

Andreas Rejbrand

Introduktion till Algoritmsimulator

Version 1.0.0.3

Innehållsförteckning

INNEHÅLLSFÖRTECKNING.....	2
INLEDNING.....	4
KONTAKTINFORMATION.....	4
NUMERISK RÄKNING I KONSOLLÄGET	5
PRIORITETSTABELL	5
OPERATORER.....	6
VARIABLER.....	7
INBYGGDA FUNKTIONER.....	8
<i>Trigonometriska och hyperboliska funktioner</i>	9
Trigonometriska funktioner.....	9
Inversa trigonometriska funktioner.....	9
Hyperboliska funktioner.....	9
Inversa hyperboliska funktioner.....	9
<i>Logaritmiska funktioner</i>	10
Logaritmfunktionen.....	10
Tiologaritmfunktionen.....	10
Tvålogaritmfunktionen.....	10
Den naturliga logaritmen.....	10
<i>Exponentialfunktionen</i>	10
<i>Absolutbeloppsfunktionen</i>	11
<i>Teckenfunktionen</i>	11
<i>Avrundningsfunktioner</i>	12
Standardavrundningsfunktionen.....	12
Nedåtarundningsfunktionen.....	12
Uppåtarundningsfunktionen.....	12
Heltalsfunktionen och decimalfunktionen.....	13
<i>Kombinationer och permutationer</i>	13
<i>Fakultetsfunktionen</i>	13
<i>Kvadratrotsfunktionen</i>	14
<i>Vinkelkonvertering</i>	14
Grader till radianer.....	14
Radianer till grader.....	14
<i>Koordinatkonvertering</i>	15
Rektangulära till polära koordinater.....	15
Polära till rektangulära koordinater.....	15
<i>Primtalsfunktioner</i>	16
Primtalsverifierare.....	16
Nästa primtal.....	16
Föregående primtal.....	16
Det n:te primtalet.....	16
<i>Största gemensamma delare</i>	18
<i>Fibonacci</i>	18
<i>Slumptalsfunktioner</i>	19
Diskreta slumptalsfunktionen.....	19
Reella slumptalsfunktionen.....	19
<i>Statistikfunktioner</i>	20
Antal observationer.....	20
Summa.....	20
Produkt.....	20
Populationsstandardavvikelse.....	20
Stickprovsstandardavvikelse.....	20
Medelvärde.....	21
Minsta värde.....	21
Första kvartilen.....	21
Median.....	21
Tredje kvartilen.....	21
Högsta värde.....	22
Frekvensen hos modalvärde(n).....	22
<i>Modulofunktionen</i>	23
DEFINIERA EGNA VARIABLER	24

<i>Ta bort en deklarerad variabel</i>	24
<i>Ta bort alla deklarerade variabler</i>	24
DEFINIERA EGNA FUNKTIONER	26
GÖR OM TAL I DECIMALFORM TILL BRÅKFORM.....	27
SVARSVARIABELN <i>ANS</i>	27
INSTÄLLNINGSVARIABLER	28
<i>Inställningsvariabeln _digits</i>	28
ANDRA SPECIALKOMMANDON	28
ALGORITMEDITORN	29
INMATNING	29
VÄRDETILLDELNING	29
VILLKOR.....	29
UTMATNING	29
SKAPA FLÖDESSCHEMAN	29
EXEKVERA ALGORITMER.....	30
EXEMPEL PÅ ALGORITM	31
<i>Vi byter inte dörr</i>	31
<i>Vi byter dörr</i>	32
FUNKTIONSANALYS	33
NUMERISK DERIVERING	33
NUMERISK INTEGRATION	33
NUMERISK LOKALISERING AV STATIONÄRA PUNKTER	33
SUMMOR OCH PRODUKTER.....	34
NUMERISK EKVATIONSLÖSNING	34
VÄRDETABELL	34
GRAFFÖNSTRET	36
STATISTISKA BERÄKNINGAR	38
INSTÄLLNINGAR	40
REFERENSFUNKTIONEN	40
SYSTEMKRAV	42
COPYRIGHTINFORMATION	43

Inledning

Detta dokument är avsett att ge användaren en introduktion till matematikprogramvaran Algoritmsimulator (härefter "Algosim"). Algosim är i grunden en numerisk kalkylator med funktioner för grafitning, statistisk envariabelanalys samt användardefinierade algoritmer. Programmet har analysfunktioner i form av numerisk derivering, integration och lokalisering av maxima, minima och terrassvärden. Vidare erbjuds numerisk ekvationslösning, beräkning av summor och produkter samt tabellfunktioner.

Detta dokument fordrar matematikkunskaper på gymnasienivå.

Kontaktinformation

Algoritmsimulator är utvecklat av Andreas Rejbrand och nya versioner kommer att publiceras på Internet:

<http://www.rejbrand.se/algosim>

All kritik, såväl positiv som negativ, välkomnas. Kommentarer likaväl som direkta frågor (som inte besvaras av detta dokument) kan skickas till Andreas Rejbrand på adressen andreas@rejbrand.se.

Numerisk räkning i konsolläget

Den första fliken i programmet, "Konsol", används till normala numeriska beräkningar. De flesta icke-grafiska funktioner i Algosim nås från detta läge. Undantaget är funktioner som tar annat än *reella tal* som argument, t.ex. funktioner för derivering och integration, vilka också tar en *funktion* som argument (den funktion som ska deriveras eller integreras).

För att utföra en beräkning, skrivs ett algebraiskt uttryck in i konsolen, följt av Enter. Algosim beräknar då uttrycket och presenterar resultatet på raden under. Uttryck som användaren skriver in färgas (som standard) svarta, medan svar som Algosim skriver ut färgas gröna.

Uttryck i Algosim följer normala regler för matematiska uttryck. Ett *uttryck* definieras här som något som returnerar ett reellt tal. Till exempel är 123 ett uttryck, liksom $\sin(0,6)$. Uttryck kan bestå av reella tal, parenteser, operatorer, funktioner och variabler. Märk att matematiska konstanter, tekniskt sett, också räknas som variabler.

Reella tal kan föregås av ett tecken (+ eller -) och kan innehålla ett decimaltecken (.). Specialoperatorn E används i betydelsen "gångar tio upphöjt till". Exponenten, d.v.s. det reella tal som kommer efter operatorn E, kan också föregås av ett tecken men måste vara ett heltal.

Parenteser används för att ange beräkningsprioriteten inom uttryck. Parenteser beräknas först. De parenteser som kan användas är (...), [...] samt { ... }. Alla parenteser måste stängas. Algosim tillåter obegränsat antal parentesnivåer. Eftersom Algosim stödjer funktioner och variabler med namn som innehåller fler än ett tecken, kan kortformen för multiplikation, såsom i $9x$ och $5\sin(2)$ inte användas. Multiplikationstecken måste alltid sättas ut: $9 \times x$ och $5 \times \sin(2)$.

Prioritetstabell

Följande prioritetsregler gäller:

1. (...), [...], { ... }
2. $f(x)$
3. !
4. ^
5. $\times$, /
6. +, -

För operatorer med samma prioritet, gäller att uttryck beräknas från vänster till höger. Till exempel gäller att $100/2/5 = (100/2)/5$.

Operatorer

Operatorer är symboler som utför operationer på intelligande tal, som kallas operander. Operatorer är antingen unära eller binära. En unär operator tar en operand, medan en binär operator tar två operander. De operatorer som kan användas i programmet följer i tabellen nedan. (R betecknar de reella talen, N de naturliga talen (de positiva heltalen samt talet 0) samt Z de hela talen.)

<i>Operator</i>	<i>Typ</i>	<i>Operander</i>	<i>Beskrivning</i>	<i>Exempel</i>
+	Binär	R	Addition	$10 + 30 = 40$
-	Binär	R	Subtraktion	$40 - 30 = 10$
$\times$, * eller $\cdot$	Binär	R	Multiplikation	$20 \times 50 = 1000$
/	Binär	R^1	Division	$1000 / 50 = 20$
^	Binär	R^2	Upphöjt till	$2^{10} = 1024$
!	Unär	N	Fakultet	$5! = 120$

Tabell 1. Operatorer

¹ Betrakta a/b . $b \neq 0$.

² Betrakta a^b . Om $a < 0$ så måste $b \in \mathbb{Z}$.

Varabler

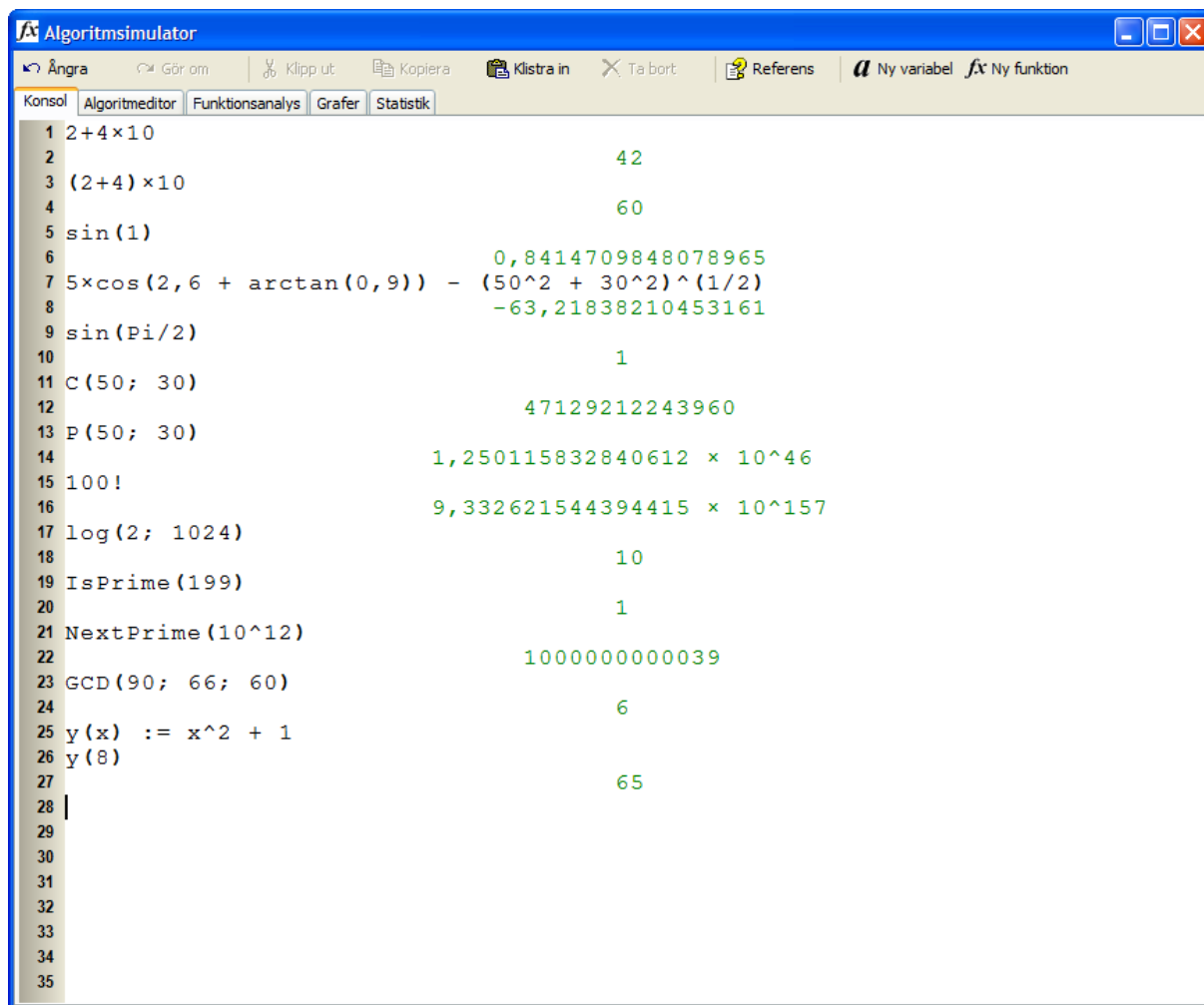
I Algosim räknas både konstanter och variabler som "variabler". Variablers namn kan innehålla tecknen A...Z, a...z, 0...9 samt `_`. Versaler och gemener är inte likvärdiga. I Algosim finns några fördefinierade matematiska konstanter.

<i>Namn</i>	<i>Värde</i>	<i>Beskrivning</i>
Pi	3,14159265358979324	Den matematiska konstanten π (pi); förhållandet mellan en cirkels omkrets och diameter.
e	2,71828182845904524	Den matematiska konstanten e .
Phi	1,61803398874989485	Den matematiska konstanten ϕ (fi); "det gyllene snittet".

Tabell 2. Fördefinierade konstanter

Inbyggda funktioner

Funktioner i Algosim kan ta ett godtyckligt antal argument större än 0. Tillåtna tecken i funktionsnamn är A...Z, a...z, 0...9 samt `_`. Dels finns inbyggda funktioner, dels kan användare själva deklarerera funktioner. På de följande sidorna beskrivs de inbyggda funktionerna.



The screenshot shows the 'Algoritmsimulator' window with the 'Konsol' (Console) tab selected. The console displays a list of 35 lines of code and their corresponding results. The results are shown in green text. The code includes various mathematical functions such as arithmetic operations, trigonometric functions, combinatorics, and a custom function definition.

Linje	Kod	Resultat
1	<code>2+4×10</code>	
2		42
3	<code>(2+4)×10</code>	
4		60
5	<code>sin(1)</code>	
6		0,8414709848078965
7	<code>5×cos(2,6 + arctan(0,9)) - (50^2 + 30^2)^(1/2)</code>	
8		-63,21838210453161
9	<code>sin(Pi/2)</code>	
10		1
11	<code>C(50; 30)</code>	
12		47129212243960
13	<code>P(50; 30)</code>	
14		1,250115832840612 × 10^46
15	<code>100!</code>	
16		9,332621544394415 × 10^157
17	<code>log(2; 1024)</code>	
18		10
19	<code>IsPrime(199)</code>	
20		1
21	<code>NextPrime(10^12)</code>	
22		1000000000039
23	<code>GCD(90; 66; 60)</code>	
24		6
25	<code>y(x) := x^2 + 1</code>	
26	<code>y(8)</code>	
27		65
28	<code> </code>	
29		
30		
31		
32		
33		
34		
35		

Figur 1. Konsolen

Trigonometriska och hyperboliska funktioner

Trigonometriska funktioner

$\sin(x)$, $\cos(x)$, $\tan(x)$, $\cot(x)$ returnerar sinus, cosinus, tangens respektive cotangens för x .

Exempel:

`sin(Pi/2)`

1

`cos(2, 9)`

-0,970958165149590522

Inversa trigonometriska funktioner

$\arcsin(x)$, $\arccos(x)$, $\arctan(x)$, $\text{arccot}(x)$ returnerar arcsinus, arccosinus, arctangens respektive arccotangens för x .

Exempel:

`arcsin(sin(0,85))`

0,85

Hyperboliska funktioner

$\sinh(x)$, $\cosh(x)$, $\tanh(x)$, $\coth(x)$ returnerar sinus hyperbolicus, cosinus hyperbolicus, tangens hyperbolicus respektive cotangens hyperbolicus för x .

Inversa hyperboliska funktioner

$\text{arsinh}(x)$, $\text{arcosh}(x)$, $\text{artanh}(x)$, $\text{arcoth}(x)$ returnerar arcsinus hyperbolicus, arccosinus hyperbolicus, arctangens hyperbolicus respektive arccotangens hyperbolicus för x .

Logaritmiska funktioner*Logaritmfunktionen*

$\log(n; x)$ returnerar n -logaritmen ur x , d.v.s. det tal a sådant att $n^a = x$.

Exempel:

$$\log(0,5; 0,002) \quad 8,96578428466208704$$

Tiologaritmfunktionen

$\log_{10}(x)$ returnerar 10-logaritmen ur x , d.v.s. det tal a sådant att $10^a = x$.

Exempel:

$$\log_{10}(2,5E-5) \quad -4,60205999132796239$$

Tvålogaritmfunktionen

$\log_2(x)$ returnerar 2-logaritmen ur x , d.v.s. det tal a sådant att $2^a = x$.

Exempel:

$$\log_2(1024) \quad 10$$

Den naturliga logaritmen

$\ln(x)$ returnerar den naturliga logaritmen (e -logaritmen) ur x , d.v.s. det tal a sådant att $e^a = x$.

Exempel:

$$\ln(9,82) \quad 2,28442112236637452$$

Exponentialfunktionen

$\exp(x)$ returnerar e^x .

Exempel:

$$\exp(1) \quad 2,71828182845904524$$

Absolutbeloppsfunktionen

$\text{abs}(x)$ returnerar x utan tecken. $\text{abs}(x) = \begin{cases} x & \text{om } x \geq 0 \\ -x & \text{om } x < 0 \end{cases}$

Exempel:

`abs(50)`
50

`abs(-50)`
50

Teckenfunktionen

$\text{sgn}(x)$ returnerar tecknet hos x . $\text{sgn}(x) = \begin{cases} -1 & \text{om } x < 0 \\ 0 & \text{om } x = 0 \\ 1 & \text{om } x > 0 \end{cases}$

Exempel:

`sgn(50)`
1

`sgn(-50)`
-1

`sgn(0)`
0

Avrundningsfunktioner

Standardavrundningsfunktionen

$\text{round}(x)$ avrundar x till närmaste heltal. Om x är precis mellan två heltal, avrundas x till det jämna talet.

Exempel:

$\text{round}(3, 2)$	3
$\text{round}(3, 6)$	4
$\text{round}(3, 5)$	4

Nedåtaavrundningsfunktionen

$\text{floor}(x)$ avrundar x nedåt till närmaste heltal.

Exempel:

$\text{floor}(6, 8)$	6
$\text{floor}(-6, 8)$	-7

Uppåtaavrundningsfunktionen

$\text{ceil}(x)$ avrundar x uppåt till närmaste heltal.

Exempel:

$\text{ceil}(6, 8)$	7
$\text{ceil}(-6, 8)$	-6

Heltalsfunktionen och decimalfunktionen

$\text{int}(x)$ returnerar heltalsdelen av x .

$\text{frac}(x)$ returnerar decimaldelen av x .

Exempel:

$\text{int}(6, 7)$

6

$\text{frac}(6, 7)$

0, 7

Märk att $x = \text{int}(x) + \text{frac}(x)$.

Kombinationer och permutationer

$C(a; b)$ ger antalet kombinationer på vilka b element kan plockas ur en mängd bestående av a element. Ordningen saknar betydelse; A, B, C är samma kombination som A, C, B.

$P(a; b)$ ger antalet kombinationer på vilka b element kan plockas ur en mängd bestående av a element. Ordningen är avgörande. A, B, C, är en annan kombination än A, C, B.

Exempel:

Hur många 5-siffriga tal kan skapas med siffrorna 1 tom 9? En siffra får endast förekomma en gång.

$P(9; 5)$

15120

Hur många olika grupper på 5 personer kan erhållas från en klass med 30 elever?

$C(30; 5)$

142506

Fakultetsfunktionen

$\text{fact}(n) = n!$

$n! = \prod_{k=1}^n k$ där $n \in \mathbb{N}$

Exempel:

$\text{fact}(5)$

120

Kvadratrotsfunktionen

$\text{sqrt}(x) = x^{1/2}$, d.v.s. kvadratroten ur x .

Exempel:

`sqrt(25)`

5

`sqrt(57600)`

240

Ett felmeddelande visas om `sqrt` anropas med $x < 0$.

Vinkelkonvertering

Grader till radianer

$\text{DTR}(x)$ returnerar vinkeln x (angiven i grader) angiven i radianer. $\text{DTR}(x) = \frac{\pi}{180} x$.

Exempel:

`DTR(270)`

4,71238898038468986

Radianer till grader

$\text{RTD}(x)$ returnerar vinkeln x (angiven i radianer) angiven i grader. $\text{RTD}(x) = \frac{180}{\pi} x$.

Exempel:

`RTD(0,952)`

54,5455820964543695

Koordinatkonvertering*Rektangulära till polära koordinater*

$RTP(x; y)$ konverterar de rektangulära koordinaterna (x, y) till polära koordinater (r, θ) . Variablerna r samt θ sätts. Värdet r returneras direkt.

Exempel:

$RTP(-10; 20)$

22,360679774997897

θ

2,03444393579570274

Polära till rektangulära koordinater

$PTR(r, \theta)$ konverterar de polära koordinaterna (r, θ) till rektangulära koordinater (x, y) . Variablerna x samt y sätts. Värdet x returneras direkt.

Exempel:

$PTR(10; \pi/4)$

7,07106781186547524

y

7,07106781186547525

Primtalsfunktioner

Primtalsverifierare

IsPrime(n) returnerar 1 om n är ett primtal; 0 annars.

Exempel:

IsPrime(51)	0
IsPrime(53)	1
IsPrime(2311)	1

Ett felmeddelande visas om $n \notin N$.

Nästa primtal

NextPrime(n) returnerar primtalet efter n , d.v.s. det lägsta primtal $p \geq n$.

Exempel:

NextPrime(10^{12})	10000000000039
------------------------	----------------

Ett felmeddelande visas om $n \notin N$.

Föregående primtal

PrevPrime(n) returnerar primtalet innan n , d.v.s. det högsta primtal $p \leq n$.

Exempel:

PrevPrime(10^{12})	999999999989
------------------------	--------------

Ett felmeddelande visas om $n \notin N$.

Det lägsta primtalet är 2. Om PrevPrime anropas med ett argument $n < 2$, så visas ett felmeddelande.

Det n :te primtalet

prime(n) returnerar det n :te primtalet. Observera att funktionen är tämligen långsam för $n \geq 100000$. Det första primtalet, prime(1), är 2.

Exempel:


```
prime(100000)
```

```
1299709
```

Ett felmeddelande visas om $n \notin N$ och om $n = 0$.

Största gemensamma delare

$\text{GCD}(a; b; \dots)$ returnerar den största gemensamma delaren till argumenten $a, b, \dots$

Exempel:

$$\text{GCD}(50; 25; 75) \quad 25$$

Om $\text{GCD}(a; b; \dots) = 1$, så är argumenten relativt prima.

Exempel:

$$\text{GCD}(7; 14; 50) \quad 1$$

Ett felmeddelande visas om något argument $\notin \mathbb{Z}$.

Även negativa tal och talet 0 kan förekomma.

Exempel:

$$\text{GCD}(-40; -120; -480) \quad 40$$

$$\text{GCD}(0; 482; 846) \quad 2$$

Om samtliga argument är 0, visas ett felmeddelande.

Fibonaccital

Fibonaccital är tal som genereras av funktionen f där $f(n) = \begin{cases} 0 & \text{om } n = 0 \\ 1 & \text{om } n = 1 \\ f(n-1) + f(n-2) & \text{om } n \geq 2 \end{cases}$

och $n \in \mathbb{N}$.

$\text{fibonacci}(n)$ returnerar det n :te Fibonaccitalet.

Ett felmeddelande visas om $n \notin \mathbb{N}$.

Slumptalsfunktioner*Diskreta slumptalsfunktioner*

`Random( $n$ )` väljer slumpmässigt ut ett heltal x , sådant att $1 \leq x \leq n$. Sannolikheten att ett specifikt heltal inom intervallet väljs, är således $1/n$.

Exempel (OBS! Svaren varierar!):

`Random(5)`

3

`Random(5)`

1

Ett felmeddelande visas om `Random` anropas med ett argument $n \notin \mathbb{N}$. `Random(o) = 1`.

Reella slumptalsfunktioner

`RandomReal( $x$ )` väljer slumpmässigt ut ett reellt tal a , sådant att $0 < a < x$.

Exempel (OBS! Svaren varierar!):

`RandomReal(5)`

3,22027781046926975

`RandomReal(5)`

4,99185574357397854

Statistikfunktioner

Samtliga statistikfunktioner tar ett godtyckligt antal reella värden – statistiska dataposter – som argument. Märk att samma statistiska material – d.v.s. samma argument – används i alla exempel.

Antal observationer

$n(a; b; \dots)$ returnerar antalet dataposter i materialet, d.v.s. antalet argument.

Exempel:

```
n(2,9; 6,8; 1,7; 6,5; 2,9; 4,5; 6,8; 3,8)
8
```

Summa

$\text{sum}(a; b; \dots)$ returnerar summan av dataposterna i materialet.

Exempel:

```
sum(2,9; 6,8; 1,7; 6,5; 2,9; 4,5; 6,8; 3,8)
35,9
```

Produkt

$\text{product}(a; b; \dots)$ returnerar produkten av dataposterna i materialet.

Exempel:

```
product(2,9; 6,8; 1,7; 6,5; 2,9; 4,5; 6,8; 3,8)
73480,518072
```

Populationsstandardavvikelse

$\text{SD}(a; b; \dots)$ returnerar populationsstandardavvikelsen hos materialet.

Exempel:

```
SD(2,9; 6,8; 1,7; 6,5; 2,9; 4,5; 6,8; 3,8)
1,87178891705234755
```

Stickprovsstandardavvikelse

$\text{SDNM1}(a; b; \dots)$ returnerar stickprovsstandardavvikelsen hos materialet.

Exempel:

```
SDNM1(2,9; 6,8; 1,7; 6,5; 2,9; 4,5; 6,8; 3,8)
```

2,00102652227728927

Medelvärde

`mean(a; b; ...)` returnerar medelvärdet av materialet.

Exempel:

```
mean(2,9; 6,8; 1,7; 6,5; 2,9; 4,5; 6,8; 3,8)
4,4875
```

Minsta värde

`min(a; b; ...)` returnerar det minsta värdet i materialet.

Exempel:

```
min(2,9; 6,8; 1,7; 6,5; 2,9; 4,5; 6,8; 3,8)
1,7
```

Första kvartilen

`Q1(a; b; ...)` returnerar första kvartilen i materialet, d.v.s. mittenvärdet i första halvan av de storleksordnade dataposterna.

Exempel:

```
Q1(2,9; 6,8; 1,7; 6,5; 2,9; 4,5; 6,8; 3,8)
2,9
```

Median

`Md(a; b; ...)` returnerar medianen i materialet, d.v.s. mittenvärdet av de storleksordnade dataposterna. Medianen benämns även andra kvartilen.

Exempel:

```
Md(2,9; 6,8; 1,7; 6,5; 2,9; 4,5; 6,8; 3,8)
4,15
```

Tredje kvartilen

`Q3(a; b; ...)` returnerar tredje kvartilen i materialet, d.v.s. mittenvärdet i andra halvan av de storleksordnade dataposterna.

Exempel:

Q3(2,9; 6,8; 1,7; 6,5; 2,9; 4,5; 6,8; 3,8)
6,65

Högsta värde

`max(a; b; ...)` returnerar det högsta värdet i materialet.

Exempel:

`max(2,9; 6,8; 1,7; 6,5; 2,9; 4,5; 6,8; 3,8)`
6,8

Frekvensen hos modalvärde(n)

`modfreq(a; b; ...)` returnerar frekvensen av modalvärdet eller modalvärdena (typvärdet eller typvärdena).

Exempel:

`modfreq(2,9; 6,8; 1,7; 6,5; 2,9; 4,5; 6,8; 3,8)`
2

(Modalvärdena är 6,8 samt 2,9, som båda förekommer två gånger i materialet.)

Modulofunktionen

$\text{mod}(a; b)$ returnerar resten vid heltalsdivisionen a/b .

Exempel:

$\text{mod}(10; 4)$	2
$\text{mod}(268; 42)$	16
$\text{mod}(10; 5)$	0

Definiera egna variabler

I Algosim kan användaren definiera egna variabler³. Detta sker i konsolen enklast med specialoperatorn för tilldelning, `:=`. För att deklarera en variabel och tilldela den ett värde, används följande syntax:

```
Variabelnamn := Uttryck
```

Exempel:

```
a := 50
b := a/10
Width := 100/(2*Pi)
```

Variabler som deklarerats kan sedan användas i uttryck i hela programmet.

Exempel:

```
Width × sin(b)
-15,2617538363448816
```

Värdet på en egendeklarerad variabel kan ändras i efterhand. Exempel:

```
a := 50
a := 100
a + 1
101
```

Ta bort en deklarerad variabel

För att ta bort en deklarerad variabel, används specialkommandot `#delete`, enligt följande syntax:

```
#delete Variabelnamn
```

Exempel:

```
#delete a
#delete b
#delete Width
```

Ta bort alla deklarerade variabler

För att ta bort alla egendeklarerade variabler, använd specialkommandot `#deleteAllVars`.

Exempel:

```
#deleteAllVars
```

³ De inbyggda konstanterna Pi, Phi och e är skyddade och deras värde kan inte ändras. De kan ej heller tas bort.

Variabler kan också deklarerars med dialogrutan "Ny variabel" som nås via knappen "Ny variabel" på verktygsfältet eller med kortkommandot Ctrl+Alt+V.

Definiera egna funktioner

Algosim tillåter användaren att definiera egna funktioner med godtyckliga antal reella argument, enligt syntaxen

Funktionsnamn(Arg1; Arg2; ...; ArgN) := Uttryck

där uttrycket innehåller Arg1, Arg2, ..., ArgN. Egendefinierade funktioner kan sedan användas i uttryck i hela programmet, och de kan definieras om.

Exempel:

```

y(x) := x^2
y(6)
                                     36

y(9)
                                     81

f(a; b) := (a + 2×b) / 3
f(5; 7)
                                     6,3333333333333333

10 + f(9; y(5))
                                     29,666666666666667

```

Om en funktion – oavsett om det är en inbyggd eller en egendefinierad sådan – anropas med ett felaktigt antal argument visas ett felmeddelande.

Funktioner kan också definieras med dialogrutan "Ny funktion" som nås via knappen "Ny funktion" på verktygsfältet eller med kortkommandot Ctrl+Alt+F.

Märk

I Algosim tillåts att en funktion har samma namn som en variabel – funktionen känns ändå igen på att den följs av tecknet "(".

Exempel:

```

y := 50
y(x) := x^2
y
                                     50

y(4)
                                     16

y(y)
                                     2500

```

Gör om tal i decimalform till bråkform

Algosim kan göra om värdet av den föregående beräkningen till formen p/q där p och $q \in \mathbb{Z}$ och (om inget avrundningsfel inträffar) är relativt prima. För att göra om den föregående beräkningens värde till bråkform används kortkommandot Ctrl+Mellanslag efter att beräkningen utförts (med Enter), d.v.s. när markören flyttats till en ny, tom, rad. Bråket p/q kommer då att infogas på den nya, tomma raden.

Exempel:

	0,123
123/1000	← ctrl+Mellanslag infogar bråket
	⋮
	-0,123456
-1929/15625	⋮
	0,496
62/125	⋮
	0,57777777777777778
26/45	

Svarsvariabeln Ans

Efter varje lyckad beräkning sätts specialvariabeln Ans lika med det beräknade värdet. Om en operator (+, -, ×, /, ^, !) sedan infogas först i nästa uttryck, kommer Ans att infogas innan denna. Detta "fortsätter" den föregående beräkningen.

Exempel:

50/8	
	6,25
Ans-40/9	← Ans infogas automatisk
	1,8055555555555556
Ans-1	
	0,8055555555555556

Inställningsvariabler

Inställningsvariabler är variabler, som samtliga inleds med `_`, vilka anger inställningar i Algosim.

Inställningsvariabeln `_digits`

`_digits` anger antalet gällande siffror som presenteras i beräkningsresultat i Algosim. Standardvärdet är 12, men detta kan ändras av användaren. Märk att antalet gällande siffror måste vara ett heltal i intervallet $[2, 18]$.

Exempel:

```
_digits := 18
Pi
3,14159265358979324

_digits := 12
Pi
3,14159265359

_digits := 3
Pi
3,14
```

Inställningsvariabler kan inte tas bort med specialkommandot `#delete`; emellertid återställs de till standardinställningarna om specialkommandot `#deleteAllVars` utförs.

Andra specialkommandon

Specialkommandot `#quit` avslutar programmet.

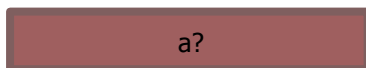
Exempel:

```
#quit
```

Algoritmeditorn

Med Algosim följer ett verktyg för att rita upp flödesscheman på skärmen, vilka programmet sedan kan följa. Man kan med andra ord skapa egna algoritmer som löser matematiska problem. Flödesscheman består av flödeselement (noder) som utför olika åtgärder. När Algosim utfört åtgärden vid en nod, så utförs åtgärden för den nod som är *länkad* till den tidigare noden. Länkar visas som streck mellan noder. De flödeselement (noder) som kan användas är följande:

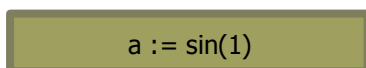
Inmatning



a?

Variabeln som anges (här a) tilldelas ett värde som användaren skriver in i en dialogruta som kommer upp när noden bearbetas.

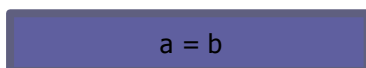
Värdetilldelning



$a := \sin(1)$

Variabeln som anges till vänster om tilldelningsoperatoren (här a) tilldelas värdet av uttrycket som anges till höger om tilldelningsoperatoren (här $\sin(1)$).

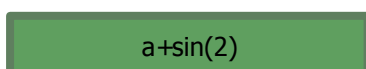
Villkor



$a = b$

Om villkoret som anges (här $a = b$) är uppfyllt så går Algosim vidare med den nod som länkas med en heldragen linje; i annat fall fortsätter exekveringen till den nod som länkas med en streckad linje.

Utmatning



$a + \sin(2)$

Värdet av uttrycket som anges (här $a + \sin(2)$) skrivs ut i en meddelandedialog.

Skapa flödesscheman

För att skapa ett flödesschema infogas noder som det ska bestå av med knappen "Infoga" under fliken "Algoritmeditor". För att ändra texten på en nod, kan användaren dubbelklicka på den. För att skapa en koppling mellan nod A och nod B, kan användaren dra med höger musknapp från nod A till nod B. För att skapa en streckad koppling (en *om falskt*-koppling), håll Shift nedtryckt samtidigt som kopplingen skapas. För att ta bort en koppling från nod A till en annan nod, dra med höger musknapp från nod A till sig själv. För att ta bort en streckad koppling, håll Shift nedtryckt samtidigt som kopplingen tas bort. För att ange vilken nod exekveringen ska börja med, håll Ctrl nedtryckt och klicka på den önskade inträdesnoden.

Noder kan fritt flyttas runt och alternativet "Justera till rutnät" ser till att objekt justeras mot ett osynligt rutnät, så att det lättare går att positionera dem sida vid sida.

För att spara en algoritm, välj knappen "Spara algoritm" eller kortkommandot Ctrl+S (Save). För att öppna en sparad algoritm, välj knappen "Öppna algoritm" eller kortkommandot Ctrl+O (Open). För att rensa fönstret och skapa en ny algoritm, välj knappen "Ny algoritm" eller kortkommandot Ctrl+N (New).

Märk att algoritmfiler som standard använder filtillägget ".algorithm".

En bildfil föreställandes flödesschema kan skapas och sparas med knappen "Spara flödesschema som bild". Bilden sparas i EMF-formatet (Win32 Enhanced Metafile), vilket gör att den kan förstöras, skalas om och skrivas ut med bibehållen kvalitet.

Exekvera algoritmer

För att exekvera den aktuella algoritmen, välj knappen "Exekvera". En del algoritmer kan ta väldigt lång tid på sig att utföras; en analog klocka visar hur lång tid algoritmen exekverats. För att avbryta en exekverande algoritm, välj knappen "Avbryt". För att direkt avbryta en algoritm som "låst sig" och inte kan avbrytas med ovan beskrivna metod, håll Ctrl nedtryckt och välj knappen "Avbryt tråden".

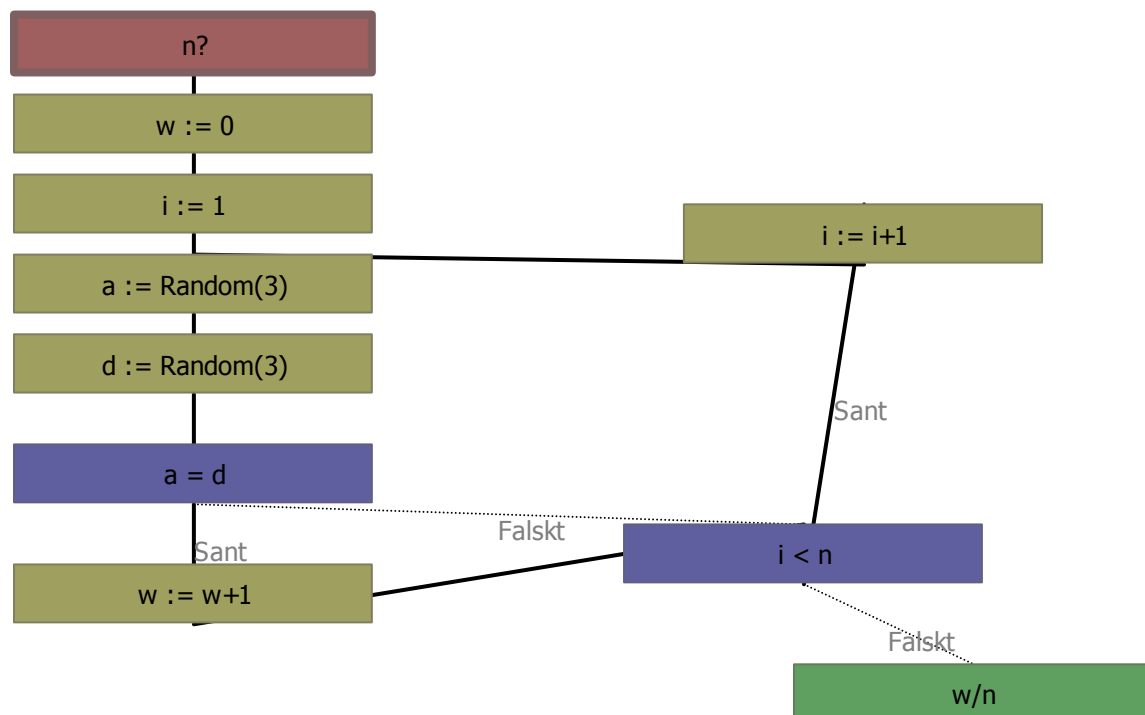
Exempel på algoritm

Ett välkänt sannolikhetsproblem är följande:

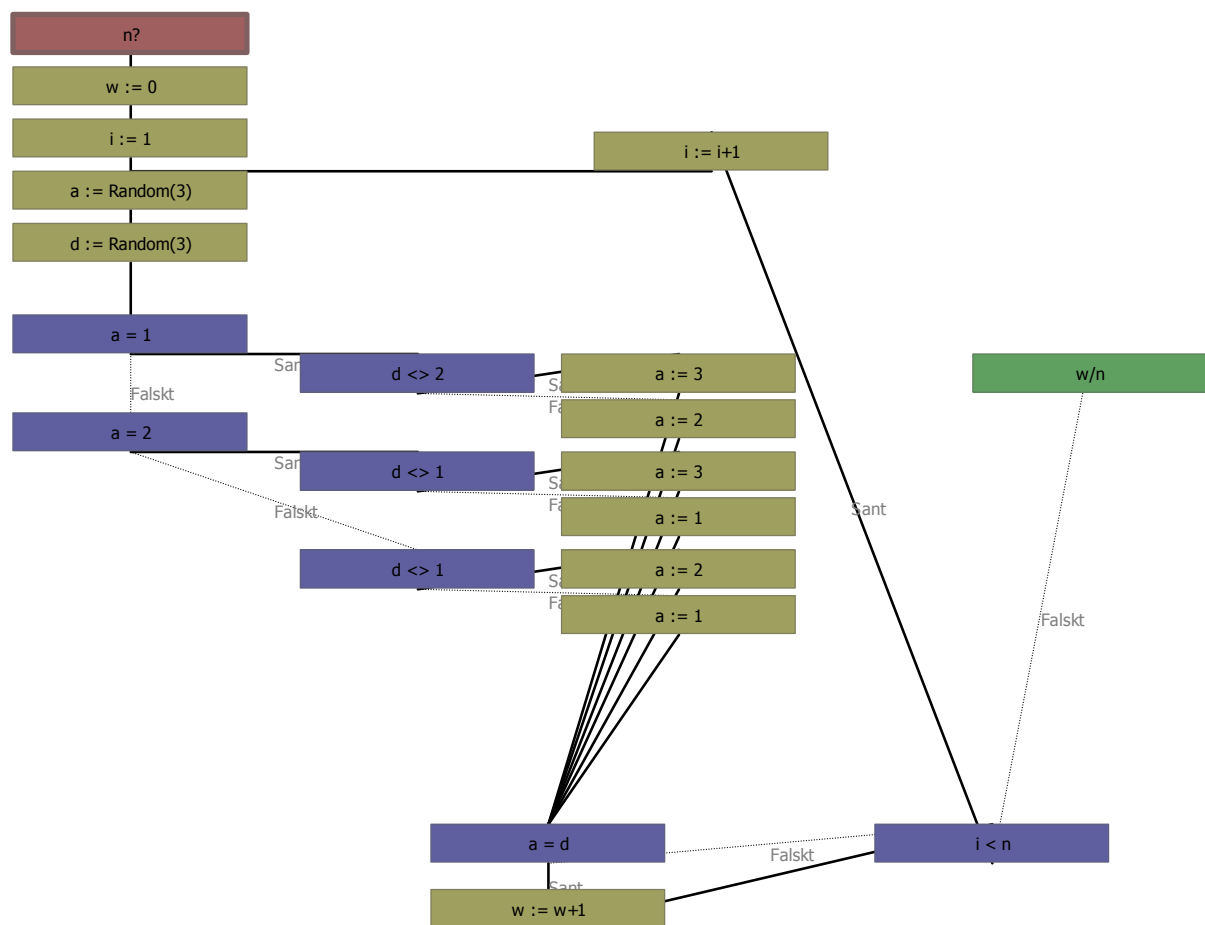
Individ I har vunnit en tävling och får därför tävla om en bil. Tävlingsledaren presenterar för I tre stycken dörrar: A, B och C. Bakom två av dörrarna finns leksaksbilar och bakom en av dörrarna finns den riktiga bilen. Individ I väljer dörr A. Sedan öppnar tävlingsledaren den dörr av de I inte har valt som innehåller en av leksaksbilarna, B. För att få maximalt hög sannolikhet att få den riktiga bilen, ska individ I stå kvar vid dörr A eller byta till dörr C?

Vi testar båda scenariona med två algoritmer!

Vi byter inte dörr



Exekvering av denna algoritm ger, med $n = 1000$, som väntat, $w/n = 0,329 \approx 1/3$.

Vi byter dörr

Exekvering av denna algoritm ger, med $n = 1000$, $w/n = 0,678 \approx 2/3$.

Vi bör således byta dörr⁴.

⁴ För en mer teoretisk diskussion kring problemet, se
Rejbrand, Andreas. *Ett klassiskt sannolikhetsproblem*. Katrineholm 2004.

Funktionsanalys

Under fliken "Funktionsanalys" återfinns funktioner för bl.a. numerisk derivering, integration och ekvationslösning.

Numerisk derivering

Ange namnet på den funktion som ska deriveras samt ange för vilket värde på argumentet x derivatan ska beräknas. Ange också h , halva avståndet i x -led mellan de två punkter på sekanten som approximerar tangenten. Man skulle kunna tro att avtagande h -värden ger ökande exakthet, men så är inte fallet. Ofta fungerar standardvärdet $h = 10^{-6}$ väl, medan mindre värden kan ge alltför stora avrundningsfel.

Algosim använder en central differenskvot för att approximera derivatan.

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

Välj "Derivera" för att beräkna derivatan. Derivatans approximerade värde skrivs ut i konsolen.

Välj "Beräkna tangentens ekvation" för att beräkna ekvationen av tangenten till funktionen i den aktuella punkten. Ekvationen ges av enpunktsformeln och derivatan:

$$y - f(x_0) = f'(x_0)(x - x_0)$$

Välj "Rita tangenten" för att rita upp tangenten i graffönstret. Märk att fönsterinställningarna måste vara lämpliga. Se avsnittet "Graffönstret" på sidan 36.

Numerisk integration

Ange namnet på den funktion som ska integreras och ange den undre integrationsgränsen a och den övre integrationsgränsen b . Ange också steglängden h . Standardvalet är att 500 steg ska användas totalt sett. Om gränserna t.ex. är 0 respektive 5 så blir $h = 0,01$. Klicka på "Auto" intill h -fältet för att anpassa värdet till de aktuella gränserna (så att antalet steg blir 500). Lägre värden på h ger bättre approximation, men beräkningen tar längre tid.

Klicka på "Integrera" för att beräkna integralen. Approximationen skrivs ut i konsolen. Algosim använder Simpsons formel för integralapproximationen.

Numerisk lokalisering av stationära punkter

Panelen "Extrempunkter och terrasspunkter" kan användas till att lokalisera stationära punkter, d.v.s. lokala maxima, minima samt terrasspunkter på grafen till en funktion. Välj funktionen samt gränserna a respektive b inom vilka sökningen sker. Steglängden h fungerar så att ett lägre värde ger ett exaktare resultat, men att tidsåtgången för beräkningen då ökar. "Auto" sätter h så att antalet steg blir 500.

Välj knappen "Maximum", "Minimum" eller "Terrasspunkt" för att finna den första lokala maximipunkten, minimipunkten eller terrasspunkten på kurvan från ursprungsvärdet a . Vid sökning efter terrasspunkter blir användaren förfrågad om hur stora tal får vara, om de ska bli betraktade som 0. Standard är tal under *nollgränsen* 0,0005.

Summor och produkter

Panelen "Funktionssummor och -produkter" används till att beräkna summan samt produkten av funktionsvärdena av en funktion mellan två gränser (inklusive gränserna). Ange funktionen samt gränserna a och b . Välj därefter om du vill beräkna summan (Σ) eller produkten (Π).

Numerisk ekvationslösning

Med panelen "Ekvationslösning" löses ekvationer med en variabel numerisk. Ange ekvationen samt den variabel som ska finnas. Ange också ett uppskattat värde på roten i fältet "Ursprung". Det föreligger emellertid inget krav att detta värde ska vara "nära" den riktiga roten. Ange också antalet iterationer som ska utföras. Teoretisk ökar exaktheten med ökande antal iterationer. I praktiken fungerar dock tämligen små värden i de flesta fall bra; i många fall ges åtskilliga korrekta decimaler med endast fem iterationer. Standardvalet är dock 50.

Välj "Solve" för att lösa ekvationen. Om ett beräkningsfel inträffar kan det löna sig att ändra värdet på ursprungsroten. Värdet skrivs ut i konsolen. Information om HL (högra ledet) och VL (vänstra ledet) i ekvationen, samt om differensens absolutbelopp $|HL - VL|$ visas också. Programmet provar således om den funna roten verkligen satisfierar ekvationen. Detta ger också användaren en uppfattning om hur bra approximationen är.

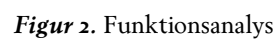
Ekvationslösaren använder Newton-Raphsons metod.

Värdetabell

Med funktionen "Värdetabell" kan användaren få en tabell uppritad, där första kolumnen består av funktionsargument från a till och med b (inklusive gränserna), med steget h mellan varje värde, medan andra kolumnen består av funktionsvärdet av motsvarande argument. Välj "Skapa tabell" när värdena är inskrivna, för att skapa tabellen. Tabellen använder Statistiklista x för den första kolumnen och Statistiklista y för den andra. Att funktionen använder statistiklistorna betyder att de framräknade värdena dels kan analyseras och dels plottas i graffönstret. (Se avsnittet "Statistiska beräkningar" på sidan 38.)

Exempel: Följande lista visar de 10 första primtalen. (Funktion: $\text{prime}(n)$; $a = 1$; $b = 10$; $h = 1$)

n	$\text{prime}(n)$
1	2
2	3
3	5
4	7
5	11
6	13
7	17
8	19
9	23
10	29



Graffönstret

Under fliken "Grafer" finns ett fönster där funktionsgrafer såväl som statistiska diagram ritas upp. Till vänster om fönstret anges områdets gränskoordinater i x -led och i y -led. "Scl"-värdena anger avståndet mellan stöddlinjerna (i rutnätet). Användaren kan också välja huruvida koordinataxlar, rutnät och etiketter (tal längs axlarna) ska ritas upp. Listan under dessa inställningar anger de funktionsgrafer som ska ritas upp samt deras färg. Följande fördefinierade färgvärden kan användas:

- clBlack (svart)
- clRed (röd)
- clGreen (grön)
- clBlue (blå)
- clYellow (gul)
- clNavy (mörkblå)
- clMaroon (mörkröd)

Andra färger kan också väljas, genom att användaren dubbelklickar i färgrutan.

Den aktuella bilden kan sparas som en fil i antingen EMF-formatet eller AlgoSim:s egna grafikformat ASG (AlgoSim-graf) med knappen "Spara bild".

ASG-formatet sparar endast funktionsgrafer - inte statistiska diagram. Graferna sparas med all matematisk information, vilket innebär att de i efterhand kan förstöras, förminskas och skalas om utan att kvaliteten försämras. Bildfilerna blir mycket små till storleken. AlgoSim-grafer kan emellertid endast öppnas i AlgoSim:s bildvisningsprogram. Användare av datorer som inte har AlgoSim installerat, kan inte öppna bildfilen.

EMF-formatet sparar hela bilden - både funktionsgrafer och eventuella statistiska diagram. Den mesta informationen sparas, förutom beträffande funktionsgrafer, vilka sparas som polygoner. Detta innebär att bilderna kan förstöras, förminskas, skalas om och skrivas ut med bibehållen kvalitet, förutom beträffande funktionsgraferna som blir "kantigare" ju mer bilden förstöras. Bildfiler tar lite större plats i detta format. De kan emellertid öppnas på alla Windows-datorer, skrivas ut och infogas i externa program såsom Microsoft Word.

	ASG	EMF
Innerhåller funktionsgrafer	Ja	Ja
Innehåller statistikdiagram	Nej	Ja
Sparar funktionsgrafer	Exakt	Approximativt (polygoner)
Filstorlek	Mycket liten	Medelstor
Kan förstöras och skalas om	Ja	Ja (funktionsgrafer "kantiga")
Kan öppnas i AlgoSim:s bildvisare	Ja	Ja
Kan öppnas i Windows	Nej	Ja

Tabell 3 Jämförelse mellan AlgoSim-grafer (ASG) och Win32 Enhanced Metafiles (EMF).

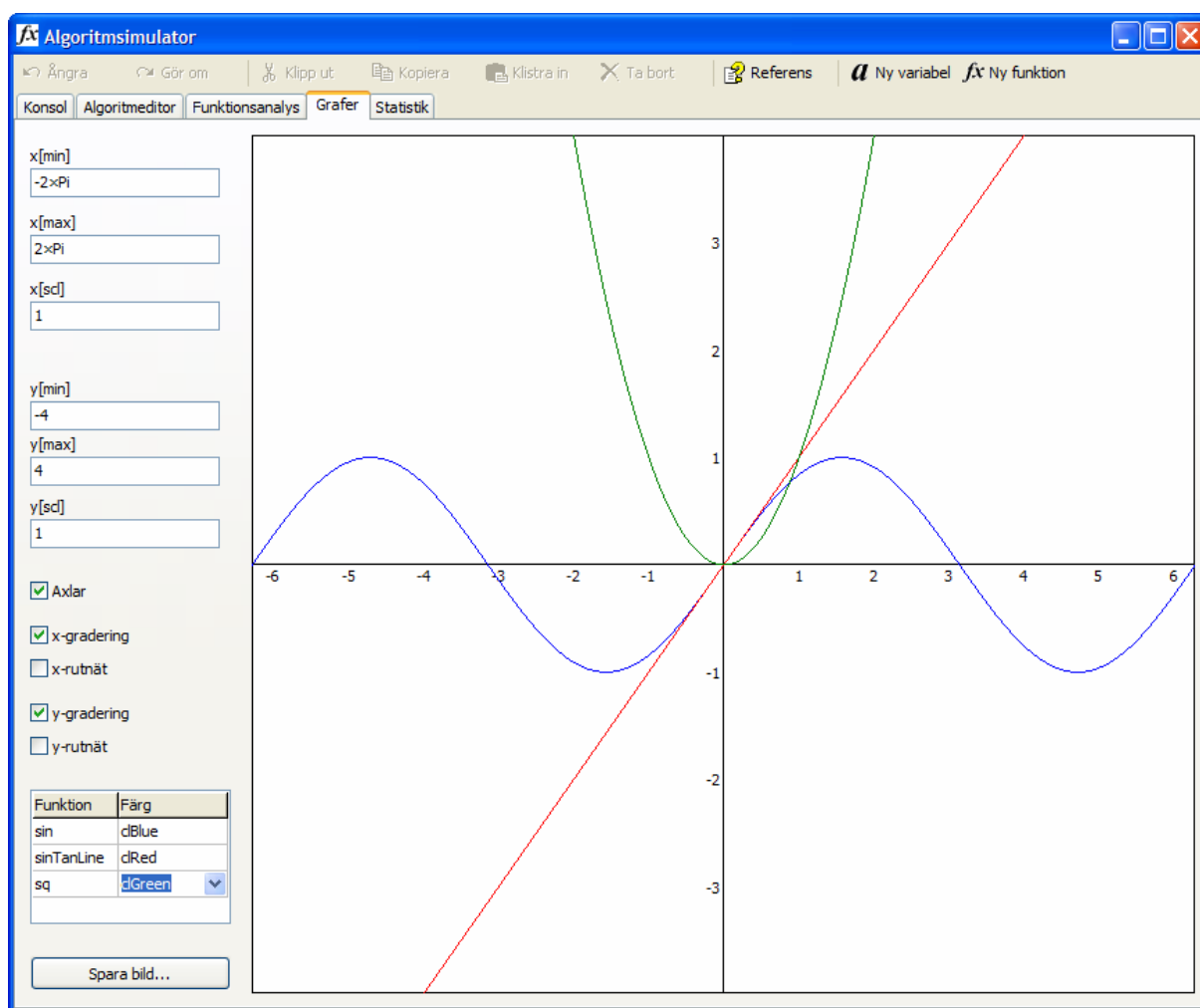
Förstoring av ett rektangulärt område i graffönstret kan göras genom att en rektangel dras med vänster musknapp i fönstret.

Märk

Det är inte möjligt att rita upp en graf till en funktion som tar fler än ett argument, eftersom y motsvarar funktionsvärdet och x motsvarar det enda argumentet. Det är emellertid möjligt att rita upp en graf ändå, om man låter ett av argumenten motsvara x och håller de andra argumenten fixa (konstanta). Detta kan göras med en extra funktionsdefinition.

$$\begin{aligned}y(a; b) &:= a + 2 \times b \\ y2(b) &:= y(a; b)\end{aligned}$$

Nu kan funktionen $y2$ av b ritas upp. Istället för att a är ett argument, så hämtas värdet från (den fixa) variabeln a .



Figur 3. Graffönstret. En tangent till sinusfunktionen har ritats med deriveringsfunktionen.

Statistiska beräkningar

Under fliken "Statistik" kan mer avancerade statistiska beräkningar enklare utföras än i konso-
len. Statistiska data skrivs in i Statistiklista x och, om materialet består av parade variabler, i
Statistiklista y. Med knapparna "Spara lista" och "Öppna lista" kan innehållet i ovanstående lista
sparas respektive läsas in från en fil.

För att analysera datan i en lista, väljs listan i fältet "Visa statistik för listan" samt knappen "Be-
räkna". De parametrar som beräknas är följande:

- n Antalet observationer
- Σ Summan
- Π Produkten
- μ Medelvärdet
- σ Populationsstandardavvikelse
- $\sigma(n-1)$ Stickprovsstandardavvikelse
- min Minsta värdet
- Q_1 Första kvartilen
- Md Medianen (Andra kvartilen)
- Q_3 Tredje kvartilen
- max Högsta värdet
- f-list Lista med de absoluta frekvenserna för respektive datapost

Om kryssrutan "Visa diagram" är ifylld ritas statistikgrafer upp i graffönstret. Den aktuella graf-
typen anges i listrutan "Typ". Följande typer kan användas:

- xy-plot
Plottar värdena (x, y) i graffönstret. Värdena hämtas från listorna enligt $(x_i, y_i) = (\text{Lista } X_i, \text{Lista } Y_i)$
- Frekvensstolpar
Använder endast Lista X. För varje unikt värde i listan ritas en stolpe upp på rätt x -
koordinat med höjden lika med frekvensen för värdet i materialet.
- Lådagram
Använder endast Lista X. En "låda" som markerad min, Q_1 , Md, Q_3 samt max ritas upp
med vertikalt centrum vid $y = 5$.



Statistikgrafer får den färg som anges i rutan "Färg".

Inställningar

Inställningar för Algosim nås via systemmenyn (högerklicka på den blå namnlisten överst i fönstret) och menykommandot "Inställningar".

Användaren kan ange en kommandofil (skriptfil) som körs varje gång programmet startar. En sådan fil ska vara en ren textfil och innehålla vanliga konsolkommandon och kan användas till att automatiskt definiera egna variabler och funktioner som användaren ofta använder direkt vid programstart.

Exempel på skriptfil som deklarerar några fysikaliska konstanter och höjer antalet gällande siffror:

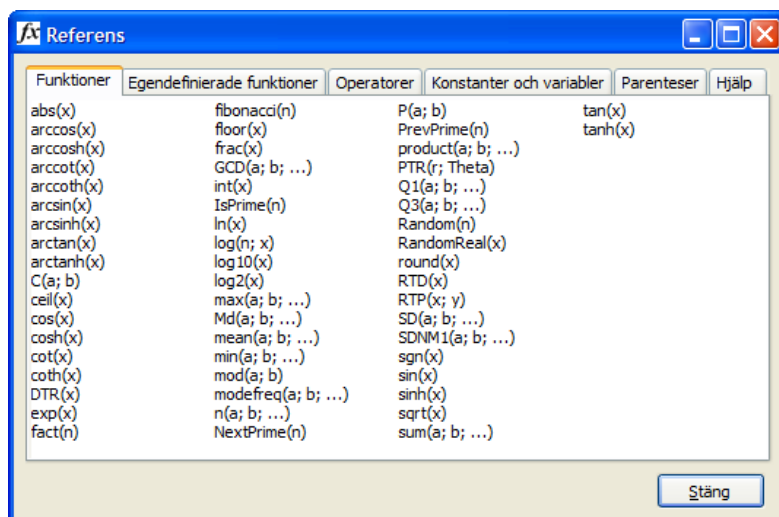
```
c := 299792458
g := 9,82
h := 6,6260687E-34
N := 6,0221419E23
_digits := 16
```

Också konsolens bakgrundsfärg, textfärg, teckensnitt och teckenstorlek kan ställas in. Alla inställningar sparas mellan sessionerna. Det är också möjligt att ange ett annat, externt, program som startas istället för AlgoSim när AlgoSim startas med Ctrl nedtryckt. (AlgoSim avslutas, och det externa programmet startas, således.) Detta kan bland annat vara användbart för användare med vissa tangentbord från Microsoft med en snabbknapp för Kalkylator. Med detta alternativ kan användaren starta två olika program med snabbknappen, beroende på om Ctrl hålls nedtryckt eller inte. Till exempel kan ett CAS startas istället för numeriska AlgoSim med Ctrl.

I systemmenyn återfinns också menykommandot "Visa/dölj verktygsfält" som visar och döljer verktygsfältet. Även kommandot "Om Algosim" återfinns. Detta kommando visar programinformation, versionsnummer och copyrightinformation om Algosim.

Referensfunktionen

Knappen "Referens" i verktygsfältet och kortkommandot F1 (Help) kan användas för att öppna ett fönster, där alla inbyggda funktioner, egendefinierade funktioner, variabler, operatorer och parenteser visas. Även kort referensinformation (om bl.a. hur variabler deklarerats) visas.



Figur 4. Referensfönstret

Systemkrav

Algosim är testat att fungera med följande rekommenderade systemkonfiguration:

	<i>Minsta krav</i>	<i>Rekommenderat</i>
Operativsystem	Windows 98, ME, 2000	Windows XP
Processor	Pentium 200 MHz	Pentium 4 2,0 GHz
RAM-minne	32 MB	256 MB
System för kantutjämning av skärmteckensnitt	(inget krav)	ClearType

Tabell 4. Systemkrav

Copyrightinformation

Algorithmsimulator är skapat av och utvecklas av Andreas Rejbrand (<http://www.rejbrand.se>).

Algosim © 2005-2006 Andreas Rejbrand

Algorithmsimulator är skyddat enligt svensk upphovsrättslag och får fritt spridas utan vinstsyfte om alla originalfiler inklusive installationsprogrammet och alla textfiler (inklusive denna manual) medföljer.

Algosim

Copyright © 2005 Andreas Rejbrand

<http://www.rejbrand.se/algosim>