.

$6-2$	4
$\pi-\frac{\pi}{6}$	$\frac{5\cdot\pi}{6}$

Liste1−*Liste2*⇒*liste*

Matrix1−*Matrix2*⇒*matrix*

$\left\{22,\pi,\frac{\pi}{2}\right\}-\left\{10{,}5,\frac{\pi}{2}\right\}$	$\{12,\pi-5,0\}$
$[3\ \ 4]-[1\ \ 2]$	$[2\ \ 2]$

−(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*⇒*liste*

Liste1−*Udtr*⇒*liste*

Subtraherer hvert *Liste1* element fra *Udtr* eller subtraherer *Udtr* fra hvert *Liste1* element og returnerer en liste med resultaterne.

$15-\{10,15,20\}$ $\{5,0,-5\}$

$\{10,15,20\}-15$ $\{-5,0,5\}$

Udtr −*Matrix1*⇒*matrix*

Matrix1−*Udtr*⇒*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.

$20-\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$ $\begin{bmatrix}19 & -2\\ -3 & 16\end{bmatrix}$

Bemærk: Anvend **.**− (punktum minus) til at subtrahere et udtryk fra hvert element.

·(multiplicer) ×-tast

Udtr1 ·*Udtr2*⇒*udtryk*

Returnerer produktet af de to argumenter.

$2\cdot 3.45$ 6.9

$x\cdot y\cdot x$ $x^2\cdot y$

Liste1 ·*Liste2*⇒*liste*

Returnerer en liste med produkterne af de tilsvarende elementer i *Liste1* og *Liste2*.

Listernes dimensioner må være ens.

$\{1.,2,3\}\cdot\{4,5,6\}$ $\{4.,10,18\}$

$\left\{\frac{2}{a},\frac{3}{2}\right\}.\cdot\left\{a^2,\frac{b}{3}\right\}$ $\left\{2\cdot a,\frac{b}{2}\right\}$

Matrix1 ·*Matrix2*⇒*matrix*

Returnerer matrixproduktet af *Matrix1* og *Matrix2*.

Antallet af kolonner i *Matrix1* skal være lig med antallet af rækker i *Matrix2*.

$$\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix}.\begin{bmatrix}a & d\\ b & e\\ c & f\end{bmatrix}$$

$$\begin{bmatrix}a+2\cdot b+3\cdot c & d+2\cdot e+3\cdot f\\ 4\cdot a+5\cdot b+6\cdot c & 4\cdot d+5\cdot e+6\cdot f\end{bmatrix}$$

·(multiplicer) ×-tast

Udtr ·*Liste1*⇒*liste*

Liste1 ·*Udtr*⇒*liste*

Returnerer en liste med produkterne af *Udtr* og hvert element i *Liste1*.

$\pi \cdot \{4,5,6\} \quad \{4\cdot\pi, 5\cdot\pi, 6\cdot\pi\}$

Udtr ·*Matrix1*⇒*matrix*

Matrix1 ·*Udtr*⇒*matrix*

Returnerer en matrix med produkterne af *udtryk* og hvert element i *Matrix1*.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 0.01 \quad \begin{bmatrix} 0.01 & 0.02 \\ 0.03 & 0.04 \end{bmatrix}$

$\lambda \cdot \text{identity}(3) \quad \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}$

Bemærk: Anvend **.** ·(punktum gangetegn) til at gange et udtryk med hvert element.

/ (divider) ÷-tast

Udtr1/ *Udtr2*⇒*udtryk*

Returnerer kvotienten af *Udtr1* divideret med *Udtr2*.

$\frac{2}{3.45} \quad 0.57971$

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

Bemærk: Se også **Brøkskabelon**, side 1.

Liste1 / *Liste2*⇒*liste*

Returnerer en liste med kvotienterne af *Liste1* divideret med*Liste2*.

Listernes dimensioner må være ens.

$\frac{\{1.,2,3\}}{\{4,5,6\}} \quad \left\{0.25, \frac{2}{5}, \frac{1}{2}\right\}$

Udtr/*Liste1*⇒*liste*

Liste1 / *Udtr*⇒*liste*

Returnerer en liste med kvotienterne af *Udtr* divideret med *Liste1* eller *Liste1* divideret med *Udtr*.

$\frac{a}{\{3,a,\sqrt{a}\}} \quad \left\{\frac{a}{3}, 1, \sqrt{a}\right\}$

$\frac{\{a,b,c\}}{a\cdot b\cdot c} \quad \left\{\frac{1}{b\cdot c}, \frac{1}{a\cdot c}, \frac{1}{a\cdot b}\right\}$

Matrix1/*Udtr*⇒*matrix*

Returnerer en matrix med kvotienterne af *Matrix1*/*Udtr*.

$\frac{[a \quad b \quad c]}{a\cdot b\cdot c} \quad \left[\frac{1}{b\cdot c} \quad \frac{1}{a\cdot c} \quad \frac{1}{a\cdot b}\right]$

Bemærk: Anvend **./** (punktum divisionstegn) til at dividere et udtryk med hvert element.

^ (potens)

^-tast

Udtr1 **^** *Udtr2* ⇒ *udtryk*

Liste1 **^** *Liste2* ⇒ *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.

4^2 → 16

$\{a,2,c\}^{\{1,b,3\}}$ → $\{a,2^b,c^3\}$

Udtr **^** *Liste1*⇒*liste*

Returnerer *Udtr* opløftet til potensen af elementerne i *Liste1*.

$p^{\{a,2,-3\}}$ → $\left\{p^a,p^2,\frac{1}{p^3}\right\}$

Liste1 **^** *Udtr*⇒*liste*

Returnerer elementerne i *Liste1* opløftet til potensen i *Udtr*.

$\{1,2,3,4\}^{-2}$ → $\left\{1,\frac{1}{4},\frac{1}{9},\frac{1}{16}\right\}$

kvadratMatrix1 **^** *heltal*⇒*matrix*

Returnerer *kvadratMatrix1* opløftet til *heltals*potensen.

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.

$\begin{bmatrix}1&2\\3&4\end{bmatrix}^2$ → $\begin{bmatrix}7&10\\15&22\end{bmatrix}$

$\begin{bmatrix}1&2\\3&4\end{bmatrix}^{-1}$ → $\begin{bmatrix}-2&1\\\frac{3}{2}&\frac{-1}{2}\end{bmatrix}$

$\begin{bmatrix}1&2\\3&4\end{bmatrix}^{-2}$ → $\begin{bmatrix}\frac{11}{2}&\frac{-5}{2}\\\frac{-15}{4}&\frac{7}{4}\end{bmatrix}$

x² (kvadrat)

x²-tast

*Udtr1*²⇒*udtryk*

Returnerer kvadratet af argumentet.

*Liste1*²⇒*liste*

Returnerer en liste med kvadraterne på elementerne i *Liste1*.

*kvadratMatrix1*² ⇒*matrix*

Returnerer matrix i anden potens af *kvadratMatrix1*. Dette er ikke det samme som at beregne kvadratet på hvert element. Brug .^2 til at beregne kvadratet på hvert element.

$4^2 \quad 16$

$\{2,4,6\}^2 \quad \{4,16,36\}$

$\begin{bmatrix}2&4&6\\3&5&7\\4&6&8\end{bmatrix}^2 \quad \begin{bmatrix}40&64&88\\49&79&109\\58&94&130\end{bmatrix}$

$\begin{bmatrix}2&4&6\\3&5&7\\4&6&8\end{bmatrix}.^\wedge 2 \quad \begin{bmatrix}4&16&36\\9&25&49\\16&36&64\end{bmatrix}$

.+ (punktum plustegn)

.+-taster

Matrix1 **.+** *Matrix2* ⇒ *matrix*

Udtr **.+** *Matrix1* ⇒ *matrix*

Matrix1 **.+** *Matrix2* returnerer en matrix, der er summen af hvert par tilsvarende elementer i *Matrix1* og *Matrix2*.

Udtr **.+** *Matrix1* returnerer en matrix, der er summen af *Udtr* og hvert element i *Matrix1*.

$\begin{bmatrix}a&2\\b&3\end{bmatrix}.+\begin{bmatrix}c&4\\5&d\end{bmatrix} \quad \begin{bmatrix}a+c&6\\b+5&d+3\end{bmatrix}$

$x.+\begin{bmatrix}c&4\\5&d\end{bmatrix} \quad \begin{bmatrix}x+c&x+4\\x+5&x+d\end{bmatrix}$

.− (punktum minus.)

.−-taster

Matrix1 **.−***Matrix2* ⇒ *matrix*

Udtr **.−***Matrix1* ⇒ *matrix*

Matrix1 **.−***Matrix2* returnerer en matrix, der er differensen mellem hvert par af tilsvarende elementer i *Matrix1* og *Matrix2*.

Udtr **.−***Matrix1* returnerer en matrix, der er differensen af *Udtr* og hvert element i *Matrix1*.

$\begin{bmatrix}a&2\\b&3\end{bmatrix}.-\begin{bmatrix}c&4\\d&5\end{bmatrix} \quad \begin{bmatrix}a-c&-2\\b-d&-2\end{bmatrix}$

$x.-\begin{bmatrix}c&4\\d&5\end{bmatrix} \quad \begin{bmatrix}x-c&x-4\\x-d&x-5\end{bmatrix}$

. ·(punktum mult.) [.] [×]-taster

Matrix1 **.** · *Matrix2* ⇒ *matrix*

Udtr **.** ·*Matrix1* ⇒ *matrix*

Matrix1 **.** · *Matrix2* returnerer en matrix, der er produktet af hvert par tilsvarende elementer i *Matrix1* og *Matrix2*.

Udtr **.** · *Matrix1* returnerer en matrix med produkterne af *Udtr* og hvert element i *Matrix1*.

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} .\cdot \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} \qquad \begin{bmatrix} a\cdot c & 8 \\ 5\cdot b & 3\cdot d \end{bmatrix}$$

$$x .\cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad \begin{bmatrix} a\cdot x & b\cdot x \\ c\cdot x & d\cdot x \end{bmatrix}$$

. / (punktum divider) [.] [÷]-taster

Matrix1 **. /** *Matrix2* ⇒ *matrix*

Udtr **. /** *Matrix1* ⇒ *matrix*

Matrix1 **./** *Matrix2* returnerer en matrix, der er kvotient af hvert par af tilsvarende elementer i *Matrix1* og *Matrix2*.

Udtr **./** *Matrix1* returnerer en matrix, der er kvotienten af *Udtr* og hvert element i *Matrix1*.

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} ./\left(\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}\right) \qquad \begin{bmatrix} \frac{a}{c} & \frac{1}{2} \\ \frac{b}{5} & \frac{3}{d} \end{bmatrix}$$

$$x ./\left(\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}\right) \qquad \begin{bmatrix} \frac{x}{c} & \frac{x}{4} \\ \frac{x}{5} & \frac{x}{d} \end{bmatrix}$$

.^ (punktum potens) [.] [^]-taster

Matrix1^*Matrix2*⇒*matrix*

Udtr.^*Matrix1*⇒*matrix*

Tal.^*Matrix1*⇒*matrix*

Matrix1 **.^** *Matrix2* returnerer en matrix, hvor hvert element i *Matrix2* er eksponent til det tilsvarende element i *Matrix1*.

Udtr **.^** *Matrix1* returnerer en matrix, hvor hvert element i *Matrix1* er eksponenten til *Udtr*.

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} .\hat{}\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} \qquad \begin{bmatrix} a^c & 16 \\ b^5 & 3^d \end{bmatrix}$$

$$x .\hat{}\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} \qquad \begin{bmatrix} x^c & x^4 \\ x^5 & x^d \end{bmatrix}$$

-(neger)

(−)-tast

-Udtr1 ⇒ *udtryk*

-Liste1 ⇒ *liste*

-Matrix1 ⇒ *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· -b	a· b

I binær tilstand:

Vigtigt: Tallet nul, ikke bogstavet O

```
-0b100101
 0b1111111111111111111111111111111
```

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

% (procent)

ctrl 📖-taster

Udtr1 **%** ⇒ *udtryk*

Liste1 **%** ⇒ *liste*

Matrix1 **%** ⇒ *matrix*

Returnerer $\frac{argument}{100}$

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 ctrl enter.
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*⇒*Boolsk udtryk*

Liste1 = *Liste2*⇒ *Boolsk liste*

Matrix1 = *Matrix2*⇒ *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.

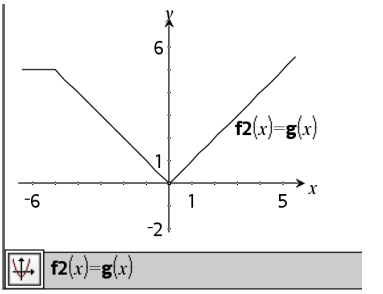
For lister og matricer returneres sammenligninger element for element.

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)

ctrl =-taster

Udtr1 ≠ *Udtr2* ⇒ *Boolsk udtryk*

Liste1 ≠ *Liste2* ⇒ *Boolsk liste*

Matrix1 ≠ *Matrix2* ⇒ *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.

Se "=" (lig med)-eksemplet.

≠ (forskellig fra)

ctrl = -taster

Bemærk: Du kan indsætte denne operator fra tastaturet ved at skrive **/=**

< (mindre end)

ctrl = taster

Udtr1 **<** *Udtr2* ⇒ *Boolsk udtryk*

Se "=" (lig med)-eksemplet.

Liste1 **<** *Liste2* ⇒ *Boolsk liste*

Matrix1 **<** *Matrix2* ⇒ *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.

≤ (mindre end eller lig med)

ctrl = taster

Udtr1 ≤ *Udtr2* ⇒ *Boolsk udtryk*

Se "=" (lig med)-eksemplet.

Liste1 ≤ *Liste2* ⇒ *Boolsk liste*

Matrix1 ≤ *Matrix2* ⇒ *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 **<=**

> (større end)

ctrl = taster

Udtr1 > *Udtr2* ⇒ *Boolsk udtryk*

Liste1 > *Liste2* ⇒ *Boolsk liste*

Matrix1 > *Matrix2* ⇒ *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.

Se "=" (lig med)-eksemplet.

≥ (større end eller lig med)

ctrl = taster

Udtr1 ≥ *Udtr2* ⇒ *Boolsk udtryk*

Liste1 ≥ *Liste2* ⇒ *Boolsk liste*

Matrix1 ≥ *Matrix2* ⇒ *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 **>=**

Se "=" (lig med)-eksemplet.

⇒ (medfører)

ctrl = -taster

BoolskUdtryk1 ⇒ *BoolskUdtryk2* returnerer *Boolsk udtryk*

BoolskListe1 ⇒ *BoolskListe2* returnerer *Boolsk liste*

BoolskMatrix1 ⇒ *BoolskMatrix2* returnerer *Boolsk matrix*

Heltal1 ⇒ *Heltal2* returnerer *Heltal*

5>3 or 3>5	true
5>3 ⇒ 3>5	false
3 or 4	7
3 ⇒ 4	-4
{1,2,3} or {3,2,1}	{3,2,3}
{1,2,3} ⇒ {3,2,1}	{-1,-1,-3}

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*

BoolskListe1 ⇔ *BoolskListe2* returnerer *Boolsk liste*

BoolskMatrix1 ⇔ *BoolskMatrix2* returnerer *Boolsk matrix*

Heltal1 ⇔ *Heltal2* returnerer *Heltal*

5>3 xor 3>5	true
5>3 ⇔ 3>5	false
3 xor 4	7
3 ⇔ 4	-8
{1,2,3} xor {3,2,1}	{2,0,2}
{1,2,3} ⇔ {3,2,1}	{-3,-1,-3}

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! ⇒ *udtryk*

Liste1! ⇒ *liste*

Matrix1! ⇒ *matrix*

Returnerer fakultetværdien af argumentet.

For en liste eller matrix returneres en liste eller matrix af elementerne.

5!	120
{5,4,3}!	{120,24,6}
$\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}!$	$\begin{bmatrix}1 & 2\\ 6 & 24\end{bmatrix}$

& (tilføj)

ctrl -taster

Streng1 & *Streng2* ⇒ *streng*

Returnerer en tekststreng, der er *Streng2* adderet til *Streng1*.

"Hello "&"Nick"	"Hello Nick"

d() (differentialkvotient)

Katalog >

Udtr1, *Var*[**,***Orden*]**)**⇒*udtryk*

d(*1***,** *Var*[**,***Orden*]**)**⇒*liste*

d(*Matrix1***,** *Var*[**,***Orden*]**)**⇒*matrix*

Returnerer differentialkvotioenten af første orden med hensyn til variablen *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

$\frac{d}{dx}(f(x)\cdot g(x))$	$\frac{d}{dx}(f(x))\cdot g(x)+\frac{d}{dx}(g(x))\cdot f(x)$
$\frac{d}{dy}\left(\frac{d}{dx}(x^2\cdot y^3)\right)$	$6\cdot y^2\cdot x$
$\frac{d}{dx}(\{x^2,x^3,x^4\})$	$\{2\cdot x,3\cdot x^2,4\cdot x^3\}$

værdi for variablen, der er bestemt af trin 1.

3. Bestemmer den symbolske differentialkvotient af resultatet af trin 2 med hensyn til variablen fra trin 1.

Hvis variablen 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.

∫(*Udtr1*, *Var*[, *Nedre*, *Øvre*]) ⇒ *udtryk*

∫(*Udtr1*, *Var*[, *Konstant*]) ⇒ *udtryk*

Returnerer integralet af *Udtr1* med hensyn til variablen *var* fra *Nedre* til *Øvre*.

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

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*.

$$\int x^2 \, dx \qquad \frac{x^3}{3}$$

$$\int(a \cdot x^2, x, c) \qquad \frac{a \cdot x^3}{3} + c$$

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.

∫() (integrer) Katalog >

∫() returnerer sig selv for dele af *Udtr1*, som ikke kan bestemmes som en eksplicit endelig kombination af dens indbyggede funktioner og operatorer.

$$\int b\cdot e^{-x^2}+\frac{a}{x^2+a^2}\,dx \quad b\cdot\int e^{-x^2}\,dx+\tan^{-1}\left(\frac{x}{a}\right)$$

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.

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 ≈.

$$\int_{-1}^{1} e^{-x^2}\,dx \qquad 1.49365$$

∫() kan indlejres og dermed klare multiple integraler. Integrationsgrænser kan afhænge af integrationsvariable uden for dem.

Bemærk: Se også **nInt()**, side 127.

$$\int_0^a\int_0^x \ln(x+y)\,dy\,dx$$

$$\frac{a^2\cdot\ln(a)}{2}+\frac{a^2\cdot(4\cdot\ln(2)-3)}{4}$$

√() (kvadratrod) ctrl x² -taster

√ (*Udtr1*)⇒*udtryk*

√ (*Liste1*)⇒*liste*

Returnerer kvadratroden af argumentet.

For en liste returneres kvadratrødderne af alle elementer i *Liste1*.

$$\sqrt{4} \qquad 2$$

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

√() (kvadratrod)

ctrl **x²** **-taster**

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

Bemærk: Se også **Kvadratrodsskabelon**, side 1.

Π() (prodSeq)

Katalog >

Π(*Udtr1*, *Var*, *Lav*, *Høj***)⇒***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 **Produktskabelon (Π)**, side 5.

Π(*Udtr1*, *Var*, *Lav*, *Lav*−1**)⇒**1

Π(*Udtr1*, *Var*, *Lav*, *Høj***) ⇒1/Π(***Udtr1*, *Var*, *Høj*+1, *Lav*−1**)** hvis *Høj* < *Lav*−1

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

$\prod_{k=1}^{n}\left(k^2\right)$ → $(n!)^2$

$\prod_{n=1}^{5}\left(\left\{\frac{1}{n},n,2\right\}\right)$ → $\left\{\frac{1}{120},120,32\right\}$

$\prod_{k=4}^{3}(k)$ → 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.

$\prod_{k=4}^{1}\left(\frac{1}{k}\right)$ → 6

$\prod_{k=4}^{1}\left(\frac{1}{k}\right)\cdot\prod_{k=2}^{4}\left(\frac{1}{k}\right)$ → $\frac{1}{4}$

Σ() (sumSeq)

Katalog >

Σ(*Udtr1*, *Var*, *Lav*, *Høj***)⇒***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.

$$\sum_{n=1}^{5}\left(\frac{1}{n}\right) \qquad \frac{137}{60}$$

$$\sum_{k=1}^{n}\left(k^2\right) \qquad \frac{n\cdot(n+1)\cdot(2\cdot n+1)}{6}$$

$$\sum_{n=1}^{\infty}\left(\frac{1}{n^2}\right) \qquad \frac{\pi^2}{6}$$

Σ(*Udtr1*, *Var*, *Lav*, *Lav*−*1***)** ⇒ 0

Σ(*Udtr1*, *Var*, *Lav*, *Høj***)** ⇒ -**Σ(***Udtr1*, *Var*, *Høj*+*1*, *Lav*−**1)** hvis *Høj* < *Lav*−*1*

$$\sum_{k=4}^{3}(k) \qquad 0$$

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.

$$\sum_{k=4}^{1}(k) \qquad -5$$

$$\sum_{k=4}^{1}(k)+\sum_{k=2}^{4}(k) \qquad 4$$

ΣInt()

Katalog >

ΣInt(*NPmt1*, *NPmt2*, *N*, *I*, *PV*,[*Pmt*], [*FV*], [*PpY*], [*CpY*], [*PmtAt*], [*afrundVærdi*]**)⇒***værdi*

ΣInt(*NPmt1*,*NPmt2*,*amortTabel***)⇒***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 201.

- Hvis du udelader *Pmt*, sættes den som

ΣInt(1,3,12,4.75,20000,,12,12) -213.48

standard til *Pmt*=**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.

ΣInt(*NPmt1,NPmt2,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 ΣPrn() nedenfor og **Bal()**, side 17.

tbl:=amortTbl(12,12,4.75,20000,,12,12)

0	0.	0.	20000.
1	-77.49	-1632.43	18367.6
2	-71.17	-1638.75	16728.8
3	-64.82	-1645.1	15083.7
4	-58.44	-1651.48	13432.2
5	-52.05	-1657.87	11774.4
6	-45.62	-1664.3	10110.1
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

ΣInt(1,3,*tbl*) -213.48

ΣPrn(*NPmt1***,** *NPmt2***,** *N***,** *I***,** *PV***,** [*Pmt*]**,** [*FV*]**,** [*PpY*]**,** [*CpY*]**,** [*PmtAt*]**,** [*afrundVærdi*]**)⇒***værdi*

ΣPrn(*NPmt1***,***NPmt2***,***amortTabel***)⇒***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 201.

- Hvis du udelader *Pmt*, sættes den som standard til *Pmt*=**tvmPmt (***N,I,PV,FV,PpY,CpY,PmtAt***)**.
- Hvis du udelader *FV*, sættes den som standard til *FV*=0.
- Standardværdierne for *PpY*, *CpY* og *PmtAt* er de samme som for TVM-funktionerne.

ΣPrn(1,3,12,4.75,20000,,12,12) -4916.28

tbl:=amortTbl(12,12,4.75,20000,,12,12)

0	0.	0.	20000.
1	-77.49	-1632.43	18367.57
2	-71.17	-1638.75	16728.82
3	-64.82	-1645.1	15083.72
4	-58.44	-1651.48	13432.24
5	-52.05	-1657.87	11774.37
6	-45.62	-1664.3	10110.07
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

ΣPrn(1,3,*tbl*) -4916.28

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)

ctrl 📖 **-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
"r"→*s1*	"r"
#*s1*	10

Returnerer værdien af variablen (r), hvis navn er lagret i variablen s1.

E (videnskabelig notation)

EE **-tast**

*mantisse***E***eksponent*

Indtaster et tal i videnskabelig notation. Tallet fortolkes som *mantisse* × 10^eksponent.

Tip: Hvis du vil indtaste en 10. potens uden at give et decimalværdiresultat, skal du bruge 10^*heltal*.

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive **@E**. Skriv for eksempel **2.3@E4** for at indtaste 2.3**E**4.

23000.	23000.
2300000000.+4.1E15	4.1E15
$3 \cdot 10^4$	30000

g (nygrader)

1-tast

*Udtr1*g⇒*udtryk*

*Udtr1*g⇒*udtryk*

*Liste1*g⇒*liste*

*Matrix1*g⇒*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 π/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**.

I vinkeltilstanden Grader, Nygrader eller Radianer:

$$\cos(50^g) \qquad \frac{\sqrt{2}}{2}$$

$$\cos(\{0,100^g,200^g\}) \qquad \{1,0,-1\}$$

r (radian)

1-tast

*Udtr1*r⇒*udtryk*

*Liste1*r⇒*liste*

*Matrix1*r⇒*matrix*

Denne funktion giver mulighed for at angive en vinkel i radianer, mens programmet er i vinkeltilstanden Grader eller Nygrader.

I vinkeltilstanden Grader ganges argumentet med 180/π.

I vinkeltilstanden Radian, returneres argument uændret.

I vinkeltilstanden Nygrader ganges argumentet med 200/π.

Tip: Anvend r, hvis du vil fremtvinge radianer i en funktionsdefinition uanset den aktuelle tilstand, når funktionen anvendes.

I vinkeltilstanden Grader, Nygrader eller Radianer:

$$\cos\left(\frac{\pi}{4^r}\right) \qquad \frac{\sqrt{2}}{2}$$

$$\cos\left(\left\{0^r,\frac{\pi}{12}^r,-(\pi)^r\right\}\right) \qquad \left\{1,\frac{\left(\sqrt{3}+1\right)\cdot\sqrt{2}}{4},-1\right\}$$

ʳ (radian)

1-tast

Bemærk: Du kan indsætte dette symbol fra computerens tastatur ved at skrive `@r`.

° (grader)

1-tast

Udtr1°⇒*udtryk*

Liste1°⇒*liste*

Matrix1°⇒*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 π/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:

$$\cos(45°) \qquad \frac{\sqrt{2}}{2}$$

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 ≈.

$$\cos\left(\left\{0,\frac{\pi}{4},90°,30.12°\right\}\right) \qquad \{1.,0.707107,0.,0.864976\}$$

°, ', " (grader/minutter/sekunder)

ctrl 📖-taster

gg°*mm*'*ss.ss*''⇒*udtryk*

*gg*Et positivt eller negativt tal

*mm*Et ikke-negativt tal

*ss.ss*Et ikke-negativt tal

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 (").

I vinkeltilstanden Grader:

25°13'17.5"	25.2215
25°30'	$\frac{51}{2}$

∠ (vinkel)

ctrl 🕮-taster

[*Radius*,∠θ_*Vinkel*]⇒*vektor*
(polært input)

[*Radius*,∠θ_*Vinkel*,*Z_Koordinat*]⇒*vektor*
(cylindrisk input)

[*Radius*,∠θ_*Vinkel*,∠θ_*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 **@<**.

I vinkeltilstanden Radian og vektorformatet indstillet til:

rektangulær

$$\left[5 \quad \angle 60° \quad \angle 45°\right] \qquad \left[\frac{5\cdot\sqrt{2}}{4} \quad \frac{5\cdot\sqrt{6}}{4} \quad \frac{5\cdot\sqrt{2}}{2}\right]$$

cylindrisk

$$\left[5 \quad \angle 60° \quad \angle 45°\right] \qquad \left[\frac{5\cdot\sqrt{2}}{2} \quad \angle\frac{\pi}{3} \quad \frac{5\cdot\sqrt{2}}{2}\right]$$

sfærisk

$$\left[5 \quad \angle 60° \quad \angle 45°\right] \qquad \left[5 \quad \angle\frac{\pi}{3} \quad \angle\frac{\pi}{4}\right]$$

(*Størrelse* ∠ *Vinkel*)⇒*kompleksVærdi*
(polært input)

Indtaster en kompleks værdi i den polære form (r∠θ). *Vinkel* fortolkes efter den aktuelle indstilling af vinkeltilstand.

I vinkeltilstanden radian og rektangulært komplekst format:

$$5+3\cdot i-\left(10\angle\frac{\pi}{4}\right) \qquad 5-5\cdot\sqrt{2}+\left(3-5\cdot\sqrt{2}\right)\cdot i$$

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 ≈.

$$5+3\cdot i-\left(10\angle\frac{\pi}{4}\right) \qquad -2.07107-4.07107\cdot i$$

' (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.

$$\text{deSolve}\left(y''=y^{\frac{-1}{2}} \text{ and } y(0)=0 \text{ and } y'(0)=0, t, y\right)$$
$$\frac{2\cdot y^{\frac{3}{4}}}{3}=t$$

_ (understregning som et tomt element)

Se |“Tomme (ugyldige) elementer” side 259, .

_ (understregning som en enhedsbenævnelse)

ctrl [␣]-taster

*Udtr*_*Enhed*

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 Catalog og vise fanen med enhedsomregninger. Du kan vælge enhederne fra Katalog eller skrive enhederne direkte.

$3\cdot \_m \blacktriangleright \_ft$ → $9.84252\cdot \_ft$

Bemærk: Du kan finde konverteringstegnet, ▸, i Katalog. Klik på [∫Σ], og klik derefter på **Matematiske operatorer**.

*Variabel*_

Når *Variabel* ikke har nogen værdi, behandles den, som om den repræsenterer et komplekst tal. Uden_ behandles variablen 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_.

Det antages, at z er udefineret:

$\text{real}(z)$	z
$\text{real}(z\_)$	$\text{real}(z\_)$
$\text{imag}(z)$	0
$\text{imag}(z\_)$	$\text{imag}(z\_)$

▸ (konverter) ctrl 📖-taster

Udtr_Enhed1 ▸ *_Enhed2*⇒*Udtr_Enhed2*

3·_m▸_ft	9.84252·_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 ctrl ␣.

Bemærk: Temperaturenhederne konverteres med **tmpCnv()** og **ΔtmpCnv()**. Konverteringsoperatoren ▸ håndterer ikke temperaturenheder.

10^() Katalog >

10^ (*Udtr1***)**⇒*udtryk*

$10^{1.5}$	31.6228

10^ (*Liste1***)**⇒*liste*

$10^{\{0,-2,2,a\}}$	$\left\{1,\frac{1}{100},100,10^{a}\right\}$

Returnerer 10 opløftet til potensen af argumentet.

I en liste returneres 10 opløftet til potensen af elementerne i *Liste1*.

10^(*kvadratMatrix1***)**⇒*kvadratmatrix*

$$10^{\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix}}$$

$$\begin{bmatrix}1.14336\text{E}7 & 8.17155\text{E}6 & 6.67589\text{E}6\\ 9.95651\text{E}6 & 7.11587\text{E}6 & 5.81342\text{E}6\\ 7.65298\text{E}6 & 5.46952\text{E}6 & 4.46845\text{E}6\end{bmatrix}$$

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*

Liste1 **^-1**⇒*liste*

Returnerer den reciprokke værdi af argumentet.

For en liste returneres de reciprokke værdier af elementerne i *Liste1*.

$(3.1)^{-1}$ → 0.322581

$\{a,4,-0.1,x,-2\}^{-1}$ → $\left\{\frac{1}{a},\frac{1}{4},-10.,\frac{1}{x},\frac{-1}{2}\right\}$

kvadratMatrix1 **^-1**⇒*kvadratMatrix*

Returnerer den inverse værdi af *kvadratMatrix1*.

kvadratMatrix1 skal være en ikke-singulær kvadratisk matrix.

$\begin{bmatrix}1&2\\3&4\end{bmatrix}^{-1}$ → $\begin{bmatrix}-2&1\\\frac{3}{2}&\frac{-1}{2}\end{bmatrix}$

$\begin{bmatrix}1&2\\a&4\end{bmatrix}^{-1}$ → $\begin{bmatrix}\frac{-2}{a-2}&\frac{1}{a-2}\\\frac{a}{2\cdot(a-2)}&\frac{-1}{2\cdot(a-2)}\end{bmatrix}$

| (betingelses-operator)

ctrl -taster

Udtryk **|** *BoolskUdtryk1* [**and***BoolskUdtryk2*]...

Udtryk **|** *BoolskUdtryk1* [**or***BoolskUdtryk2*]...

("|")-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
- Udelukkelser

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*.

$x+1|x=3$ → 4

$x+y|x=\sin(y)$ → $\sin(y)+y$

$x+y|\sin(y)=x$ → $x+y$

$x^3-2\cdot x+7\rightarrow f(x)$ → *Done*

$f(x)|x=\sqrt{3}$ → $\sqrt{3}+7$

$(\sin(x))^2+2\cdot\sin(x)-6|\sin(x)=d$ → $d^2+2\cdot d-6$

| (betingelses-operator)

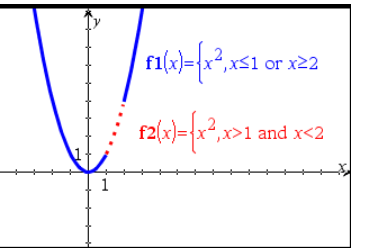
ctrl [📖]-taster

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.

$\text{solve}(x^2-1=0, x) \mid x>0 \text{ and } x<2 \quad x=1$

$\sqrt{x}\cdot\sqrt{\frac{1}{x}} \mid x>0 \quad 1$

$\sqrt{x}\cdot\sqrt{\frac{1}{x}} \quad \sqrt{\frac{1}{x}}\cdot\sqrt{x}$



Udelukkelser benytter "forskellig fra"-kommandoen (/= eller ≠) 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.

$\text{solve}(x^2-1=0, x) \mid x\neq 1 \quad x=-1$

→ (lagring)

ctrl var-tast

Udtr → *Var*

Liste → *Var*

Matrix → *Var*

Udtr → *Funktion*(*Param1*,...)

List → *Funktion*(*Param1*,...)

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*.

$\frac{\pi}{4} \rightarrow myvar \quad \frac{\pi}{4}$

$2\cdot\cos(x) \rightarrow y1(x) \quad Done$

$\{1,2,3,4\} \rightarrow lst5 \quad \{1,2,3,4\}$

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

"Hello" → *str1* "Hello"

→ (lagring)

ctrl var -tast

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)=Func
    © Declare variables
    Local i,result
    result:=0
    For i,1,n,1 ©Loop n times
    result:=result+i^2
    EndFor
    Return result
    EndFunc
                                    Done
g(3)                                14
```

0b, 0h

0 B-taster, 0 H-taster

0b *binærtTal*

0h *hexadecimaltTal*

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).

Resultatet vises i den valgte tilstand for talsystem.

I decimal tilstand:

```
0b10+0hF+10                         27
```

I binær tilstand:

```
0b10+0hF+10                    0b11011
```

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 eller PublishView (.tnsp) 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.

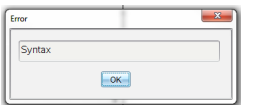
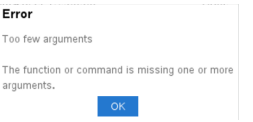
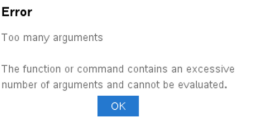
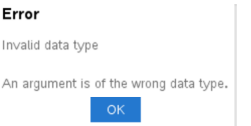
Grafikskærm	**Standard**
Højde	212
Bredde	318
Farve	hvid: 255,255,255

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 grafikskærmen

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 grafikkonteksten. 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*

Rydder hele skærmen, hvis der ikke er angivet nogen parametre.

Hvis *x*, *y*, *bredde* og *højde* er angivet, vil rektanglet der defineres af parametrene vil blive ryddet.

```
Ryd
```

Rydder hele skærmen

```
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

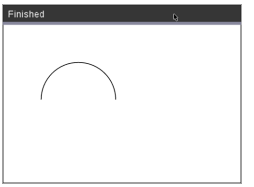
bredde, *højde*: dimensioner af et afgrænset rektangel

"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

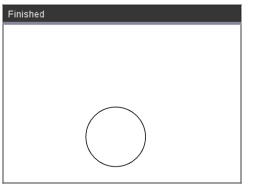
Katalog > CXII

DrawCircle *x, y, radius*

x, *y*: koordinat af centrum

radius: cirklens radius

```
DrawCircle 150,150,40
```



Se også: FillCircle

DrawLine

Katalog > CXII

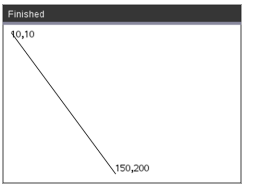
DrawLine *x1, y1, x2, y2*

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 grafikskærmen, 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*

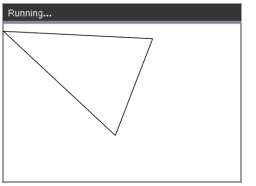
Bemærk: DrawPoly *xlist, ylist*
Form vil forbinde *x1, y1* til *x2, y2*, *x2, y2* til *x3, y3* og så videre.

Bemærk: DrawPoly *x1, y1, x2, y2, x3, y3...xn, yn*
xn, yn vil **IKKE** blive automatisk forbundet med *x1, y1*.

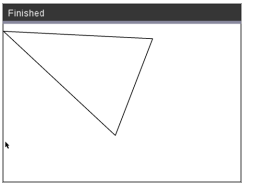
Udtryk der evalueres til en liste over reelle floats
xlist, ylist

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
```



DrawPoly

Katalog > CXII

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

Katalog > CXII

DrawRect *x, y, bredde, højde*

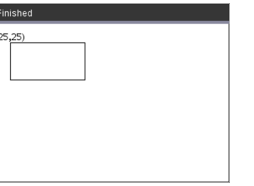
x, y: øverste venstre koordinat af rektangel

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

Katalog > CXII

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*

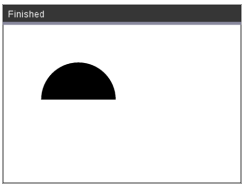
x, y: øverste venstre koordinat af afgrænsende rektangel

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

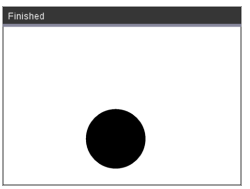
FillCircle *x, y, radius*

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

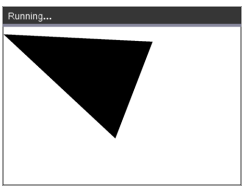
FillPoly *xlist, ylist*

or

FillPoly *x1, y1, x2, y2, x3, y3...xn, yn*

Bemærk: Linjen og farven er angivet af SetColor og SetPen

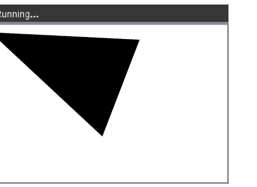
```
xlist:={0,200,150,0}
ylist:={10,20,150,10}
FillPoly xlist,ylist
```



FillPoly

Katalog > CXII

```
FillPoly 0,10,200,20,150,150,0,10
```



FillRect

Katalog > CXII

FillRect *x, y, bredde, højde*

x, y: øverste venstre koordinat af rektangel

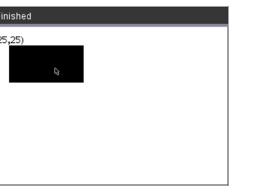
bredde, højde: bredde og højde af rektangel

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

PaintBuffer

Katalog > 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

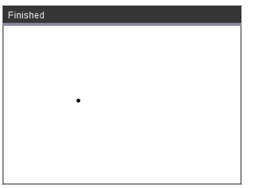
10 - større punkt, udfyldt

11 - større punkt, tomt

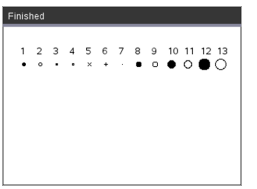
12 - største punkt, udfyldt

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

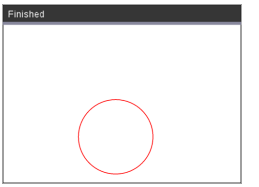
SetColor

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

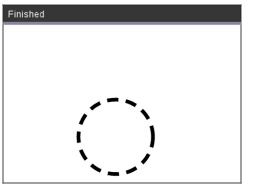
tykkelse, stil

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

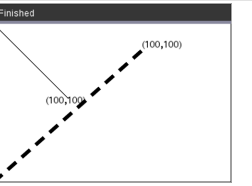
xMin, xMax, yMin, yMax

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 xmin er større end eller lig med xmax eller ymin er større end eller lig med ymax, 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 51, og **isVoid()**, side 98.

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 ctrl ␣.

Beregninger, der omfatter ugyldige elementer

De fleste beregninger med et ugyldigt input giver et ugyldigt resultat. Se de specielle tilfælde i det følgende.

$\lvert \_ \rvert$	_
$\gcd(100,\_)$	_
$3+\_$	_
$\{5,\_,10\}-\{3,6,9\}$	$\{2,\_,1\}$

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	$\{1.,\_,1.,1.,1.\}$

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	$\{\_,\_,0.,0.\}$
stat.XReg	$\{\_,\_,4.,5.\}$
stat.YReg	$\{\_,\_,5.,6.6\}$
stat.FreqReg	$\{\_,\_,1.,1.\}$

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	$\{1.,\_,1.,1.\}$

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	**pi**
θ	**theta**
∞	**uendeligt**
$\leq$	**<=**
$\geq$	**>=**
$\neq$	**/=**
$\Rightarrow$ (medfører)	**=>**
$\Leftrightarrow$ (ensbetydende med)	**<=>**
$\rightarrow$ (gem operator)	**=:**
| | (absolut værdi)	**abs(...)**
$\sqrt{}$**()**	**sqrt(...)**
$d()$	**derivative(...)**
$\int()$	**integral(...)**
Σ**()** (Sumskabelon)	**sumSeq(...)**
Π**()** (Produktskabelon)	**prodSeq(...)**
$\sin^{-1}$**()**, $\cos^{-1}$**()**, ...	**arcsin(...)**, **arccos(...)**, ...
Δ**List()**	**deltaList(...)**
Δ**tmpCnv**	**deltaTmpCnv(...)**

På computerens tastatur

Indsæt dette:	**- ved at skrive denne genvej:**
c1, *c2*, ... (konstanter)	**@c1**, **@c2**, ...
n1, *n2*, ... (heltalskonstanter)	**@n1**, **@n2**, ...
i (imaginær konstant)	@i

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, variabler 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 variablen 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→r og "r"→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 (−) efterfulgt af tallet. Postoperationer og eksponentiering udføres før negation. For eksempel er resultatet af $^-x2$ 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

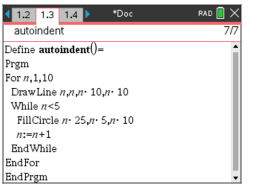
Automatisk indrykning i programmeringseditoren

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

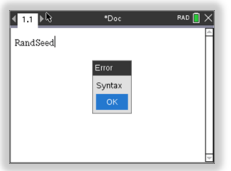
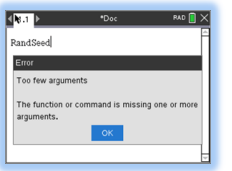
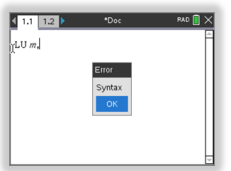
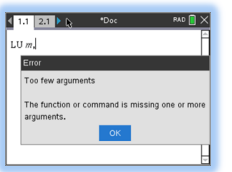
Fejlmeddelelser

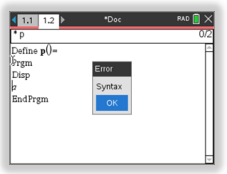
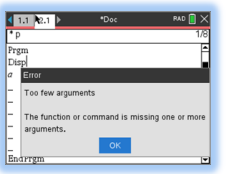
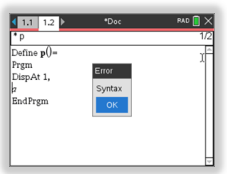
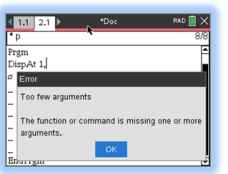
Fejltilstand	**Ny meddelelse**
Fejl i betingelseserklæring (**If/While**)	En betinget erklæring løstes ikke til **TRUE** eller **FALSE** **BEMÆRK:** Med ændringen for at placere markøren på linjen med fejlen, behøver vi ikke længere at angive, om fejlen er i en "**If**"-erklæring eller en "**While**"-erklæring.
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**

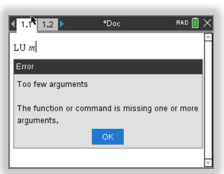
Syntaksfejl

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
	
	

Aktuel adfærd	Ny CX II-adfærd
	
	

Bemærk: Når en ufuldstændig liste over argumenter ikke følges af et komma, er fejlmeddelelsen: “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. 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	8987551787.3682 _m/_F
_Fc	Faradays konstant	96485.33289 _coul/_mol
_g	Tyngdeaccelerationen	9.80665 _m/_s^2
_Gc	Tyngdekonstant	6.67408E-11 _m^3/_kg/_s^2
_h	Plancks konstant	6.626070040E-34 _J _s
_k	Boltzmanns konstant	1.38064852E-23 _J/_°K
_μ0	Vakuumpermeabiliteten	1.2566370614359E-6 _N/_A^2
_μb	Bohr-magneton	9.274009994E-24 _J _m^2/_Wb
_Me	Elektronens hvilemasse	9.10938356E-31 _kg
_Mμ	Myonens masse	1.883531594E-28 _kg
_Mn	Neutronens hvilemasse	1.674927471E-27 _kg
_Mp	Protonens hvilemasse	1.672621898E-27 _kg
_Na	Avogadros konstant	6.022140857E23 /_mol
_q	Elementarladningen	1.6021766208E-19 _coul
_Rb	Bohr-radius	5.2917721067E-11 _m
_Rc	Gaskonstanten	8.3144598 _J/_mol/_°K
_Rdb	Rydbergs konstant	10973731.568508/_m
_Re	Elektronradius	2.8179403227E-15 _m
_u	Atommasse	1.660539040E-27 _kg
_Vm	Det molare volumen	2.2413962E-2 _m^3/_mol
_ε0	Permittivitet for et vakuum (konstant)	8.8541878176204E-12 _F/_m
_σ	Stefan-Boltzmanns konstant	5.670367E-8 _W/_m^2/_°K^4
_ϕ0	Magnetisk fluxkvantum	2.067833831E-15 _Wb

Fejlkoder 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**, side 197.

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 If a<b denne fejl, enten fordi a eller b 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 esc eller on-tasten under en længere beregning eller under en programeksekvering.
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->a, hvor a er en udefineret variabel give denne fejl.
200	Ugyldigt begrænsningsudtryk For eksempel vil solve(3x^2-4=0,x) | x<0 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 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 For eksempel er solve(3x^-4,x) ugyldig, fordi det første argument ikke er en ligning.

Fejlkode	Beskrivelse
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 Programmer kan ikke kaldes i funktioner eller udtryk som 1+p(x), hvor p er et program.
600	Ugyldig tabel

Fejlkode	Beskrivelse
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. For at kunne tillade komplekse resultater skal du ændre tilstanden Reel eller Kompleks til REKTANGULÆR eller POLÆR

Fejlkode	Beskrivelse
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 variablen. 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
1000	Talområde for Vindue variabel
1010	Zoom

Fejlkode	Beskrivelse
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 Kompleks 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 biblioteksstinavn Et stinavn skal være på formen *xxx**yyy*, hvor: • *xxx*-delen kan have 1 til 16 tegn. • *yyy*-delen kan have 1 til 15 tegn.

Fejlkode	Beskrivelse
	Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1170	Ugyldig brug af biblioteksstinavn • En værdi kan ikke tildeles et stinavn med **Define**, **:=** eller sto →. • 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: Funktionerne tangentLine og normalLine understøtter kun reelle funktioner.
1230	Domænefejl. Trigonometriske konverteringsoperatorer understøttes ikke i vinkeltilstandene Grader eller Nygrader.
1250	Argumentfejl Brug et system af lineære ligninger .

Fejlkode	Beskrivelse
	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 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.

Fejlkoder og -meddelelser

Du kan bruge funktionen **warnCodes()** til at gemme fejlkoder genereret ved beregning af et udtryk. Denne tabel viser de numeriske fejlkoder og de tilhørende meddelelser.

Se **warnCodes()**, side 206 for at få et eksempel på lagring af fejlkoder.

Fejlkode	Meddelelse
10000	Operationen kan måske indføre falske løsninger.
10001	Differentiation af en ligning kan give en falsk ligning.
10002	Tvivlsom løsning
10003	Tvivlsom præcision
10004	Operationen kan miste løsninger.
10005	cSolve kan give flere nulpunkter.
10006	Solve kan give flere nulpunkter.
10007	Der kan være flere løsninger. Forsøg at specificere relevante nedre og øvre grænser og/eller et gæt. Eksempler med brug af solve(): • solve(Ligning, Var=Gæt)|nedreGrænse<Var<øvreGrænse • solve(Ligning, Var)|nedreGrænse<Var<øvreGrænse • solve(Ligning,Var=Gæt)
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 erstattet med 1
10015	1^∞ eller 1^undef erstattet med 1
10016	1^undef erstattet med 1
10017	Overflow erstattet med ∞ eller -∞
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.

Fejlkode	Meddelelse
10022	Angivelse af korrekte nedre og øvre grænser kan give en løsning.
10023	Skalaren er multipliceret med identitetsmatricen.
10024	Resultatet er opnået vha. tilnærmet aritmetik.
10025	Ækvivalensen kan ikke verificeres i tilstanden EXACT.
10026	Begrænsningen ignoreres muligvis. Angiv begrænsningen på formen "\" 'Variabel MatTestSymbol Konstant' eller en konjunktion af disse former, f.eks. '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 vedrørende oplysninger om garantiens varighed og betingelser eller om produktservice.

Begrænset reklamationsret. Denne garanti påvirker ikke dine lovbestemte rettigheder.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243

Indeks

-

-, subtrahere 217

!

!, fakultet 228

"

", sekundnotation 236

#

#, henvisning 234
#, henvisningsoperator 264

%

%, procent 223

&

&, tilføj 228

*

*, multiplicere 218

.

.-, punktum subtraktion 221
.*, punktum multiplikation 222
./, punkt division 222
.^, punktum potens 222
.+, punktum addition 221

/

/, dividere 219

:

:=, tildel 242

^

^⁻¹, reciprok 240
^, potens 220

_

_, enhedsbenævnelse 238

|

|, betingelses-operator 240

’

’, minutnotation 236
’, mærke 238

+

+, adder 217

=

≠, forskellig fra 224
=, lig med 223

>

>, større end 226

∏

∏, produkt 231

Σ

∑( ), sum 232
∑Int( ) 232
∑Prn( ) 233

√

√, kvadratrod 230

∫

∫, integrer 229

≤

≤, mindre end eller lig med 225

≥

≥, større end eller lig med 226

▸

▸, konvertere enheder 239
▸approxFraction( ) 14
▸Base10, vise som decimalt heltal ... 19
▸Base16, vise som hexadecimalt 20
▸Base2, vise som binær 18
▸cos, vises i termer af cosinus 31
▸Cylind, vise som cylindrisk vektor .. 44
▸DD, vise som decimal vinkel 47
▸Decimal, vise resultat som decimal . 47
▸DMS, vise som
grader/minutter/sekunder 57
▸exp, viser i termer af e 66
▸Grad, konvertere til vinkelmål i
nygrader 90
▸Polar, vise som polær vektor 137
▸Rad, konverter til vinkelmål i
radianer 148
▸Rect, vis som rektangulær vektor .. 151
▸sin, vis udtryk i termer af sinus 172
▸Sphere, vise som en kuglevektor .. 180
▸tmpCnv() 195

→

→, lagring 241

⇒

⇒, medfører[*] 227, 261

⇔

⇔, ensbetydende med[*] 227

©

©, kommentar 243

°

°, grader/minutter/sekunder 236
°, gradnotation 236

0

0b, binær indikator 243
0h, hexadecimal indikator 243

1

10^( ), tiende potens 239

A

abs( ), absolut værdi 8
Absolut værdi
skabelon til 3-4
adder, + ... 217
afrund, round( ) 160
afslut løkke, EndLoop 114
afslut, Exit 66
afslutte funktion, EndFunc 80
amortiseringstabel, amortTbl( ) 8, 17
amortTbl( ), amortiseringstabel 8, 17
and, Boolsk operator 9
anden afledede
skabelon til 6
andengradsregression, QuadReg ... 145
angle( ), vinkel 10
Anmodning 154
ANOVA, envejs variansanalyse 11
ANOVA2way, to-vejs variansanalyse 11
ans, sidste resultat 13
approx( ), tilnærmet 14
approxRational( ) 14
arccos() 15
arccosh() 15
arccot() 15
arccoth() 15
arccsc() 15
arccsch() 15
arcLen( ), buemål 15
arcsec() 15
arcsech() 16
arcsin() 16
arcsinh() 16
arctan() 16
arctanh() 16

arcuscosinus, $\cos^{-1}()$ 33
arcussinus, $\sin^{-1}()$ 173
arcustangens, $\tan^{-1}()$ 188
argumenter i TVM-funktioner 201
augment( ), udvid/sammenkæd 16
avgRC( ), gennemsnitlig ændringshastighed 17

B

beregne polynomium, polyEval( ) ... 139
beregning, rækkefølge af 263
Bestemt integral
  skabelon til 6
betingelses-operator "|" 240
betingelses-operator, rækkefølge for en beregning 263
Bibliotek (Library)
  danner genveje til objekter 100
binomCdf( ) 21, 96
binomPdf( ) 21
binær
  indikator, 0b 243
  vise, ▸Base2 18
Boolske operatorer
  ⇒ 227, 261
  ⇔ 227
  and 9
  nand 124
  nor 128
  not 130
  or 134
  xor 207
brugerdefinerede funktioner 48
brugerdefinerede funktioner og programmer 49
brøker
  skabelon til 1
  ægteBrøk 143
buemål, arcLen( ) 15

C

Cdf( ) 72
ceiling( ), rund op 21
centralDiff( ) 22
cFactor( ), kompleks faktor 22
char( ), tegnstreng 24
charPoly( ) 24
$\chi^2$2way 24
ClearAZ 26
ClrErr, slet fejl 26
colAugment 27
colDim( ), matrix kolonnedimension 27
colNorm( ), matrix kolonnenorm ... 27
comDenom( ), fællesnævner 28
completeSquare( ), complete square 29
conj( ), komplekst konjugat 29
constructMat( ), konstruer matrix . 30
corrMat( ), korrelationsmatrix 31
$\cos^{-1}$, arcuscosinus 33
cos( ), cosinus 32
$\cosh^{-1}()$, hyperbolsk arcuscosinus .. 34
cosh( ), hyperbolsk cosinus 34
cosinus
  viser udtryk ud fra 31
cosinus, cos( ) 32
$\cot^{-1}()$, arcuscotangens 35
cot( ), cotangens 35
cotangens, cot( ) 35
$\coth^{-1}()$, hyperbolsk arcuscotangens 36
coth( ), hyperbolsk cotangens 36
count( ), tælle elementer i en liste ... 36
countif( ), tælle elementer i en liste betinget 37
cPolyRoots() 38
crossP( ), vektorprodukt 38
$\csc^{-1}()$, invers cosecans 39
csc( ), cosecans 38
$\operatorname{csch}^{-1}()$, invers hyperbolsk cosecans 39
csch( ), hyperbolsk cosecans 39
cSolve( ), kompleks løse 40
CubicReg, tredjegradsregression ... 42
cumulative sum, cumulativeSum( ) . 43
cumulativeSum( ), cumulative sum . 43
Cycle, cyklus 44
cyklus, Cycle 44
cylindrisk vektorvisning, ▸Cylind 44

cZeros( ), komplekse nulpunkter ... 45

D

d( ), første differentialkvotient 228
dage mellem datoer, dbd( ) 46
dbd( ), dage mellem datoer 46
decimal
heltal vise, ▸Base10 19
vinkelvisning, ▸DD 47
Define 48
Define LibPriv 49
Define LibPub 49
Define, definere 48
definere
offentlig funktion eller program 49
privat funktion eller program .. 49
definere, Define 48
definitionsmængde-kommando, domain( ) 58
delmatrix, subMat( ) 185, 187
deltaList() 50
deltaTmpCnv() 50
DelVar, slette variabel 50
delVoid( ), fjerner ugyldige elementer 51
derivative() 51
deSolve( ), løsning 51
det( ), matrixdeterminant 54
diag( ), matrixdiagonal 54
differentialkvotient eller differentialkvotienten af nte orden
skabelon til 6
differentialkvotienter
første differentialkvotient, d( ) . 228
numerisk differentialkvotient, nDeriv( ) 126-127
numerisk differentialkvotient, nDerivative( ) 125
dim( ), dimension 54
dimension, dim( ) 54
Disp, vis data 55, 163
DispAt 55
distributionsfunktioner
binomCdf( ) 21
binomPdf( ) 21
normCdf( ) 129
normPdf( ) 129
poissCdf( ) 137
poissPdf( ) 137
tCdf( ) 191
tPdf( ) 196
$\chi^2$2way( ) 24
χ^2Cdf( ) 25
χ^2GOF( ) 25
χ^2Pdf( ) 26
dividere, P 219
division af heltal, intDiv( ) 94
domain( ), definitionsmængde-kommando 58
dominantTerm( ), dominerende led 59
dominerende led, dominantTerm( ) 59
dot
produkt, dotP( ) 60
dotP( ), prikprodukt 60

E

e eksponent
skabelon til 2
e til en potens, e^( ) 60, 67
E, eksponent 234
e, viser udtryk i termer af 66
e^( ), e til en potens 60
eff( ), omregn nominel til effektiv rente 61
effektiv rente, eff( ) 61
egenvektor, eigVc( ) 61
egenværdi, eigVl( ) 62
eigVc( ), egenvektor 61
eigVl( ), egenværdi 62
eksakt, exact( ) 66
eksponent, E 234
eksponenter
skabelon til 1
eksponentiel regression, ExpReg ... 69
ellers hvis, ElseIf 62
ellers, Else 91
ElseIf, ellers hvis 62

end
for, EndFor ... 76
løkke...EndLoop ... 114
EndWhile, slut while-kompleks ... 207
enheder
konvertere ... 239
enhedsvektor, unitV( ) ... 203
ensbetydende med, ⇔ ... 227
EOS ligningsoperativsystem (Equation Operating System) ... 263
Equation Operating System (EOS) ligningsoperativsystem ... 263
etiket, Lbl ... 99
euler( ), Euler function ... 63
exact( ), eksakt ... 66
Exit, afslut ... 66
exp( ), e til en potens ... 67
exp▸list( ), udtryk til liste ... 67
expand( ), ledform ... 68
expr( ), streng til udtryk ... 69, 111
ExpReg, eksponentiel regression ... 69

F

F-test- med 2 målinger ... 79
factor( ), opløse i faktorer ... 70
fakultet, ! ... 228
fejlkoder og -meddelelser ... 277
fejlmeddelelser og fejlfinding
slette fejl, ClrErr ... 26
videresende fejl, PassErr ... 136
Fill, matrixudfyldning ... 73
finansfunktioner, tvmFV( ) ... 199
finansfunktioner, tvmI( ) ... 200
finansfunktioner, tvmN( ) ... 200
finansfunktioner, tvmPmt( ) ... 200
finansfunktioner, tvmPV( ) ... 201
FiveNumSammendrag ... 73
fjerdegradsregression, QuartReg ... 146
floor( ), runde ned ... 74
flyt, shift( ) ... 169
fMax( ), funktionsmaksimum ... 74
fMin( ), funktionsminimum ... 75
For ... 76
for, For ... 76
For, for ... 76
fordelingsfunktioner
binomCdf( ) ... 96
invNorm( ) ... 97
invt( ) ... 97
Invχ²( ) ... 95
format( ), formatere streng ... 76
formatere streng, format( ) ... 76
forskellig fra, ≠ ... 224
fortegn, sign( ) ... 170
fpart( ), funktionsdel ... 77
freqTable( ) ... 77
frequency( ) ... 78
Frobenius-norm, norm( ) ... 129
Func, funktion ... 80
Func, programfunktion ... 80
funktioner
brugerdefinerede ... 48
del, fpart( ) ... 77
maksimum, fMax( ) ... 74
minimum, fMin( ) ... 75
programfunktion, Func ... 80
funktioner og variable
kopiere ... 30
fyld ... 251-252
fællesnævner, comDenom( ) ... 28
første orden
skabelon til ... 5

G

g, nygrader ... 235
gcd( ), største fælles divisor ... 80
gennemsnitlig ændringshastighed, avgRC( ) ... 17
genveje, tastatur ... 261
geomCdf( ) ... 81
geomPdf( ) ... 81
Get ... 81, 253
getDenom( ), hente/returnere nævner ... 83
getKey() ... 83

getLangInfo( ), hent/returner sproginformation 86
getLockInfo( ), tester låsestatus af en variabel eller variabelgruppe 87
getMode( ), hente tilstandsindstillinger 87
getNum( ), hente/returnere tal 88
GetStr 88
getType( ), get type of variable 89
getVarInfo( ), hent/returner variabeloplysninger 89
Goto, gå til 90
grader/minutter/sekunder-notation 236
gradnotation, g 236
grupper, låse og låse op 110, 204
grupper, teste låsestatus 87
grænseværdi
skabelon til 7
gå til, Goto 90

H

heltal, int( ) 94
heltalsdel, iPart( ) 97
hente/returnere
nævner, getDenom( ) 83
tal, getNum( ) 88
variabeloplysninger, getVarInfo( ) 86, 89
henvisning, # 234
henvisningsoperator (#) 264
hexadecimal
indikator, 0h 243
vise, ▸Base16 20
hvis, If 91
hvisFn( ) 92
hyperbolsk
arctangens, tanh⁻¹( ) 190
arcuscosinus, cosh⁻¹( ) 34
arcussinus, sinh⁻¹( ) 174
cosinus, cosh( ) 34
sinus, sinh( ) 174
tangens, tanh( ) 189
højre, right( ) 95, 157

I

i streng, inString( ) 94
identitetsmatrix, identity( ) 91
identity( ), identitetsmatrix 91
If, hvis 91
imag( ), imaginær del 93
imaginær del, imag( ) 93
ImpDif( ), implicit differentialkvotient 93
implicit differentialkvotient, Impdif( ) 93
indstille
tilstand, setMode( ) 167
indstillinger, hente aktuelle 87
input, Input 93
Input, input 93
inString( ), i streng 94
int( ), heltal 94
intDiv( ), division af heltal 94
integrere, ∫ 229
interpolate( ), interpolation 95
invers kumuleret normalfordeling (invNorm( ) 97
invers, ^⁻¹ 240
invF( ) 95
invNorm( ), invers kumuleret normalfordeling) 97
invt( ) 97
Invχ²( ) 95
iPart( ), heltalsdel 97
irr( ), intern rente
intern rente, irr( ) 97
isPrime( ), primtalstest 98
isVoid( ), test for ugyldighed 98

K

kombinationer, nCr( ) 125
kommentar, © 243
kompleks
faktor, cFactor( ) 22
løse, cSolve( ) 40
nulpunkter, cZeros( ) 45
komplekst
konjugat, conj( ) 29

konstant
i solve( ) 177
konstanter
genveje til 261
i cSolve( ) 42
i cZeros( ) 46
i deSolve( ) 52
i solve( ) 179
Konstruer matrix, constructMat ( ) . 30
konverter
▸Rad 148
konvertere
4Grad 90
enheder 239
kopiere variabel eller funktion,
CopyVar 30
korrelationsmatrix, corrMat( ) 31
kuglevektor vise, ▸Sphere 180
kvadratrod
skabelon til 1
kvadratrod, √( ) 181, 230

L

lagre
symbol, & 241-242
Lbl, etiket 99
lcm, mindste fælles multiplum 99
ledform, expand( ) 68
left( ), venstre 99
LibPriv 49
LibPub 49
libShortcut( ), danner genveje til
biblioteksobjekter 100
lig med, = ... 223
ligningssystem (2-ligninger)
skabelon til 3
ligningssystem (N-ligning)
skabelon til 3
limit
lim( ) 101
limit( ) 101
limit( ) eller lim( ), limit 101
Lineær reg t-test 105
lineær regression, LinRegAx 103
lineær regression, LinRegBx 102, 104
LinRegBx, lineær regression 102
LinRegMx, lineær regression 103
LinRegtIntervals, lineær regression . 104
linSolve() 106
list▸mat( ), liste til matrix 107
liste til matrix, liste▸mat( ) 107
liste, tælle elementer betinget 37
liste, tælle elementer i 36
lister
cumulative sum,
cumulativeSum( ) 43
differens, @list( ) 107
differenser i en liste, @list( ) 107
liste til matrix, list▸mat( ) 107
maksimum, max( ) 115
matrix til liste, mat▸list( ) 115
midt-streng, mid( ) 118
minimum, min( ) 119
ny, newList( ) 126
prikprodukt, dotP( ) 60
produkt, product( ) 143
sortere faldende, SortD 180
sortere stigende, SortA 179
summation, sum( ) 186
tomme elementer i 259
udtryk til liste, exp▸list( ) 67
udvid/sammenkæd, augment( ) 16
vektorprodukt, crossP( ) 38
ln( ), naturlig logaritme 108
LnReg, logaritmisk regression 108
Local, lokal variabel 110
Lock, låse variabel eller
variabelgruppe 110
Log
skabelon til 2
logaritmer 108
logaritmisk regression, LnReg 108
Logistik, logistisk regression 112
LogistikD, logistisk regression 113
logistisk regression, Logistik 112
logistisk regression, LogistikD 113
lokal variabel, Local 110

lokal, Local 110
Loop, løkke 114
LU, matrix opløsning i øvre-nedre .. 114
længde af streng 54
løkke, Loop 114
løs, solve( ) 176
løsning, deSolve( ) 51
låse variable og variabelgrupper 110

M

maksimum, max( ) 115
mat▸list( ), matrix til liste 115
matricer
cumulative sum, cumulativeSum( ) 43
delmatrix, subMat( ) 185, 187
determinant, det( ) 54
diagonal, diag( ) 54
dimension, dim( ) 54
egenvektor, eigVc( ) 61
egenværdi, eigVl( ) 62
identitet, identity( ) 91
kolonnedimension, colDim( ) .. 27
kolonnenorm, colNorm( ) 27
liste til matrix, list▸mat( ) 107
maksimum, max( ) 115
matrix til liste, mat▸list( ) 115
minimum, min( ) 119
multiplikation og addition af rækker, mRowAdd( ) .. 121
ny, newMat( ) 126
opløsning i øvre-nedre, LU 114
produkt, product( ) 143
punktum addition, .+ 221
punktum division, ./ 222
punktum multiplikation, .* 222
punktum potens, .^ 222
punktum subtraktion, .- 221
QR opløse i faktorer, QR 144
reduceret række-echelonform, rref( ) 161
rækkeoperation, mRow( ) 120
summation, sum( ) 186
transponere, T 187
udfylde, Fill 73
udvid/sammenkæd, augment( ) 16
vilkårlige, randMat( ) 149
matrix (1 × 2)
skabelon til 4
matrix (2 × 1)
skabelon til 4
matrix (2 × 2)
skabelon til 4
matrix (m × n)
skabelon til 4
matrix til liste, mat▸list( ) 115
matrixer
række-echelonform, ref( ) 151
rækkeaddition, rowAdd( ) 160
rækkedimension, rowDim( ) ... 160
rækkenorm, rowNorm( ) 161
rækkeskift, rowSwap( ) 161
max( ), maksimum 115
mean( ), middel 116
med, | 240
medfører, ⇒ 227, 261
median( ), median 116
median, median( ) 116
medium-medium linjeregression, MedMed 117
MedMed, medium-medium linjeregression 117
mens, While 207
mid( ), midt-streng 118
middel, mean( ) 116
midt-streng, mid( ) 118
min( ), minimum 119
mindre end eller lig med, { 225
mindste fælles multiplum, lcm 99
minimum, min( ) 119
minutnotation, ... 236
mirr( ), modificeret intern forrentning 119
mod( ), modulo 120
modificeret intern forrentning, mirr( ) 119
modulo, mod( ) 120

mRow( ), matrixrækkeoperation ... 120
mRowAdd( ), multiplikation og addition af matrixrækker .. 121
Multipel lineær regression ttest 122
multiplicere, * 218
MultReg 121
MultRegIntervals( ) 121
MultRegTests( ) 122
mærke, ' 238

N

nand, Boolsk operator 124
naturlig logaritme, ln( ) 108
nCr( ), kombinationer 125
nDerivative( ), numerisk differentialkvotient 125
negation, indtaste negative tal 264
nettonutidsværdi, npv( ) 131
newList( ), ny liste 126
newMat( ), ny matrix 126
nfMax( ), numerisk funktionsmaksimum 126
nfMin( ), numerisk funktionsminimum 127
nInt( ), numerisk integral 127
nom), omregn effektiv til nominel rente 128
nominel rente, nom( ) 128
nor, Boolsk operator 128
norm( ), Frobenius-norm 129
normal linje, normalLine( ) 129
normalLine( ) 129
normalsandsynlighedsfordeling, normCdf( ) 129
normCdf( ) 129
normPdf( ) 129
not, Boolsk operator 130
nPr( ), permutationer 130
npv( ), nettonutidsværdi 131
nSolve( ), numerisk løsning 131
nte rod
skabelon til 1
nulpunkter, zeroes( ) 208
numerisk
diffentialkvotient, nDerivative( ) 125
differentialkvotient, nDeriv( ) .. 126-127
integral, nInt( ) 127
løsning, nSolve( ) 131
ny
liste, newList( ) 126
matrix, newMat( ) 126
nygrader notation, g 235
nævner 28
når, when( ) 206

O

objekter
danner genveje til bibliotek 100
OneVar, statistik med en variabel ... 132
operatorer
rækkefølge af beregning 263
opløse i faktorer, factor( ) 70
oplåsning af variable og variabelgrupper 204
or (Boolsk), or 134
or, Boolske operatorer 134
ord( ), numerisk tegnkode 135

P

P▸Rx( ), rektangulært x-koordinat .. 135
P▸Ry( ), rektangulært y-koordinat .. 135
PassErr, videresende fejl 136
Pdf( ) 77
permutationer, nPr( ) 130
piecewise( ) 136
poissCdf( ) 137
poissPdf( ) 137
polyCoef( ) 138
polyDegree( ) 139
polyEval( ), beregne polynomium ... 139
polyGcd( ) 139-140
polynomier
beregne, polyEval( ) 139
vilkårlige, randPoly( ) 150
PolyRoots() 141

polær
koordinat, R▸Pr( ) 148
koordinat, R▸Pθ( ) 147
vektorvisning, ▸Polar 137
potens, ^ 220
PotensReg, potensregression 141
potensregression,
PowerReg 141, 154-155, 192
Prgm, definere program 142
primtalstest, isPrime( ) 98
procent, % 223
prodSeq() 143
product( ), produkt 143
produkt ∏( )
skabelon til 5
produkt, ∏( ) 231
produkt, product( ) 143
programmer
definere privat bibliotek 49
programmer og programmering
slette fejl, ClrErr 26
vis I/O-skærm, Disp 55
vis I/O-skærm, disp. 163
programmering
definere program, Prgm 142
videresende fejl, PassErr 136
vis data, Disp 55, 163
propFrac, ægte brøk 143
punktum
addition, .+ 221
division, ./ 222
multiplikation, .* 222
potens, .^ 222
subtraktion, .- 221

Q

QR opløse i faktorer, QR 144
QuadReg, andengradsregression ... 145
QuartReg, fjerdegradsregression ... 146

R

R, radian 235
R▸Pr( ), polær koordinat 148
R▸Pθ( ), polær koordinat 147
radian, R 235
rand( ), vilkårligt tal 148
randBin, vilkårligt tal 148
randInt( ), vilkårligt heltal 149
randMat( ), vilkårlig matrix 149
randNorm( ), vilkårlig norm 149
randPoly( ), vilkårligt polynomium .. 150
randSamp( ) 150
RandSeed, vilkårligt tal-seed 150
real( ), reel 150
real, reel( ) 150
reciprok, ^-1 240
reduceret række-echelonform, rref( ) 161
ref( ), række-echelonform 151
RefreshProbeVars 153
regressioner
andengrads, QuadReg 145
eksponentiel, ExpReg 69
fjerdegrads, QuartReg 146
lineær regression, LinRegAx 103
lineær regression, LinRegBx 102
logistiske, Logistik 113
potensregression,
PowerReg ..141, 154-155, 192
sinus, SinReg 175
tredjegrads, CubicReg, CubicReg 42
Regressioner
lineær regression, LinRegBx 104
logaritmisk, LnReg 108
Logistisk 112
medium-medium linje, MedMed 117
MultReg 121
rektangulær vektorvisning, ▸Rect .. 151
rektangulært x-koordinat, P▸Rx( ) .. 135
rektangulært y-koordinat, P▸Ry( ) .. 135
remain( ), rest 154
RequestStr 155
rest, remain( ) 154
resultat
vis i termer af cosinus 31
viser i termer af e 66
viser ud fra sinus 172

resultat (sidste), ans 13
resultater, statistiske 182
resultatværdier, statistik 183
Return, returner 156
returner, Return 156
right, right( ) 29, 63, 206
rk23( ), Runge Kutta-funktion 157
rotate( ), roter 158-159
roter, rotate( ) 158-159
round( ), afrund 160
rowAdd( ), addition af matrixrækker 160
rowDim( ), matrixrækkedimension 160
rowNorm( ), matrixrækkenorm 161
rowSwap( ), matrixrækkeskift 161
rref( ), reduceret række-echelonform 161
rund op, ceiling( ) 21
runde ned, floor( ) 74
runde op, ceiling( ) 22, 38
Ryd 247
række-echelonform, ref( ) 151
række, series( ) 166

S

sammenhørende ligninger, simult( ) 171
sandsynlighedsdensitivitet, normPdf ( ) 129
sec^{-1}( ), invers sekans 162
sec( ), sekans 162
$sech^{-1}$( ), invers hyperbolsk sekans 163
sech( ), hyperbolsk sekans 162
sekundnotation, " 236
sekvens, seq( ) 164
seq( ), sekvens 164
seqGen( ) 164
seqn( ) 165
sequence, seq( ) 164-165
series( ), række 166
setMode( ), indstil tilstand 167
shift( ), flyt 169
sign( ), fortegn 170
simult( ), sammenhørende ligninger 171
sin^{-1}( ), arcussinus 173
sin( ), sinus 172
$sinh^{-1}$( ), hyperbolsk arcussinus 174
sinh( ), hyperbolsk sinus 174
SinReg, sinusregression 175
sinus
  viser udtryk ud fra 172
sinus, sin( ) 172
sinusregression, SinReg 175
skabeloner
  Absolut værdi 3-4
  anden afledede 6
  bestemt integral 6
  brøk 1
  differentialkvotient eller differentialkvotienten af nte orden 6
  e eksponent 2
  eksponent 1
  første orden 5
  grænseværdi 7
  kvadratrod 1
  ligningssystem (2-ligninger) 3
  ligningssystem (N-ligning) 3
  Log 2
  matrix (1 × 2) 4
  matrix (2 × 1) 4
  matrix (2 × 2) 4
  matrix (m × n) 4
  nte rod 1
  produkt ∏( ) 5
  stykkevis-skabelon (N stykker) 3
  stykkevis funktion (2 stykker) 2
  sum ∑( ) 5
  ubestemt integral 6
slet
  fejl, ClrErr 26
slette
  ugyldige elementer fra liste 51
  variabel, DelVar 50
slut
  funktion, EndFunc 80
  hvis, EndIf 91
  mens, EndWhile 207
slut på hvis, EndIf 91

slut while-kompleks, EndWhile 207
solve( ), løs 176
som af rentebetalinger 232
SortA, sorter stigende 179
SortD, sortere faldende 180
sortere
faldende, SortD 180
stigende, SortA 179
sprog
hent sproginformation 86
sqrt( ), kvadratrod 181
standardafvigelse, stdDev( ) ...183-184, 204
stat.results 182
stat.values 183
statistik
fakultet, ! 228
kombinationer, nCr( ) 125
median, median( ) 116
middel, mean( ) 116
permutationer, nPr( ) 130
standardafvigelse, stdDev( )183-184, 204
statistik med en variabel, OneVar 132
to-variabel-resultater, TwoVar . 201
varians, variance( ) 204
vilkårlig norm, randNorm( ) 149
vilkårligt tal-seed, RandSeed ... 150
statistik med en variabel, OneVar ... 132
stdDevPop( ), population
standardafvigelse 183
stdDevSamp( ), standardafvigelse for
måling 184
Stop-kommando 185
streng
dimension, dim( ) 54
længde 54
strenge
anvende til at skabe
variabelnavne 264
flyt, shift( ) 169
formatere 76
formatere, format( ) 76
henvisning, # 234
højre, right( ) 95, 157
i, inString 94
midt-streng, mid( ) 118
roter, rotate( ) 158-159
streng til udtryk, expr( ) 69, 111
tegnkode, ord( ) 135
tegnstreng, char( ) 24
tilføj, & 228
udtryk til streng, string( ) 185
venstre, left( ) 99
string( ), udtryk til streng 185
strings
right, right( ) 29, 63, 206
student-t sandsynlighedsfordeling,
tCdf( ) 191
student-t sandsynlighedstæthed,
tPdf( ) 196
stykkevis-skabelon (N stykker)
skabelon til 3
stykkevis funktion (2 stykker)
skabelon til 2
større end eller lig med, | 226
større end, > 226
største fælles divisor, gcd( ) 80
subMat( ), delmatrix 185, 187
substitution med "|"-operator 240
subtrahere, - 217
sum ∑( )
skabelon til 5
sum af hovedstolsbetalinger 233
sum( ), summation 186
sum, S( ) 232
sumIf( ) 186
summation, sum( ) 186
sumSeq() 187

T

T (transponere) 187
t test, tTest 198
$\tan^{-1}$( ), arcustangens 188
tan( ), tangens 188
tangens, tan( ) 188
tangent linje, tangentLine( ) 189
tangentLine( ) 189
$\tanh^{-1}$( ), hyperbolsk arcustangens . 190

tanh( ), hyperbolsk tangens 189
tastaturgenveje 261
Taylor-polynomium, taylor( ) 191
taylor( ), Taylor-polynomium 191
tCdf(), student t
sandsynlighedsfordeling ... 191
tCollect( ), trigonometrisk opsamling 191
tegn 248-250
numerisk kode, ord( ) 135
streng, char( ) 24
tegnstreng, char( ) 24
test for ugyldighed, isVoid( ) 98
Test_2S, F test med 2 målinger 79
tExpand( ), trigonometrisk
ekspansion 192
Text-kommando 192
tidsdiskonterede pengeværdier,
aktuel værdi 201
tidsdiskonterede pengeværdier,
antal betalinger 200
tidsdiskonterede pengeværdier,
betalingsbeløb 200
tidsdiskonterede pengeværdier,
Fremtidig værdi 199
tidsdiskonterede pengeværdier,
Rente 200
tiende potens, 10^( ) 239
tilføj, & 228
tilnærmet, approx( ) 14
tilstande
indstille, setMode( ) 167
tilstandsindstillinger, getMode( ) ... 87
tInterval, t konfidensinterval 193
tInterval_2Samp, tomålingers t
konfidensinterval 194
tmpCnv() 195
to-variabel-resultater, TwoVar 201
tomme (ugyldige) elementer 259
tPdf( ), studentt
sandsynlighedstæthed 196
transponere, T 187
tredjegradsregression, CubicReg ... 42
trigonometrisk ekspansion, tExpand(
) 192
trigonometrisk opsamling, tCollect( ) 191
Try, fejlhåndteringskommando 197
tTest, t test 198
tTest_2Samp, to-målingers t test ... 198
TVM-argumenter 201
tvmFV( ) 199
tvmI( ) 200
tvmN( ) 200
tvmPmt( ) 200
tvmPV( ) 201
TwoVar, to-variabel-resultater 201
tælle dage mellem datoer, dbd( ) ... 46
tælle elementer i en liste betinget,
countif( ) 37
tælle elementer i en liste, count( ) ... 36

U

ubestemt integral
skabelon til 6
udelukkelse med "|"-operator 240
udtryk
streng til udtryk, expr( ) 69, 111
udtryk til liste, exp▸list( ) 67
udvid/sammenkæd, augment( ) 16
ugyldige elementer 259
ugyldige elementer, fjerne 51
ugyldighed, test for 98
understregning, _ 238
unitV( ), enhedsvektor 203
unLock, oplåse variabel eller variabel
gruppe 204

V

variabel
oprette navn fra en tegnstreng . 264
slette alle enkeltbogstaver 26
variable
lokal, Local 110
slette, DelVar 50
variable og funktioner
kopiere 30
variable, låse og låse op 87, 110, 204
varians, variance( ) 204
varPop( ) 204

varSamp( ), målingsvarians 204
vektorer
cylindrisk vektorvisning, ▸Cylind 44
enhed, unitV( ) 203
prikprodukt, dotP( ) 60
vektorprodukt, crossP( ) 38
vektorprodukt, crossP( ) 38
venstre, left( ) 99
Vent på kommando 205
videresende fejl, PassErr 136
vilkårlig
matrix, randMat( ) 149
vilkårlig norm, randNorm( ) 149
vilkårlig stikprøve 150
vilkårligt
tal-seed, RandSeed 150
vinkel, angle( ) 10
vis data, Disp 55, 163
vis som
rektangulær vektor, ▸Rect 151
vise som
binær, ▸Base2 18
cylindrisk vektor, ▸Cylind 44
decimal vinkel, ▸DD 47
decimalt heltal, ▸Base10 19
grader/minutter/sekunder, ▸DMS 57
hexadecimal, ▸Base16 20
kuglevektor, ▸Sphere 180
polær vektor, ▸Polar 137
Viser grader/minutter/sekunder, ▸DMS 57

W

warnCodes( ), Warning codes 206
when( ), når 206
While, mens 207

X

x², kvadrat 221
XNOR 227
xor, Boolsk exclusive or 207

Z

zeroes( ), nulpunkter 208
zInterval, z konfidensinterval 211
zInterval_1Prop, en-proportionalt z konfidensinterval 211
zInterval_2Prop, to-proportionalt z konfidensinterval 212
zInterval_2Samp, z konfidensinterval med to-målinger 212
zTest 213
zTest_1Prop, enproportional z test . 214
zTest_2Prop, toproportional z test . 214
zTest_2Samp, to-prøvers z test 215

Æ

ægte brøk, propFrac 143

Δ

Δlist( ), listedifferens 107

χ

χ²Cdf( ) 25
χ²GOF 25
χ²Pdf( ) 26