

## Welcome to Kalkulator



Welcome to Kalkulator! I hope you will like the program and find it useful. Please let me know what you like and dislike about it, so that I can keep improving it in the future.

Kalkulator is shareware. If you are new to the program, please read the Registration and Support page. You may also want to check What Does Kalkulator Do. Another good place to start is Evaluating Expressions.

If you find a feature not described in this help, just try it out -- it should be self-explanatory.

If you are not a hard-core scientific and engineering math user, do not feel intimidated by the more advanced capabilities of the program. The features you do not use will not get into your way; just forget about them and use only those you need. After one hour with Kalkulator you will feel at home.

- [Click here to create a separate Contents window](#)
- [Click here to see an overview of the Main Panel](#)
- [Click here to see the credits and copyrights](#)

Kalkulator Help File Version 1.101

# Contents

## Introduction

- [Welcome to Kalkulator](#)
- [What does Kalkulator do?](#)
- [Registration and support](#)
- [Updates and upgrades](#)

## How to use Kalkulator

- [The Main Panel](#)
- [Evaluating expressions](#)
- [Plotting functions](#)
- [Numerical operations](#)
- [Unit conversion](#)
- [Statistical operations](#)

## Reference

- [Constants](#)
- [Variables](#)
- [Operators](#)
- [Functions](#)
- [Expression syntax](#)
- [Keyboard](#)
- [Menu operations](#)
- [Questions and answers](#)

## What does Kalkulator do?

At its simplest level, Kalkulator is a calculator. It will accept an arithmetic expression, compute its value, display the result and store it for a later use. For example, entering

```
5+3*log 1000
```

and pressing Enter will compute and display the expression value, 14, storing it as t (the Temporary Variable, used unless you specify another one). Instead of pressing Enter, you may use the "=" key (or button), to choose a variable from a pop-up menu; this variable will be used to store the result.

Variables can be also used as components of expressions, as in

```
2.5*ln(3+exp X)-sin(2A+3t)
```

Being able to see the whole expression before and after evaluation, you can modify and recompute it at any time. This already is enough to justify using Kalkulator instead of any alternative. (Try to compute the same expression with any other calculator program.)

An expression can be more than just evaluated. Treated as a function of t, it can be used to compute integrals, derivatives, sums or roots. It can be also plotted on the screen.

The calculation result can be converted between various measurement units, or displayed in a selected format (for example, as a fraction or as degrees, minutes and seconds).

Kalkulator also has a data buffer to store a number of data points; it can perform simple statistical operations on those points: computing population parameters, drawing histograms or scatter plots (with polynomial curve fitting), etc. It will also recompute, using a given formula, all X or Y values in the buffer.

## Registration and support

Kalkulator is distributed as shareware: you are granted a trial license for a period of 30 days to see whether you will find the program useful and worth the price I'm asking. After that you have the options of either purchasing Kalkulator, or removing it from your system.

If you decide to keep Kalkulator, registering (i.e. buying it) will bring you additional benefits: the newest program version with enhanced features, free updates and technical support. Last but not least, the program will also get rid of the nagging screen which will start showing up after one month expires.

Technical support is offered not only to registered users, but also (to a reasonable extent) during the trial period. If you have any questions, problems or suggestions, feel free to write at my address or (preferably) use email:

<b>CompuServe:</b>	<b>Andrzej (or 70611,2552)</b>
<b>America Online:</b>	<b>Andrzej</b>
<b>Internet:</b>	<b>wrotniak@freeflight.com</b>
<b>or</b>	<b>andrzej@compuserve.com</b>
<b>or</b>	<b>andrzej@aol.com</b>

And, most importantly, let me know what you like and dislike about the program: what new features would you like to see, what do you find inconvenient, confusing or mathematically incorrect (nah!). Your feedback is most valuable in making Kalkulator the program I want it to be.

### See also:

[How to register](#)

[Passing copies on to others](#)

[Updates and upgrades](#)

[How to get the newest version](#)

[Vendor Information](#)

## How to register

This is very simple: just send a check or money order for \$40 (U.S. currency only, drawn on a U.S. bank) to my address:

**Dr. J. Andrzej Wrotniak**  
**2057 Happy Lane**  
**Crofton, MD 21114-1917**  
**USA**

Thank you. Please include the following details:

- Your name and shipping address
- The version number of the evaluation copy you have
- Your email address, if applicable

Upon registration a diskette with the latest program version will be shipped to you promptly. If you have a CompuServe account, I can just send you a ZIPped file via email -- it will be faster. Please let me know which way do you prefer (otherwise I will use the regular mail).

Upon request, you will receive a receipt for tax or audit purposes. For corporate and institutional users site licenses are available.

### See also:

[Available discounts](#)

[Registration on the CompuServe](#)

[Registered version -- enhanced features](#)

## **Discounts available**

Here are the special prices for some categories of users:

- Full-time students -- \$30. Please enclose a proof (e.g., a copy of a school ID) with order.
- Registered users of Ex or Spheric on the HP Palmtop/ PC-DOS platform, and of El Cal or Star Base on the Atari ST -- \$30.

The discounted prices (check, cash or money order in US\$) are applicable only to orders by mail.

## **Registration on the CompuServe**

Members of the CompuServe can register Kalkulator on-line.

To register, GO SWREG at any prompt (or click on the traffic light icon at the top of your WinCIM screen and type SWREG), then follow the directions. The registration number is #8743, file name -- KALKUL.ZIP.

The registration fee of \$40 will be included into your next CompuServe bill.

Please note that none of the discounts mentioned elsewhere in this document are available when registering this way.

## Registered version -- enhanced features

The registered version of Kalkulator enhances some of the features of your evaluation copy:

- The program uses 24 expression lines, not just six
- The data buffer stores 6400 or 3200 data points (in the X or XY mode, respectively), not just 128 or 64
- Multiple status (.KAL) files can be used
- There is a capability to write the data buffer to an ASCII file or to read from it; this allows to exchange the data with other programs
- The polynomial regression goes up to the ninth degree, not to the second
- You can reseed the random number generators

### See also:

[How to register](#)



## **Passing copies on to others**

You are allowed and encouraged to pass unregistered copies of Kalkulator along as long as you do not alter the original distribution files and do not charge money for the program.

Distributing of the original archive (or a floppy with its freshly unpacked contents) is recommended: if you just take the files off your system, Kalkulator will count the elapsed part of your trial period towards that of the new recipient.

No distribution of the registered copies of the program or the accompanying files is allowed.

## Updates and upgrades

A program is like a living thing: it evolves, grows, changes. Kalkulator is no exception here and it follows the same pattern.

Every new release of our program is denoted with a decimal version number, for example, 1.53. The following numbering scheme has been adopted:

- A change in the last decimal digit (e.g., from 1.53 to 1.54) denotes a minor update. These are maintenance releases, including bug fixes, touch-ups and minor improvements.
- A change in the first digit after the decimal point (e.g., from 1.54 to 1.60) denotes an update: some new features, visible improvements to the user interface etc.
- A change in the digit preceding the decimal point (major version) reflects an upgrade, containing significant new features. Possible examples of new features justifying an upgrade (these are NOT promises!) might include matrix algebra, complex variable arithmetic or differential equations; usually more than one such feature at a time.

Updates to Kalkulator are free (if distributed electronically), and upgrades for registered users, significantly discounted.

### See also:

[Update and upgrade policy](#)

[Where to get the newest version](#)

## Update and upgrade policy

### Updates

For registered users of Kalkulator, updates within the same major version are free. This means that, for example, those who registered Version 1.12 do not pay for updating to versions up to 1.99 inclusive, regardless of when these new versions are released.

After you download the new version from an on-line service (or get it from any other source), just copy its files into the directory where your registered Kalkulator resides (do NOT delete any of the old files!), and the program will recognize your registration.

If you prefer to have a disk with the update sent to your address, this service requires a shipping fee of \$5.

### Upgrades

Program upgrades to the next major version will be available to registered users of previous versions at a 50% discount from the new version price.

If, however, an upgrade is released within six months from the date you registered the program, you can receive it as if it were an update (emailed for free, shipped for \$5). This policy also includes switching to the Windows 95 version of Kalkulator.

### See also:

[Does Kalkulator work under Windows 95?](#)

## **Does Kalkulator work under Windows 95?**

Although Kalkulator has been written as a 16-bit application, it is well-behaved and runs smoothly under Windows 95 (or NT). As a matter of fact, since Version 1.04 I am using Kalkulator mostly in Win95.

I will consider making Kalkulator a 32-bit application by recompiling it under the '95 version of Delphi when it becomes available in Spring '96, although there is no real need to do it ("if it ain't broke, don't fix it").

## Where to find the newest version

The newest version of Kalkulator can be downloaded at no cost (other than the connection itself, that is) from one of the following sources:

- World Wide Web, courtesy of my employer, Aerospace Engineering and Research, Inc. This is the Kalkulator's primary distribution site; the most recent version can be available here a couple of weeks before it appears anywhere else. Point your browser to

<http://www.freeflight.com/wrotniak/>

- CompuServe: the Science Forum (GO SCIENCE), Library 5 (Mathematics), file name: KALKUL.ZIP (the same file can be also found on some other forums).
- America Online: Software Libraries - Windows - Applications - Productivity - Calculators - KALKUL.ZIP (you may have to search for the file by name, as the AOL people tend to move things around).

## Vendor information

Non-registered copies of Kalkulator can be distributed by BBS operators, on-line services, disk and CD-ROM vendors after obtaining my explicit permission.

This permission is not required for vendors and BBS operators approved by the Association of Shareware Professionals. In these cases contacting me, although not required, is still recommended, as I may be able to provide the most recent version of the program.

In any case it is required that

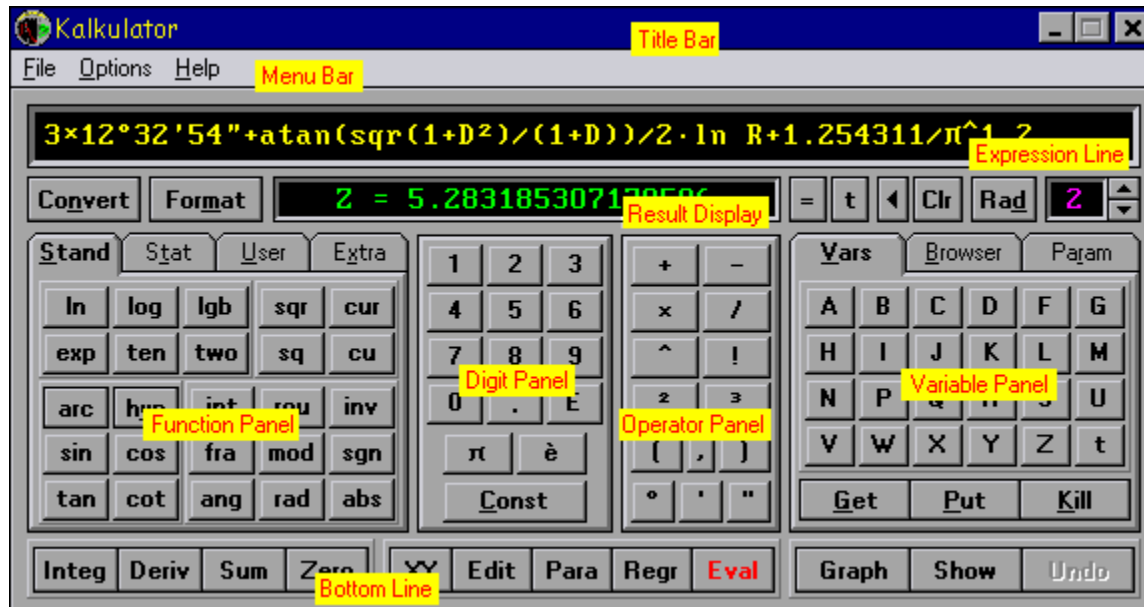
- All files in the original distribution archive are included without alteration, as listed in the PACKING LIST section of the accompanying README.1ST file.
- The recipient of the program is informed in advance that the distribution fee paid to the vendor is not equivalent to purchasing the program: a registration is required if the recipient continues using Kalkulator after the expiration of the trial period.

### See also:

[The author's address](#)

[The author's email address](#)

## The Main Panel

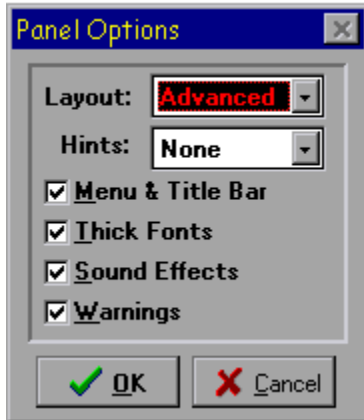


Click on the yellow stickers for more information about Main Panel components.

**See also:**

[Panel options](#)

## Panel options



The appearance and behavior of the Main Panel can be adjusted from the Panel Options entry in the Options menu. A dialog box will present you with a number of choices.

All these settings are saved in the KALKUL.KAL file when you exit the program, and loaded back when you run it again -- there is no need to customize Kalkulator every time you use it.

- Layout: Advanced (the whole Main Panel shown), Basic (no Bottom Line) or Minimal (just the Expression Line, Result Display and the buttons next to it).
- Hints: Balloon (comic book style), System (Windows standard) or None.
- Menu & Title Bar -- removes or restores the Main Menu and the Title Bar. Without the Title Bar the Main Panel can still be moved around by click-and-dragging at its edges, and the Main Menu functions are also available from the Main Pop-up Menu.
- Thick Fonts -- toggles between thin and bold fonts in numeric displays (these on the black background). If your monitor is not very sharp, then you will probably prefer bold fonts. On the other hand, thin fonts look great on active-matrix displays.
- Sound Effects -- toggles sound effects on and off.
- Warnings -- disables or enables the program warnings. With warnings disabled, the program status file will be always saved upon exit.

### See also:

[Sound effects](#)



## Sound effects

Kalkulator uses sound effects to denote some events. This can be turned off or on from the Panel Options dialog.

The sounds are stored in standard Windows' .WAV files in the Kalkulator home directory. The following files are included:

- KALK\_ERR.WAV -- the sound used for errors (syntax or arithmetic)
- KALK\_DON.WAV -- to announce a completed calculation
- KALK\_OK.WAV -- denotes some other operations successfully completed
- KALK\_INP.WAV -- accompanies an input box (numeric or not)
- KALK\_QUE.WAV -- accompanies a question (Yes/No or Yes/No/Cancel)
- KALK\_SHO.WAV -- well, this one is used just to spice the program up...

You may copy any .WAV files of your choice (renamed as above) to the Kalkulator directory; these sounds will be then used by the program.

Although the supplied sound files are rather small, you may want to save some disk space by deleting them from the Kalkulator directory. In that case it is recommended (although not required) that you turn the sounds off, as this will speed up the program operation.

## Evaluating expressions

First, enter the expression into the Expression Line. Use the mouse or type the expression in. In the latter case you will need to use special keystroke combinations to enter some of the symbols.

The expression syntax follows (mostly, at least) the customary "blackboard" notation.

After the expression has been entered, you have to tell Kalkulator where to store the result: press the "=" key or click on the [=] button, then select a variable from a pop-up menu. Instead of doing that, you may just press the Enter key -- the result will then be stored in the Temporary Variable, t.

If the expression syntax is not proper, an error will be reported and you will be able to correct the mistake. The cursor will be placed at the error.

Otherwise Kalkulator will try to evaluate the expression. Some errors may occur at that stage (e.g., division by zero, or referring to a variable without a previously assigned value). If this is the case, another error alert will show up.

In no case should an error in an expression make Kalkulator crash or misbehave in any other way. All that may happen is an error alert, specifying the function or operator where the error occurred.

The result of the calculation will be shown in the Result Display. The [Format] button can be used to change the display precision and/or format.

### See also:

[Checking expression syntax](#)

[Expression syntax](#)

[Accuracy](#)

## Checking expression syntax

You may check the syntax of the current Expression Line, without actually computing it, by pressing Ctrl-Enter. If an error is detected, a diagnostics will be displayed; if the syntax is OK -- the expression will just briefly blink green.

## Expression syntax

Functions and operators are executed from high to low priority:

- High: power, square, cube, factorial, implied multiplication, functions
- Medium: multiplication and division
- Low: addition and subtraction, unary plus and minus

An operand with operators on both sides will associate with the higher priority operator of the two. If both priorities are the same, then the left operator (low and medium priority) or the right one (high priority) is applied first.

This order, following the customary "blackboard" notation, can be arbitrarily changed with parentheses. Of course, you may always do it when in doubt.

### See also:

[Expression syntax examples](#)

[Syntax restrictions](#)

[Upper and lower case](#)

[Implied multiplication](#)

## Expression syntax examples

$2*\sin 2x$	means	$2*\sin(2*X)$
$2\sin 2*x$		$2*\sin(2)*X$
$a^b^cd$		$A^(B^(C*D))$
$a+b*cd^3.1$		$A+(B*(C*(D^3.1)))$
$a \sin hx$		$A*\sin(H*X)$
$\operatorname{asinh} x$		$\operatorname{asinh}(X)$
$\ln x^2$		$\ln(X^2)$

## Syntax restrictions

The "blackboard" notation is not always clear and consistent. Therefore, to avoid ambiguous expressions, we have to introduce a number of (hopefully, reasonable) restrictions.

- High-priority operators:  $^2$ ,  $^3$ ,  $!$ ,  $^{\wedge}$ , implied multiplication, cannot follow a parenthesis enclosing a function argument (or arguments)

For example, the sequence " $\log(x)^2$ " is illegal; it should be changed into " $(\log x)^2$ " or " $\log(x^2)$ ", depending on the intended meaning. On the other hand, " $\log x^2$ " is OK and means " $\log(x^2)$ ".

- Sequences like " $\sin^2x$ ", often used to denote " $(\sin x)^2$ ", are not recognized; the latter, explicit form has to be used.

**See also:**

[Implied multiplication](#)

## Implied multiplication

The multiplication operator,  $\times$ , between two tokens (operands or symbols) can be skipped if

- the first one is a variable, a constant, a postfix operator or a closing parenthesis (except when used for function arguments), AND
- the second one is a variable, a symbolic constant, a function, or an opening parenthesis.

The difference between explicit and implied multiplication is that the latter has higher priority.

### Examples

Legal	Equivalent to	Illegal
$3VXZ$	$3 \times V \times X \times Z$	$VX3Z$
$2\pi R$	$2 \times \pi \times R$	$2 \ 3.14R$
$9 \sin X$	$9 \times \sin(X)$	$\sin(Y) \sin X$
$\ln X \exp Y$	$\ln(X \times \exp(Y))$	$\ln(X) \exp Y$
$\sin 2AB$	$\sin(2 \times A \times B)$	$\sin(2A) B$
$\sin X(A+B)$	$\sin(X \times (A+B))$	$\sin(A+B) X$
$\pi R^2$	$\pi \times (R^2)$	$R^2 3.14$
$(A+B)(C+D)$	$(A+B) \times (C+D)$	$\ln(A+B)(C+D)$

### Note

The traditional notation is not quite consistent: "exp 3 ln X" is usually interpreted by a human reader as "exp(3×ln(X))" while "2 sin X cos X" as "2×sin(X)×cos(X)". Kalkulator always chooses the first way, interpreting the latter example as "2×sin(X×cos(X))".

### See also:

[Syntax restrictions](#)

## Upper and lower case

Expressions do not have to be entered in a case-sensitive manner. If a lowercase letter does not make sense in a given context, it may be raised to uppercase (becoming a variable).

This is convenient, as most of us prefer to type "2axsinyh" rather than "2 AX sin YH", letting the computer to sort it out.

Note, however, that typing "2xasinh" will be interpreted as "2X asinh Y": the program recognizes "asinh" as a function name and promotes to uppercase only as few characters as necessary.

In case of doubt, you may always add a separating space, or enter a variable name in upper case (it will never be demoted to lower).



## Accuracy

The internal accuracy of individual arithmetic operations performed by Kalkulator is 19 good decimal digits or better. These errors will, however, accumulate in computations involving multiple operations, so that 18 good digits is a safer assumption.

Moreover, algorithms used to compute some less trivial functions may have their own, additional error margin.

To be on the safe side, Kalkulator displays only sixteen significant digits of results. In a great majority of cases all these digits will be valid with a reasonably large safety margin.

(Keep in mind, however, that for any finite floating-point accuracy one can easily invent an example of a computation where the relative result error will be 100% or more -- i.e., all decimal digits wrong.)

### See also:

[Problems with some arithmetic coprocessors](#)

[The Pentium bug](#)

## Known problems

### Problems with some arithmetic coprocessors

Surprisingly, I have discovered that Delphi code executed on some of the Intel processors (some, but not all, 486 DX2 at 50 or 66MHz), provides only 16 good decimal digits in some divisions, depending on the current stack state.

In such cases, operation "x := y/z", using x, y and z of the Extended type (19-digit accuracy) gives x with only 16 good decimal digits.

To check whether your processor is affected, type in

```
1/3-.333333333333333333
```

and hit Enter (Kalkulator will reformat the expression, truncating the decimal fraction from 20 to 18 digits). If the displayed result is 2E-20 or so, then your coprocessor is OK (this also happens if your computer does not have one and floating point calculations are done by software emulation); a result of about 2E-17 indicates that your coprocessor has problems.

A similar problem existed under Turbo Pascal 5.5 or 6.0. It could not be detected using other compilers, as C or C++ provide floating-point types with at most 16 digits of accuracy.

In addition to reducing the accuracy of calculations, this problem may also affect reformatting the result as a fraction (when you choose "Fraction" from the menu popping up after clicking on "Format" in the Main Panel).

## **Pentium bug**

The infamous bug in some of the early Pentium processors is related to some division operations. Kalkulator detects a faulty processor and uses additional code to work around the problem. (Frankly speaking, it was the good people of Borland who took care of this: the Delphi compiler, used to develop this program, does it automatically.)

## Plotting functions

### Plotting a function on an empty canvas

First, enter the function into the Expression Line, using  $t$  as the function argument (a "dummy parameter"). Then click on [Graph] in the Bottom Line. You will be asked for the  $x$ - and  $y$ -limits by the Graph Range dialog, from which you also can set one or both axes to a log scale.

The Graph Canvas will now show up, on which the drawing takes place, obscuring a large part of the Main Panel. The drawing process can be interrupted by pressing any key.

### Hiding and showing the Graph Canvas

Click on [Hide] (or press Enter or Escape) to hide the Graph Canvas without erasing the graph. Click on [Show] (the same button, renamed at that time) to display it again.

### Adding a function plot to an existing graph

While the graph canvas is active, you may click on [Add] to add the function defined by the current Expression Line to the canvas. Obviously, this time you will not be asked for the range. You also can add a function plot to an existing histogram or scatter plot, but not vice versa. The last added plot can be removed from the canvas by clicking on the [Undo] button.

### Coordinate readout

A left mouse click inside the canvas invokes a pop-up menu with coordinates of the selected point. The menu allows you to store the  $x$ - and/or  $y$ - values of the point in the variables  $X$  and/or  $Y$ .

### Copying canvas to clipboard

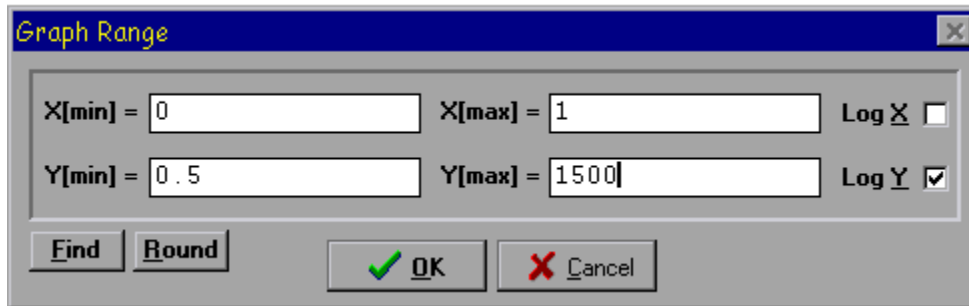
Pressing the F12 key while the Graph Canvas is up, copies the canvas to the Windows clipboard. From here it can be pasted into any application accepting the clipboard Paste command.

### See also:

[Graph range](#)

[Graph options](#)

## Graph range



Graph Range

X[min] = 0 X[max] = 1 Log X

Y[min] = 0.5 Y[max] = 1500 Log Y

Find Round OK Cancel

The Graph Range dialog box is used to enter x- and y-limits for function plots and regression line graphs. Any valid arithmetic expressions can be entered.

Clicking on [Find] will estimate the y range for the x-limits previously entered, except for the case of regression graphs, where the x range will also be estimated. On a similar basis, the [Round] button will round the limits to "reasonable" values.

The buttons at the right set the x- and/or y-scale to logarithmic. Obviously, this requires that the corresponding range limits are positive.

### See also:

[Entering numerical values](#)

## Entering numerical values

At some times Kalkulator will ask for some floating-point numerical values (e.g., plot range, integration interval limits, a new value for a variable, etc.) to be entered into editable fields of a dialog box.

The program will accept here not just literal values (i.e. numbers in a usual notation), but also any legal arithmetic expressions, following the Kalkulator [expression syntax](#) (with functions, [variables](#) and all). In case of syntax or evaluation errors, the appropriate diagnostics will be provided.

(In other words, any entry field becomes by itself a mini-Kalkulator. In order to compute a value and store it in a variable it is not necessary to go via the [Expression Line](#): you may just click on [Put] in the [Variable Panel](#) and type the expression into the data entry box!)

If the dialog box stays open after the entered expression has been evaluated (this happens in some cases), the entry field will show just its value, not the expression itself. The display is limited to twelve decimal digits or so, but as long as the displayed text remains unmodified, the value is remembered with full Kalkulator accuracy. Once you type anything into the field, its contents will be re-parsed and evaluated again. (The same is true when the dialog is shown the next time, showing the previously entered values.)

The fields expecting integer (as opposed to floating-point) values do not accept expressions.

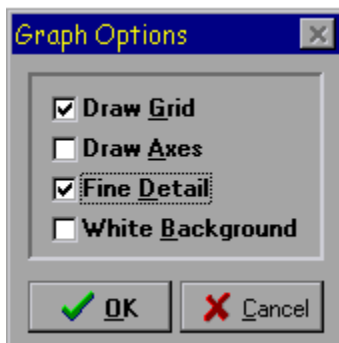
### See also:

[Expression syntax](#)

## Graph options

The [Options|Graph] menu entry brings up a dialog box allowing you to set the following:

- Grid -- with this option enabled, a reasonably spaced grid will be drawn in the background
- Axes -- (yes, that's plural of "axis"!), solid lines will be drawn for  $x=0$  and  $y=0$
- White Background -- on some monitors graphs may look better this way; this is also the setting you may use when copying the a graph to the Windows clipboard to use it in another program.
- Fine Detail -- this tells Kalkulator to be more cautious adjusting the step of the plotting process, so that functions with discontinuities, sharp peaks and dips can be drawn more accurately. For most purposes you may leave this setting at "off", unless you do not mind waiting a few seconds longer for the drawing to complete.



You can always invent a function to fool any graphing program. Kalkulator, however, is more difficult to fool than most. It adjusts the plotting step to the local line curvature and ignores non-computable points (singularities), still trying to approach them as close as reasonably possible.

To compare Kalkulator against any other function graphing program you have, try plotting  $\sin(1/t)$  for  $t$  from -1 to 1 with the Fine Detail setting at "on". Another good example is "int t".

## Numerical operations

These operations can be performed on a single-argument function  $f(t)$ , defined by the current expression, displayed in the Expression Line. The dummy variable  $t$  has to be used as the function argument.

Clicking on [Integ], [Deriv], [Sum] or [Zero] in the Bottom Line will first bring a dialog box to define the parameters of the operation (e.g. argument limits, required accuracy). When these parameters have been entered, the operation is carried out, and upon its completion the result is stored in the Temporary Variable,  $t$ .

The computation can be interrupted while the red timer box is being displayed, by pressing any key (or clicking inside that box).

### See also:

[Computing integrals](#)

[Computing derivatives](#)

[Computing sums](#)

[Finding function zeros](#)

[Entering numerical values](#)



## Computing integrals

After clicking on [Integ], a dialog box prompts for the integral limits,  $t_1$  and  $t_2$ , and for the desired accuracy parameter,  $d$ .

The iterative computation process terminates when the difference between two consecutive approximations,  $h$ , is less than  $d \cdot (1 + \text{abs } P)$ , where  $P$  is the last iteration result. This means that for  $P \ll 1$  the value of  $d$  denotes an absolute accuracy, while for  $P \gg 1$  -- a relative one.

## Computing derivatives

After a click on [Deriv] you will be prompted for the value of  $t$ , at which the function derivative,  $df(t)/dt$ , is to be computed.

The other field in the query dialog is the initial step of the step-adjusting algorithm used here (this is not the denominator of the finite increment approximation!). For most of well-behaved functions it never needs adjustments.

Obviously, if you attempt to compute the derivative at a point where the function is not continuous, funny things may happen.

## Computing sums

After clicking on [Sum], a dialog will ask you for the start and end value,  $t_1$  and  $t_2$ , and for the step,  $dt$ . The sum will be computed as  $f(t_1)+f(t_1+dt)+\dots+f(t_2)$ .

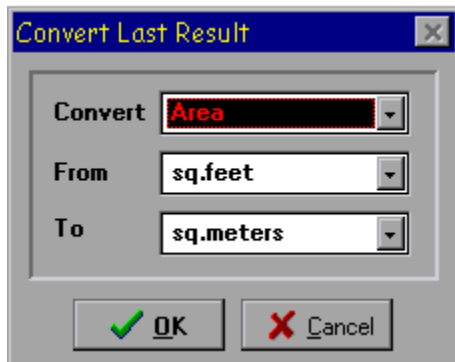
More precisely, if  $(t_2-t_1)/dt$  is not an integer, then the last value of  $t$  will be  $t_1+k*dt$ , where  $k$  is the largest integer for which  $t_1+k*dt < t_2+dt/2$ . This assures that any rounding errors are handled properly.

## Finding function zeros

This operation finds a value of  $t$ , for which  $f(t)=0$ . After a click on [Zero], a dialog box will ask for the interval  $(t_1,t_2)$ , bracketing the solution, and for the desired absolute accuracy,  $dt$ .

The interval  $(t_1,t_2)$  has to contain a single solution (or an odd number of solutions, in which case one of them will be found). If this is not the case, an error will be diagnosed.

## Unit conversion



The value shown in the Result Display can be converted from one kind of measurement units into another. This will affect not only the display, but also the actual value of the variable to which it refers.

Clicking on the [Convert] button will bring up the Unit Conversion Panel. It has three combo boxes with drop-down lists of items to choose from.

- The type of units, i.e. the physical quantity being converted (area, mass, etc.)
- The units to convert from -- only these relevant to the selected type will be available
- The units to convert to (ditto)

The actual conversion will take place when we exit the panel by clicking on [OK]. Note that the variable storing the result will also be affected by the conversion.

Note: although the temperature is not, strictly speaking, a physical quantity, temperature conversion has been included under pressure from disgruntled program users.

## Statistical operations

These operations are performed on data stored in the Data Buffer. The buffer may be in one of two modes: X (single data values) or XY (value pairs). The current mode is shown as [X] or [XY] in a button in the Bottom Row. Clicking on that button toggles the mode.

### Topics:

[Data Editor](#)

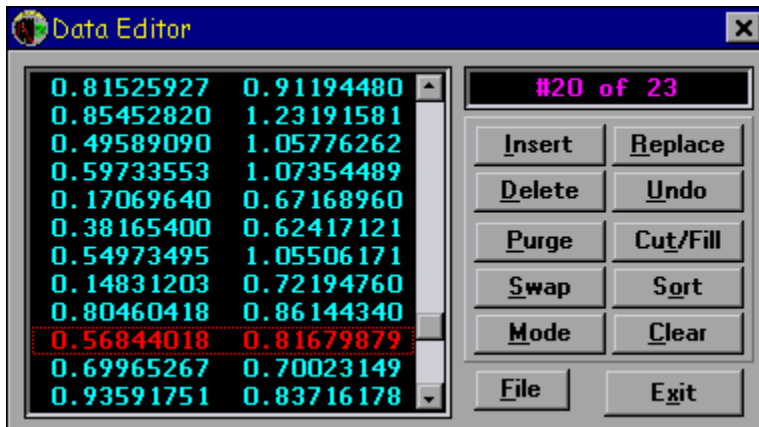
[Computing data points](#)

[Population parameters](#)

[Drawing a histogram](#)

[Polynomial regression](#)

## Data Editor



From here you can manipulate the contents of the Data Buffer.

Click on the buttons in the picture to see what operations do they perform.

Use Up, Down, PgUp and PgDn keys to move up or down the list.

Instead of entering the data manually, you can also use the current expression to compute the values of points in the X or Y column. This feature is accessed via the [Eval] button in the Bottom Row of the Main Panel.

The data in the buffer is saved together with the program status when you exit Kalkulator, to be automatically read back when you run it the next time. The registered version also allows to save the data to a separate text (ASCII) file and to load it from there.

### See also:

[Computing data points](#)

## Computing data points

The expression in the Expression Line can be used to compute new values for the points in the Data Buffer. To do it, use [Eval] in the Bottom Line.

A pop-up menu will give you a choice between computing all x-values or only those tagged as bad data; in the XY mode there will be also these options for y-values.

Kalkulator will compute the expression value and assign it to every data point as chosen above.

All this would not have much use, if not for the fact that the expression defining the computed values may refer (via fetch functions x# and y# ) to the existing data points, and to the index, denoted with the dummy variable t, of the point being computed. Thus, for example, the expression

$$y\#t + .5x\#t$$

used to compute all y values, will replace every y in the buffer with its previous value incremented by half of the value of the corresponding x. In other words, this is a full-fledged column arithmetic.

Another common use of this feature would be to replace all bad data values with a given constant, e.g., zero.

For some purposes (not only educational) it may come handy to use the random number generators in computing the data points in the buffer.



## Population parameters

Clicking on [Param] in the Bottom Row will compute the parameters of the points in the Data Buffer:

- nn -- the number of data points
- mx, my -- mean values of x and y (if applicable)
- sx, sy -- standard deviations
- vx, vy -- variances
- x1, x2, y1, y2 -- the minimum and maximum values
- cv -- the covariance (XY mode only)

Parameter values are shown in the Parameter Page of the Variable Panel. Fresh values are displayed in green; when the data in the buffer changes, the color will change to olive.

The parameters will not be automatically recomputed when the buffer contents changes; this has to be done explicitly again. This means that old parameter values can still be used after data points have been modified (for example, you can use them in computation of new data values).

Parameter symbols can be used in expressions like variables. Clicking on a parameter value in the scrolling list will enter its symbol into the Expression Line.

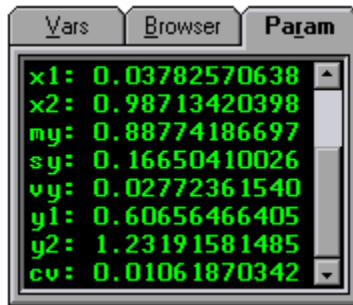
### See also:

[Parameter Page](#)

[Polynomial regression](#)

[Variables](#)

## The Parameter Page



### See also:

[Population parameters](#)

[Polynomial regression](#)

## Polynomial regression

The [Regr] button in the Bottom Row is used to compute polynomial regression coefficients for the XY data points in the Data Buffer, i.e. to find the best least-squares polynomial fit to the data.

You can choose the polynomial degree from a pop-up menu (up to 2 in the trial version, up to 9 in registered copies of Kalkulator). This choice will be limited to values lower than the number of data points in the buffer. The menu has also an option to just draw the points, without computing the regression.

The computed regression coefficients will be shown in the Parameter Page of the Variable Panel. They are denoted as a0...a9 (depending on the polynomial degree) and can be used as components of expressions.

After the regression coefficients have been computed, you will be given an option to plot the points and the regression line, and then asked for a plot range.

The function poly can be used at any time to compute polynomial values for the most recently computed coefficients (if any) and for a given argument value.

### See also:

[Population parameters](#)

[Parameter Page](#)

[Variables](#)

[Fetch functions](#)

[Plotting functions](#)

## Variables

Variables can be used as elements of expressions and to store results of operations.

- There are 23 "regular" variables, from A to Z (excluding E, T and O). They are entered in upper case, although Kalkulator will recognize them anyway.
- t (lowercase!) is the Temporary Variable, used to store the last result sent to the Result Display (even if another variable was also used to store the result).
- When an expression is interpreted as a function (e.g., in plotting or in numerical operations), t is understood as the argument (dummy parameter) of that function.
- The symbolic constants, ¶ and é, behave in expressions exactly like variables, except that they cannot be assigned new values.
- The statistical parameters describing the points in the Data Buffer are also treated like read-only variables, i.e. once computed, they can be used as elements of expressions.

### See also:

[Accessing variables](#)

## Accessing variables

The buttons in the Variable Button Page can be used to enter variable names into the Expression Line. The same can be done by clicking on variable values shown in the Browser Page (except that variables without assigned values will not be available from there), or on values shown in the Parameter Page.

### Setting, fetching and clearing variable values

The Variable Button Page and the Browser Page contain some buttons used for accessing or setting variable values (with variable selected from a pop-up menu):

- Get -- copy a variable value into the Result Display (and into the Temporary Variable, t)
- Put -- store a given value (you will be prompted for it) in a variable
- Kill -- clear a variable (it will have no value assigned)

Additionally, clicking on the Result Display (or pressing Shift-Enter) allows to store the last result in a selected variable.

#### See also:

[Variable Button Page](#)

[Browser Page](#)

[Parameter Page](#)

## The Variable Button Page

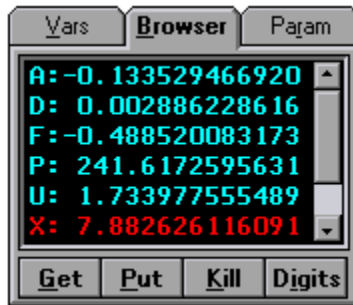


### See also:

[Variables](#)

[Accessing variables](#)

## The Browser Page



### See also:

[Variables](#)

[Accessing variables](#)

## Fetch functions



These functions allow you to access values from the Data Buffer in order to use them as components of expressions. Their buttons can be found in the Extra Page of the Function Panel.

The data fetch functions are x# and y# (lowercase characters!). The latter function is available only in the buffer XY mode; an attempt to use it in the X mode will cause an expression evaluation error.

An argument of a fetch function will be rounded to the nearest integer; its value can be from 1 to the number of points in the buffer. A value from outside the legal range will cause an expression evaluation error.

### Examples

```
(x#1+x#(i+1))/2  
x#max(t-1,1)
```

### Polynomial

There is one more function related to the data in the buffer: poly. It will use the current set of polynomial regression coefficients to compute a polynomial value for its argument. If the argument is x, then the polynomial is defined as  $a_0 + a_1*x + a_2*x^2 +$

...



## Drawing a histogram

Clicking on [Hist] in the Bottom Line (in the X mode only) will draw a histogram of the points in the Data Buffer. A dialog box will ask for the histogram limits and the number of bins (maximum: 50). You can use [Find] and [Round] to let Kalkulator do the work here.

## Constants

### Numeric constants

Constants are used as components of expressions. A constant consists of digits 0..9, an optional decimal point and an optional decimal exponent symbol, "E"; the exponent may be signed or not. For example, "1.5E-3" means the same as "0.0015". A zero preceding the decimal point can be skipped, as in ".0015".

A leading minus (or plus) sign is not treated as a part of a constant, but rather as an operator, although this is a purely academic distinction.

When an expression is reformatted before being computed, the constants in it may become reformatted as well, to a (possibly) simpler representation.

Important for European users: regardless of your regional settings, Kalkulator always uses a point, ".", as the decimal separator.

### Angular constants

These are entered with use of digits and the symbols of degrees, minutes and seconds, available in the [Digit Panel](#) (to enter the degree symbol from the keyboard, use Ctrl-d).

An angular constant used in an expression will be converted to a number, corresponding to the angular value in radians or degrees, depending on the current Kalkulator [angular mode](#).

### Symbolic constants

There are two predefined [symbolic constants](#): pi (3.14...) and Euler's e (2.71...). They are denoted with symbols  $\pi$  and  $e$ , respectively, available as buttons on the Digit Panel. To enter them from the keyboard, use the Ctrl-p and Ctrl-e [keystroke combinations](#).

#### See also:

[Physical constants](#)

## Physical constants

Kalkulator is aware of some elementary physical constants. Clicking on the [Const] button in the Operator Panel will bring a pop-up menu from which a constant can be selected. Its value will be stored in the Temporary Variable, t, and shown in the Result Display (from where it can be copied into any variable by a mouse click or by pressing Shift-Enter).

# Operators

## Available operators

- The four basic operators: +, -, ×, /
- The power operator: ^
- Factorial: !
- Square and cube: <sup>2</sup>, <sup>3</sup>
- Implied multiplication: · (a tiny dot, sometimes omitted at all)

When entered from the keyboard, some of the operators will require special keystroke combinations.

The multiplication operator, shown in the display as "×", is entered with the asterisk key, "\*", which is a commonly accepted procedure. (In the examples shown here we will be using the asterisk as well.)

## Operator priority

- High: ^, <sup>2</sup>, <sup>3</sup>, ! and ·
- Medium: ×, /
- Low: +, =

### See also:

[Expression syntax](#)

[Implied multiplication](#)

## Functions

### Arguments

Any arithmetic expression can be used as a function argument. Multiple arguments (in functions which expect them) are separated with commas.

Many of the functions accept just a single argument. If this argument is a number (constant), a variable, or another function call, then it does not need to be enclosed in parentheses. For example, "exp(3.8)-ln(X)" can be entered as "exp 3.8-ln X", but in "sqr(1+X)" the parentheses are necessary; without them the expression would be understood as "sqr(1)+X".

### Function groups:

[Standard functions](#)

[User functions](#)

[Conditional functions](#)

[Statistical distribution functions](#)

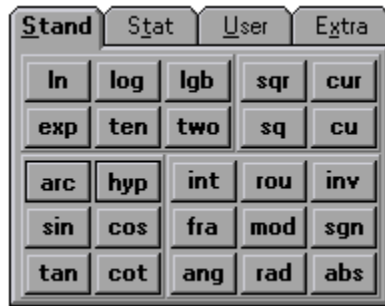
[Random number generators](#)

[Euler and related functions](#)

[Fetch functions](#)

[Feet, pounds, hours](#)

## Standard functions



These functions, accessible from the Standard Function Page, are usually considered "standard", whatever that means.

The following function groups can be identified here:

- Logarithms and exponents base e, 10 and 2: [ln](#), [log](#), [lgb](#), [exp](#), [ten](#), [two](#)
- Roots and powers: [sqr](#), [sq](#), [cur](#), [cu](#)
- Trigonometric and inverse: [sin](#), [cos](#), [tan](#), [cot](#), [asin](#), [acos](#), [atan](#), [acot](#)
- Hyperbolic and inverse: [sinh](#), [cosh](#), [tanh](#), [coth](#), [asinh](#), [acosh](#), [atanh](#), [acoth](#)
- Polar coordinate functions: [ang](#), [rad](#)
- Other functions: [int](#), [fra](#), [rou](#), [mod](#), [abs](#), [sgn](#), [inv](#)

Some other, "almost standard", functions can be found in the Extra Function Page:

- Smaller/greater choices: [min](#), [max](#), [mxa](#)
- Arithmetic, geometric and harmonic averages: [avg](#), [gav](#), [hav](#)

### See also:

[Angular mode](#)

[Extra Function Page](#)

## Angular mode

The current angular mode is shown as [Rad] or [Deg] in a button to the right of the Result Display. Clicking on that button will change the mode.

Depending on this mode, Kalkulator assumes that arguments to trigonometric functions are expressed in radians or degrees. It will also return results of inverse trigonometric functions in the same units.

Changing the angular mode to degrees can often lead to some confusion. I would recommend staying in the radian mode and using the angular constants explicitly. Then the calculation results can be displayed in the desired angular representation as needed.

### See also:

[Formatting the result](#)

## Formatting the result

Clicking on the Format button to the left of the Result Display offers (from a pop-up menu) a selection of various display formats.

- Default -- a reasonable compromise, the choice between floating and fixed point is made based on number readability. The values are shown with the full accuracy of 16 significant digits, and the trailing insignificant zeros are omitted.
- Fixed Point -- a fixed-point representation, with a selected number of digits after the decimal point (if the selected number would result in more than 16 significant digits, it will be appropriately reduced).
- Exponential -- a floating-point display, with a selected number of significant digits (up to 16) and at least two digits in the decimal exponent.

Any of these three settings becomes "permanent" -- it will also hold for all future calculation, until revoked. The other settings reformat just the current result.

- Fraction -- the value is shown as a whole part and a fraction, e.g., 3.125 becomes "3 1/8". The largest denominator is 1000 and it will be used if the value cannot be exactly represented with a smaller one.
- Deg-Min-Sec -- the value is shown in degrees, minutes and seconds of angle.
- Degrees -- the value is shown as degrees of angle (with decimal fraction, if any).
- Minutes -- minutes of angle.
- Seconds -- seconds of angle (format used by astronomers).

The four angular formats above are affected by the current angular mode setting, Rad or Deg, as the result is assumed to be an angle, originally expressed in radians or degrees, respectively.

- Feet & Inches -- the result is assumed to be in feet and it will be shown as feet, inches and fraction. For example, 1.24 will become "1 2 7/8" (one foot, two inches and 7/8 of an inch).
- Lbs & Ozs -- the result is treated as pounds and shown as pounds, ounces and fraction.

The maximum denominator value for inches or ounces is 64.

- Hrs-Min-Sec -- the result is treated as time in hours and displayed as e.g., 3:32:41

Obviously, depending on the displayed value and the format used, the display may be rounded. This never affects the actual result value, stored internally with 18 significant digits of accuracy.



## Extra Function Page



### See also:

[Conditional functions](#)

[Feet, pounds, hours](#)

[Fetch functions](#)

[Standard functions](#)

## Conditional functions



The buttons for these functions are in the Extra Page of the [Function Panel](#).

A conditional function accepts three arguments and, depending on the sign of the first argument, computes and returns the value of argument number two or three:

- [ifz](#) - "if zero", checks for the first argument being equal zero
- [ifp](#) - "if positive", checks for the first argument being positive ( $>0$ )
- [ifn](#) - "if negative", checks for the first argument being negative ( $<0$ )

The discarded argument is not evaluated at all; thus, for example, "ifz(X,1,sin X/X)" will be computed without an error for any value of X, including zero.

Note that functions for checking a "greater or equal" or "less or equal" condition are not necessary. Instead of invoking a hypothetical "if positive or zero" function `ifpz(x,a,b)` we may just use `ifn(x,b,a)`.

## Feet, pounds, hours



This group consists of three very similar functions, accessible from the Extra Page of the [Function Panel](#):

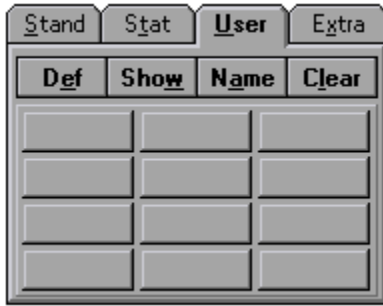
- [ft](#) converts feet and inches into feet
- [lb](#) converts pounds and ounces into pounds
- [hrs](#) converts hours, minutes and seconds into hours

### Example

This is how to compute the area of a rectangle 6 ft 7 in by 4 ft 3 5/8 in:

$$\text{ft}(6, 7) * \text{ft}(4, 3+5/8)$$

## User functions



Enter the function formula into the Expression Line, using  $t$  as the function argument, then click on [Def] in the User Page of the Function Panel. Choose a function slot from the pop-up menu, then enter the function name. This name, starting from a letter, has to be different than any function name already used.

If no syntax error is detected, the function will now be defined and its name will be shown in one of the user function buttons.

### Nesting user functions

One user function may call another, but it cannot refer to itself, directly or indirectly. A check for that is made when an expression using these functions is evaluated.

### Renaming and undefining user functions

This is done by clicking on [Name] and [Clear] in the User Page. You will not be allowed to rename or delete a user function if others are referring to it (you have to undefine those first).

Any references in expressions to a deleted or renamed function will now become invalid.

### Displaying and modifying user functions

Clicking on [Show] in the User Page will display the function definition in the Expression Line. The function can now be modified and defined again.

## Statistical distribution functions



Each of these functions can be identified by specifying two features: the distribution and the function type.

The top row buttons of the Statistical Function Page determine the type of the distribution function entered by the subsequent button clicks.

When the function type has been chosen, clicking on one of the eight buttons below the icons selects the distribution.

### Accuracy

All distribution functions are usually computed with at least sixteen good decimal digits. (In a few cases, near distribution tails, this cannot be achieved within the 19-digit internal accuracy; any of these exceptions are reasonable and justified.)

### For detailed information, see:

[Types of distribution functions](#)

[Supported distributions](#)

[Glossary of distribution functions](#)

[Fractional arguments to distribution functions](#)

## Types of distribution functions

The buttons in the top row of the Statistical Function Page determine the type of the distribution function entered by the subsequent button clicks.

The first four have their respective types shown by icons accompanied with a single letter. This letter denotes the function prefix as it will be shown in the Expression Line (and as it should be entered, in lowercase, from the keyboard). Our notation is similar to the one used in the Handbook of Mathematical Functions:

- the [F] button -- a distribution density (or probability)
- the [P] button -- a left distribution integral
- the [Q] button -- a right distribution integral
- the [A] button -- a central distribution integral (symmetric distributions only)

For the discrete distributions (Poisson and binomial) the term "distribution" density" should be replaced with "probability" and "integral" with "sum".

The rightmost button, [I], used together with one of the discussed above (P, Q or A), will denote an inverse integral distribution.

### See also:

[Supported distributions](#)

## Supported distributions

The following statistical distributions are supported by Kalkulator. The three-letter distribution abbreviations listed below, are additionally prefixed with one or two letters denoting the distribution function type.

- gau -- the Normal (Gaussian) distribution
- chi -- the chi-square distribution
- stu -- the t (Student) distribution
- sne -- the F (Snedecor) distribution
- poi -- the Poisson distribution
- bin -- the binomial distribution
- gam -- the gamma distribution
- bet -- the beta distribution

### See also:

[Types of distribution functions](#)

## Glossary of distribution functions

Here is the complete list of distribution functions provided by Kalkulator:

- Distribution density (or probability) functions: [fgau](#), [fchi](#), [fstu](#), [fsne](#), [fgam](#), [fbet](#), [fpoi](#), [fbin](#)
- Left distribution integral functions: [pgau](#), [pchi](#), [pstu](#), [psne](#), [pgam](#), [pbet](#), [ppoi](#), [pbin](#)
- Right distribution integral functions: [qgau](#), [qchi](#), [qstu](#), [qsne](#), [qgam](#), [qbet](#), [qpoi](#), [qbin](#)
- Central distribution integral functions: [agau](#), [astu](#)
- Left inverse integral distribution functions: [ipgau](#), [ipchi](#), [ipstu](#), [ipsne](#), [ipgam](#), [ipbet](#), [ippi](#), [ipbin](#)
- Right inverse integral distribution functions: [iqgau](#), [iqchi](#), [iqstu](#), [iqsne](#), [iqgam](#), [iqbet](#), [iqpoi](#), [iqbin](#)
- Central inverse integral distribution functions: [iagau](#), [iastu](#)

### See also:

[Types of distribution functions](#)

[Supported distributions](#)



## **Fractional arguments to distribution functions**

While most of the distributions deal with random variates in the real domain, the binomial and Poisson distributions are usually considered for discrete (integer) arguments. Nevertheless, Kalkulator will compute these distribution functions also for fractional values, using the same formulae.

Similarly, some of the distribution parameters are in most applications integer (e.g., degrees of freedom). Kalkulator will, however, accept non-integer distribution parameters, and use the same formulae as in the integer cases.

(There are statistical methods using fractional degrees of freedom: e.g., the test for equality of two sample means without assumption of equal variances.)

## Random number generators



These functions return randomly (strictly speaking: pseudorandomly) a different result every time they are used. The randomness is organized, however: for each generator its results are sampled from a distribution defined by the generator function and its arguments.

Three generators are provided, accessed with buttons in the Statistical Function Page:

- ran -- sampling from a uniform distribution
- rgau -- sampling from a Normal (Gaussian) distribution
- rpoi -- sampling from a Poisson distribution

### Reseeding the generators

Every time you run Kalkulator, the sequence of basic pseudorandom numbers used in the generators described above starts from the same seed. This means, that if in two different sessions with the program you repeat the same sequence of generator function calls, the results will be identical.

This is not always desirable. This is why the registered version of Kalkulator allows you to initialize (reseed) the generators with any seed value of your choice. To do it, use the [Options!Reseed generators] menu entry.

## Euler and related functions



These functions are grouped in the bottom row of the Statistical Page of the Function Panel:

- gam is the Euler Gamma function, related to the factorial operator:  $x! = \text{gam}(x+1)$ .
- bet is the Euler Beta function
- lgm is the logarithm of Gamma (handles larger arguments)
- com is the combination function

Both gam and bet are complete Euler functions; the incomplete ones can be computed with use of the Gamma and Beta integral distribution functions, pgam and pbet.

## Keyboard

In the Main Panel, most of the keyboard input goes into the Expression Line: all characters, Left and Right arrow keys, Home and End, Delete and Backspace. (Characters illegal in expressions will be just ignored, and characters without standard keyboard equivalents can be entered with use of special keystroke combinations.)

Additionally, all menu entries and many buttons have keyboard shortcuts defined in a standard Windows way: Alt + the underscored letter key.

To allow for using Kalkulator entirely without a mouse, some Main Panel operations have been assigned to keyboard shortcuts which are not obvious at a glance:

- Up and Down arrows -- switch to the previous/next expression
- Page Up, Page Down -- scroll through the values displayed in the Variable Panel (if the Browser or Parameters page is shown)
- Alt-Del or Escape -- clear the Expression Line (storing the expression in a buffer)
- Alt-Ins -- copy the buffer into the current line. If the line is not empty, it will be stored in the buffer (so that, effectively, a swap takes place).
- Enter -- compute the current expression, storing the result in the Temporary Variable, t.
- Ctrl-Enter -- check the current expression syntax, without evaluating it
- Shift-Enter -- copy the value from Result Display into a variable (chosen from a pop-up menu)
- Alt-Left, Alt-Right -- highlight the previous/next button in the Bottom Line
- Alt-Enter is equivalent to clicking on the highlighted Bottom Line button
- F1 -- call Help (Table of Contents)
- F10 -- show the Main Popup Menu
- F12 -- copy the Result Display into the Windows clipboard, so that it can be pasted into any Windows application supporting clipboard operations

When the Graph Canvas is visible, the Enter and Escape keys change their meanings: they are used to hide the canvas. (Enter does not make sense with the Result Display hidden, and Alt-Del has to be used instead of Escape.) At the same time, the F12 key will copy the Graph Canvas to the Windows clipboard.

The clipboard functions of Shift-Del (cut), Ctrl-Ins (copy) and Shift-Ins (paste) as defined in the standard Windows environment, can also be used in Kalkulator, applied to highlighted fragments of the Expression Line.

## Menu operations

These operations are accessible from the Main Menu and from the Main Pop-up Menu.

### File

- Clear -- clears everything: expressions, variables, user defined functions, data buffer, graph canvas.
- Exit -- exits from the program. There will be a warning and a choice between saving or not saving the program status to a disk file. The saved status is automatically read next time you start the program, so that you can continue the previous work.

The registered version has two more entries here:

- Load Status -- loads the Kalkulator status (settings, expressions, user functions, variables, data buffer, graph canvas) from a .KAL file, created on the last exit from the program or by explicit use of [File | Save].
- Save Status -- saves the Kalkulator status (see above) to a .KAL file.

In the non-registered version a KALKUL.KAL file is saved on exit and read when the program is executed again, but you cannot have multiple \*.KAL files.

### Options

- Panel Options -- shows the dialog to change some aspects of the Main Panel appearance and behavior.
- Graph Options -- displays the dialog to change the graph settings.
- Reseed Generators -- allows to reseed the random number generators.

There is also the usual Help menu.

### See also:

[Panel options](#)

[Graph options](#)

## Questions and Answers

These are questions submitted by the users of pre-release versions of Kalkulator. This section will be updated as new questions arrive.

[Does Kalkulator work under Windows 95?](#)

[What about the Pentium bug?](#)

[How do I uninstall Kalkulator?](#)

[How do I change colors?](#)

[The button fonts look bad on my screen. What's wrong?](#)

[Can I run two instances of Kalkulator at a time?](#)

[What are you planning for the next version?](#)

## **How do I uninstall Kalkulator?**

Just delete the Kalkulator directory with all the files in it. The program does not put any files anywhere else.

## **How do I change colors?**

You don't. The Kalkulator panel colors are not changed from within the program; they are bound to your desktop colors, which can be changed from the Windows Control Panel.

The graph colors, on the other hand, are currently hardwired to some preset values. If there is much of an uproar about this, I may provide a color-setting option in one of the next releases. Let me know how important you think it is.



**The button fonts look bad on my screen. What's wrong?**

This means that the Kalkulator fonts have not been loaded. The program should have complained about this (error message). Make sure that the files KALKULM.FON, KALKULP.FON and KALKULT.FON are in the Kalkulator home directory.

If no error message about missing .FON files has been shown, then obviously Windows 3.1 is running out of resources. Close as many applications as you can, then restart Kalkulator. This problem should not occur under Windows 95.

## **Multiple program instances**

No. When you attempt to run another instance of Kalkulator while the program is already running, this will be detected and the original instance will pop to the top instead.

### **What are you planning for the next version?**

The next major upgrade of Kalkulator will do linear algebra (including matrix inversion and systems of linear equations); it may also include a separate binary, octal and hexadecimal calculator panel. It is being expected late Spring, 1996. In the meantime Kalkulator may see some updates, with minor improvements and, possibly, bug fixes.

## Credits and copyrights

Kalkulator is copyright © 1995 by J. Andrzej Wrotniak. All rights reserved.

This software has been developed using Borland's Delphi and contains parts of code copyright © 1995 by Borland International.

The on-line help has been written with the Visual Help from WinWare, Inc.

The balloon help and LED display are from DTools, a freeware Delphi component library by Tim Noonan, Todd DeBerry and Rob Skyler, who deserve a sincere thank you.

The sound files supplied with Kalkulator have been created by Ted Tatman (thanks!) and are in public domain.

Kalkulator inherited many design ideas and a significant amount of code from two of my previous projects:

- from El\_Cal, the Elementary Calculator for the Atari ST series (1988-94) -- general concept, large parts of the user interface design
- from Ex, a Scientific Calculator for the HP LX Palmtop computers (1992-95) -- expression parser, most of the underlying math engine

As you can see, this is not quite a new program; under various disguises it has been under development for more than seven years.

[Important: read the enclosed disclaimer before using the program!](#)

## **Disclaimer**

This is what my lawyer told me to put here:

Kalkulator is licensed on the "as is" basis.

Although the author has extensively tested the software and reviewed the documentation, he makes no warrant or representation, explicit or implied, with respect to this package, its quality, performance, merchantability or fitness for a particular purpose, in any hardware or software environment.

In no event shall the author be liable for direct, indirect, special, incidental or consequential damages arising out of use, misuse or inability to use the software or documentation.

In addition, drinking excessive amounts of coffee when using this program may lead to insomnia, headaches and loss of facial hair.

## **Agnieszka**

Ten program jest dla Agnieszki.

Agnieszka jest nieduża i w okularach. Chyba, że się jej zdejmie, ale już jak zdjąć, to wszystko.

Z Agnieszka dobrze się śmiać, dokuczać jej trochę, waleś się po Pieninach albo w Nevadzie (szczególnie latem), kapać się z meduzami na Tangier Island, walczyć z krabem w bulce, mieć znajomego konia na Placu Zamkowym, bobrować w lodowce o trzeciej nad ranem, namydlać się w wannie (bo bardzo higieniczna jest ta Agnieszka!) i dużo więcej. Az musiałem napisać Kalkulator żeby to wszystko policzyć.

Czasem jest Agnieszka trochę smutna -- i wtedy trzeba coś szybko zrobić, bo tyle dobrego się marnuje.

Dobrze, że sobie te Agnieszke znalazłem. Przyda się.

## **The Bottom Line**

This is a row of buttons for accessing some of the advanced operations. The buttons at the left access the numerical operations, the buttons in the middle are used for statistical operations on the data buffer, and the buttons on the right - for function plotting.

The Bottom Line can be hidden by using the [Options|Advanced operations] menu entry and restored in the same way.

## **Clear Data Buffer**

The [Clear] button clears the buffer. The data is lost.



## Cut / Fill

The [Cut/Fill] button will set the number of data points in the buffer to a new value (within the buffer maximum size).

If the new number is smaller than the current one, the points at the end will be discarded. If it is larger, then new bad data points will be appended at the end.

### **Delete data point**

Clicking on [Delete] (or pressing the Delete key) will remove the selected data point from the buffer.

## Data Browser

You can view the contents of the Data Buffer here. In the XY mode, the left column shows x-values, the right one -- y-values. One of the data points will be always selected (highlighted).

## **Close editor window**

The [Exit] button (or pressing Escape) closes the Data Editor window and returns to the Main Panel.

You do not have to do this, as the Data Editor window is non-modal, i.e. it can remain open while the Main Panel is being accessed.

## **File operations**

This button, accessible only in the registered copies of Kalkulator, allows to save the buffer contents to a plain text (ASCII) file or to read the buffer from such file.

## **Data point index**

This shows the index of the selected data point, as well as the current number of points in the buffer.

## **Insert data point**

Clicking on the [Insert] button (or pressing the Insert key) will insert a new data point before the selected one.

## **Change stat mode**

The [Mode] button toggles between X and XY modes



## **Purge Data Buffer**

Clicking on [Purge] will delete all bad data points from the buffer.

## **Replace data point**

Clicking on [Replace] (or pressing Enter) will allow you to modify the value(s) of the selected data point.

## **Sort data points**

Clicking on [Sort] will sort the data points in the increasing sequence of x- or y-values.

## **Swap x and y columns**

The [Swap] button (available in the XY mode only) will swap the x- and y- values in the buffer.

### **Undelete data point**

Clicking on [Undo] (if available) will insert the last deleted data point back into the buffer.

## **The Digit Panel**

This panel, in addition to digit buttons, contains the decimal point and exponent, the constants: pi and Euler's e, and the button for accessing physical constants.

## **The Expression Line**

This is where the expression you are entering is being shown. The maximum expression length is 60 characters. The arrows at the right move between different expressions, and the current expression number is shown next to them.

## **The Function Panel**

This panel contains buttons used to enter function names into expressions. It has four pages, selectable with the tabs at the panel top:

[Standard Function Page](#)

[Statistical Function Page](#)

[User Function Page](#)

[Extra Function Page](#)



## Standard Function Page



**See also:**

[Standard functions](#)

## Statistical Function Page



### See also:

[Statistical distribution functions](#)

[Random number generators](#)

[Euler and related functions](#)

## User Function Page



### See also:

[User functions](#)

## The Menu Bar

This is a regular Windows menu bar. Some less frequently used operations are accessible from here, see Menu Operations.

You may hide the Menu Bar (and the Title Bar) by clicking on the [Options|Panel Options] menu entry and then unchecking the "Menu & Title Bar" option.

When the Menu Bar is hidden, its functions are accessible from the Main Pop-up Menu (press F10 or click the right mouse button in the Main Panel).

## **The Operator Panel**

This panel contains the buttons for common arithmetic operators. It also includes the parentheses and the comma (used to separate multiple function arguments), and the buttons for entering degrees, minutes and seconds of angle.

## **The Result Display**

Here the result of the last operation is shown. This result can be reformatted to a another representation or precision with use of the [Format] button, or converted between different measurement units with use of [Convert].

## **The Title Bar**

This is a regular Windows title bar. You may use it to move the Main Panel around. It has a System Menu button on the left and a Minimize button on the right.

To save the screen space, you may hide the Title Bar (and the Menu Bar) by clicking on the [Options | Panel Options] menu entry. Use the same entry (obviously, from the pop-up menu this time) to revert the change.

When the Title Bar is hidden, you may still drag the Main Panel around the screen by pressing and holding the mouse near the panel edge (the cursor changes to signal that you are there). The same applies to the Data Editor.

## The Variable Panel

This panel allows to access and review Kalkulator variables and statistical parameters of the data in the buffer. It has three pages, selectable with the tabs at the panel top:

[Variable Button Page](#)

[Browser Page](#)

[Parameter Page](#)



**a0...a9**

Polynomial regression coefficients

**abs**

abs x computes the absolute value of x.

**acos**

`acos x` computes an arc cosine of  $x$ .

**acosh**

`acosh x` computes an inverse hyperbolic cosine of  $x$ .

**acot**

`acot x` computes an arc cotangent of  $x$ .

**acoth**

`acoth x` computes an inverse hyperbolic cotangent of  $x$ .

**agau**

`agau(x,m,s)` returns the integral from  $m-(x-m)$  to  $x$  of the Gaussian (Normal) distribution with mean  $m$  and standard deviation  $s$ .

**Andrzej**

Pronounced "Un-Jay"



**ang**

`ang(x,y)` computes the polar angle of a Cartesian point  $(x,y)$ , i.e. the angle between the x-axis and the line from  $(0,0)$  to  $(x,y)$ .

**asin**

asin x computes an arc sine of x.

**asinh**

asinh x computes an inverse hyperbolic sine of x

**astu**

astu(x,r) returns the integral from -x to x of the t (Student) distribution with r degrees of freedom.

**atan**

atan x computes an arc tangent of x.

**atanh**

`atanh x` computes an inverse hyperbolic tangent of  $x$ .

**avg**

avg(x,y) computes the average of x and y:  $(x+y)/2$

## **Bad data**

When points in the Data Buffer are being computed, some of them may end up without values assigned, because of computation errors. These points will be flagged as bad data, denoted in the Data Editor as "???".

This is also how the new points, created when the number of points in the buffer is being changed to a larger value, are marked.



**bet**

`bet(x,y)` computes the Euler Beta function of  $x$  and  $y$ , defined as  $\text{Gamma}(x) \cdot \text{Gamma}(y) / \text{Gamma}(x+y)$

## Beta distribution

Density:  $f_{\text{bet}}(x,a,b)$  for  $0 \leq x \leq 1$  and parameters  $a > 0$  and  $b > 0$ .

## **Binomial distribution**

Probability:  $f_{\text{bin}}(x,n,p)$  for  $0 \leq x \leq n$  and  $0 < p < 1$  where  $n$  is the total number of tries and  $p$  is the probability of a success in a single try.

## Central distribution integral

The integral from  $m-x$  to  $x$  of  $f(x)$ , i.e. the probability that the random variate will assume a value as close to the distribution center,  $m$ , as  $x$  or closer. For  $m=0$  this means just between  $-x$  and  $x$ .

## **Chi-square distribution**

Density:  $f_{\text{chi}}(x,r)$ , for  $x \geq 0$  and  $r > 0$  degrees of freedom.

**com**

`com(n,k)` computes the combinations of  $n$  by  $k$ , i.e.  $n!/[k!(n-k)!]$ , where  $n$  and  $k$  do not have to be integers

**cos**

cos x computes a cosine of x

**cosh**

`cosh x` computes a hyperbolic cosine of  $x$ .



**cot**

`cot x` computes a cotangent of  $x$

**coth**

`coth x` computes a hyperbolic cotangent of  $x$ .

**cu**

cu x computes a cube (third power) of x.

**cur**

cur x computes a cubic root of x.

**CV**

The covariance of  $x$  and  $y$  in the Data Buffer

## **Data Buffer**

This is where Kalkulator stores the data points submitted to statistical operations. The capacity of the buffer is 128 single-value points (in the X mode) or 64 value pairs (XY mode); in the registered copies these limits are raised to 6400 and 3200, respectively.

## **Distribution density**

Sometimes referred to as differential distribution function:  $f(x) = dP(x)/dx$ .

## **El\_Cal**

The final Version 2.40 (a maintenance release) is available to all registered owners for free -- just send a self-addressed, stamped envelope and a formatted DD diskette, or \$5, to my address as shown in this Help.



**Ex**

The current version 1.4 (includes linear algebra) is available for \$30. Works on HP 95/100/200 LX. Evaluation copy can be downloaded from the CompuServe (go HPHAND, Lib.7) or other on-line services. One of the top CIS Palmtop Forum downloads.

**exp**

exp x computes Euler's e to the power of x

## **F (Snedecor) distribution**

Density:  $f_{\text{sne}}(x,r,s)$  for  $x \geq 0$  and for  $r$  and  $s$  degrees of freedom.

## Factorial

Usually defined, for integer arguments only, as  $n! = 1 * 2 * \dots * n$ . Kalkulator will compute it for non-integer arguments as well, using the Gamma function:  $n! = \text{gam}(n+1)$ . For integer arguments both definitions are equivalent.

**fbet**

`fbet(x,a,b)` returns the distribution density at  $x$  for a beta distribution with parameters  $a$  and  $b$ .

**fbin**

`fbin(x,n,q)` returns the distribution probability at  $x$  (integer) for a binomial distribution with parameters  $n$  (no.of tries) and  $q$  (single-try success probability).

**fchi**

`fchi(x,r)` returns the distribution density at  $x$  for a chi-square distribution with  $r$  degrees of freedom.

**fgam**

`fgam(x,a)` returns the distribution density at  $x$  for a gamma distribution with the parameter  $a$ .



## **fgau**

`fgau(x,m,s)` returns the distribution density at  $x$  for a Gaussian (Normal) distribution with mean  $m$  and standard deviation  $s$ .

**fpoi**

fpoi(x,m) returns the distribution probability at x (integer) for a Poisson distribution with a mean value m.

**fra**

`fra x` computes the fractional part of  $x$ . For negative  $x$ , `fra(x)` will also be negative.

**fsne**

`fsne(x,r,s)` returns the distribution density at  $x$  for an F (Snedecor) distribution with  $r$  and  $s$  degrees of freedom.

**fstu**

`fstu(x,r)` returns the distribution density at  $x$  for a  $t$  (Student) distributions with  $r$  degrees of freedom.

**ft**

ft(x,y) converts x feet and y inches into feet, as  $x+y/12$

## **gam**

gam x returns the Euler gamma function of x, for positive and non-integer negative arguments

## **Gamma distribution**

Density:  $f(x,a)$  for  $x \geq 0$  and for the parameter  $a > 0$ .



**gav**

`gav(x,y)` computes the geometric average of x and y:  $\sqrt{xy}$

**graph settings**

## **Handbook of Mathematical Functions**

Abramowitz and Stegun, National Bureau of Standards.

**hav**

hav(x,y) computes the harmonic average of x and y:  $1/(1/x+1/y)$

## hrs

hrs(x,y,z) converts x hours, y minutes and z seconds into hours, as  $x+y/60+z/3600$

## **iagau**

`iagau(p,m,s)` finds a value of  $x$ , for which  $agau(x,m,s)=p$ , for a Gaussian (Normal) distribution with mean value  $m$  and standard deviation  $s$ .

## **iastu**

`iastu(p,r)` finds a value of  $x$ , for which  $\text{astu}(x,r)=p$ , for a t (Student) distribution with  $r$  degrees of freedom.

**ifn**

ifn(x,y,z) returns y if  $x < 0$ , otherwise z.



**ifp**

ifp(x,y,z) returns y if  $x > 0$ , otherwise z.

**ifz**

ifz(x,y,z) returns y if x=0, otherwise z

## Implied multiplication

The "\*" operator can often be skipped, as in "3 sin 2X". Kalkulator will then insert the implied multiplication operator, denoting it with a "." if it makes the expression more readable -- here in one of two cases: "3·sin 2X". The implied multiplication has a higher priority than "\*".

**int**

int x computes the integer part of x (rounded towards zero).

## **integrals**

(Global), 0,<Numerical operations>

**inv**

inv x computes reciprocal (inverse) of x, i.e.  $1/x$

## **Inverse integral distribution**

More precisely: a function inverse to a cumulative (integral) distribution function. These functions compute a random variate value corresponding to a given probability integral.

For example, while the prefix "p" in the distribution function name means "find the integral probability for a given value of the random variate", the "ip" prefix means "find the random variate value for which the integral probability is such and such".

## **ipbet**

`ipbet(p,a,b)` finds a value of  $x$ , for which  $\text{pbet}(x,a,b)=p$ , for a beta distribution with parameters  $a$  and  $b$ .



## **ipbin**

`ipbin(p,n,q)` finds a value of  $x$ , for which  $\text{pbin}(x,n,q)=p$ , for a binomial distribution with parameters  $n$  and  $q$ .

## **ipchi**

`ipchi(p,r)` finds a value of  $x$ , for which  $pchi(x,r)=p$ , for a chi-square distribution with  $r$  degrees of freedom.

## **ipgam**

`ipgam(p,a)` finds a value of  $x$ , for which  $\text{pgam}(x,a)=p$ , for a gamma distribution with parameter  $a$ .

## **ipgau**

`ipgau(p,m,s)` finds a value of  $x$ , for which  $pgau(x,m,s)=p$ , for a Gaussian (Normal) distribution with mean value  $m$  and standard deviation  $s$ .

**ipoi**

`ipoi(p,m)` finds a value of  $x$ , for which  $\text{ppoi}(x,m)=p$ , for a Poisson distribution with mean value  $m$ .

## **ipsne**

`ipsne(p,r,s)` finds a value of  $x$ , for which  $psne(x,r,s)=p$ , for an F (Snedecor) distribution with  $r$  and  $s$  degrees of freedom.

## **ipstu**

`ipstu(p,r)` finds a value of  $x$ , for which  $\text{pstu}(x,r)=p$ , for a t (Student) distribution with  $r$  degrees of freedom.

## **iqbet**

`iqbet(p,a,b)` finds a value of  $x$ , for which  $qbet(x,a,b)=p$ , for a beta distribution with parameters  $a$  and  $b$ .



## **iqbin**

`iqbin(p,n,q)` finds a value of  $x$ , for which  $qbin(x,n,q)=p$ , for a binomial distribution with parameters  $n$  and  $q$ .

## **iqchi**

`iqchi(p,r)` finds a value of  $x$ , for which  $qchi(x,r)=p$ , for a chi-square distribution with  $t$  degrees of freedom.

## **iqgam**

`iqgam(p,a)` finds a value of  $x$ , for which  $qgam(x,a)=p$ , for a gamma distribution with parameter  $a$ .

## **iqgau**

`iqgau(p,m,s)` finds a value of  $x$ , for which  $qgau(x,m,s)=p$ , for a Gaussian (Normal) distribution with mean value  $m$  and standard deviation  $s$ .

**iqpoi**

iqpoi(p,m) finds a value of  $x$ , for which  $qpoi(x,m)=p$ , for a Poisson distribution with mean value  $m$ .

## **iqsne**

`iqsne(p,r,s)` finds a value of  $x$ , for which  $qsne(x,r,s)=p$ , for an F (Snedecor) distribution with  $r$  and  $s$  degrees of freedom.

## **iqstu**

`iqstu(p,r)` finds a value of  $x$ , for which  $qstu(x,r)=p$ , for a t (Student) distribution with  $r$  degrees of freedom.

## **Kalkulator**

About the name: this is not a "Kountry Kitchen" trying-to-be-cute stuff. Kalkulator (pronounced: "Cal-coo-lah-tor") is a legitimate Polish word; it can be translated into modern American English as "The Mother of All Calculators and There Ain't No Better One, Period, No Kidding".



## Keystroke combinations

Ctrl-d enters the degree symbol, °  
Ctrl-e enters the Euler's e, shown as  $e$   
Ctrl-p enters pi,  $\pi$   
Ctrl-2 enters the square,  $^2$   
Ctrl-3 enters the cube,  $^3$

**lb**

lb(x,y) converts x pounds and y ounces into pounds, as  $x+y/16$

## **Left distribution integral**

Also called a cumulative distribution function,  $P(x)$ . This is an integral up to  $x$  from  $f(x)$ , i.e. the probability that the random variate will assume a value less than or equal to  $x$ .

**lgb**

`lgb x` computes a binary (base 2) logarithm of `x`

**lgm**

lgm x computes a natural logarithm of Gamma of x

**lgm**

lgm x computes a natural logarithm of the Euler Gamma function of x

**ln**

ln x computes a natural (base e) logarithm of x

**log**

log x computes a decimal (base 10) logarithm of x



## **Main Pop-up Menu**

This menu pops up when you press the right mouse button over the Main Panel. It provides all functionality of the Kalkulator's Menu Bar, at the same time saving screen space.

**max**

`max(x,y)` computes the larger value of x and y.

**min**

`min(x,y)` computes the smaller value of x and y

**mod**

`mod(x,y)` computes the remainder of the division of  $x$  by  $y$ .

**mx**

Mean value of x in the Data Buffer

**mx**

`mx(x,y)` chooses the argument with the greater absolute value

**my**

Mean value of y in the Data Buffer

## **Nagging screen**

Kalkulator will not stop working after the trial license expires. It will just display, at the program startup, the screen with licensing information, urging the reluctant (or thrifty) user to make the choice between registering the program or removing it from his/her system.



**nn**

Number of points in the Data Buffer.

## **Normal (Gaussian) distribution**

Density:  $f_{\text{gau}}(x,m,s)$ , where  $m$  is the mean value and  $s>0$  -- the standard deviation.

## **pbet**

`pbet(x,a,b)` returns the integral from 0 to  $x$  of the Beta distribution with parameters  $a$  and  $b$ . It is equivalent to the incomplete Beta function `ibet(a,b,x)`.

## **pbin**

`pbin(x,n,q)` returns the probability sum between 0 and x (integer) for a binomial distribution with parameters n and q.

## **pchi**

`pchi(x,r)` returns the integral from 0 to  $x$  of the chi-square distribution with  $r$  degrees of freedom.

## **pgam**

`pgam(x,a)` returns the integral from 0 to  $x$  of the Gamma distribution with the distribution parameter  $a$ . It is equivalent to incomplete Gamma function `igam(a,x)`.

## **pgau**

`pgau(x,m,s)` returns the integral from minus infinity to  $x$  of the Gaussian (Normal) distribution with mean  $m$  and standard deviation  $s$ .

## Poisson distribution

Probability:  $f_{\text{poi}}(x,m)$  for  $x \geq 0$  and mean  $m > 0$ .



**poly**

poly(x) computes a polynomial of x, with coefficients given by data buffer parameters a0, a1... etc.

## Power operator

This operator, shown as "^", raises the first operand to the power of the second, as in "x^y" (x to y). If y is not integer, then x has to be greater than zero.

## **ppoi**

`ppoi(x,m)` returns the probability sum from 0 to  $x$  (integer) of the Poisson distribution with the mean value  $m$ .

## **psne**

`psne(x,r,s)` returns the integral from 0 to x of the F (Snedecor) distribution with r and s degrees of freedom

**pstu**

`pstu(x,r)` returns the integral from minus infinity to  $x$  of the  $t$  (Student) distribution with  $r$  degrees of freedom.

**qbet**

`qbet(x,a,b)` returns the integral from  $x$  to 1 of the Beta distribution with parameters  $a$  and  $b$ .

## **qbin**

`qbin(x,n,q)` returns the probability sum between  $x+1$  (integer) and  $n$ , for a binomial distribution with parameters  $n$  and  $q$ .

## **qchi**

`qchi(x,r)` returns the integral from  $x$  to infinity of the chi-square distribution with  $r$  degrees of freedom.



## **qgam**

`qgam(x,a)` returns the integral from  $x$  to infinity of the Gamma distribution with the distribution parameter  $a$ .

## **qgau**

`qgau(x,m,s)` returns the integral from  $x$  to infinity of the Gaussian (Normal) distribution with mean  $m$  and standard deviation  $s$ .

## **qpoi**

`qpoi(x,m)` returns the probability sum from  $x+1$  (integer) to infinity of the Poisson distribution with the mean value  $m$ .

## qsne

`qsne(x,r,s)` returns the integral from  $x$  to infinity of the F (Snedecor) distribution with  $r$  and  $s$  degrees of freedom.

**qstu**

`qstu(x,r)` returns the integral from  $x$  to infinity of the  $t$  (Student) distribution with  $r$  degrees of freedom.

**rad**

`rad(x,y)` computes the polar radius of the Cartesian point  $(x,y)$ , i.e. the distance from  $(0,0)$  to  $(x,y)$ .

**ran**

ran x returns a random value from a uniform distribution between 0 and x.

## **Regression coefficients**

Variables  $a_0, a_1 \dots a_9$  (depending on the degree), computed for the XY data buffer with use of the [Regr] button.



**rgau**

`rgau(m,s)` returns a random value from a Gaussian (Normal) distribution with mean  $m$  and standard deviation  $s$ .

### **Right distribution integral**

The integral of  $f(x)$  from  $x$  up, or the probability that the random variate will assume a value greater than  $x$ . Note that  $Q(x)=1-P(x)$ .

**rou**

$\text{rou}(x,y)$  rounds  $x$  to the nearest whole multiple of  $y$ .

**rpoi**

rpoi m returns a random value (integer) from a Poisson distribution with a mean value m.

**sgn**

sgn x returns the sign of x: -1 for negative, 1 for positive and 0 for x=0.

**sin**

sin x computes a sine of x

**sinh**

`sinh x` computes a hyperbolic sine of  $x$ .

## **Site licenses**

Site licenses are priced at \$100 plus \$15 per user (e.g., for 20 users  $\$100 + 20 * \$15 = \$400$ ). We will invoice institutions within the U.S. Obtaining a site license will allow your company to avoid problems in case of a software audit.



**sq**

sq x computes a square (second power) of x

**sqr**

sqr x computes a square root of x

**SX**

Standard deviation of x in the Data Buffer

**sy**

Standard deviation of y in the Data Buffer

## **t (Student) distribution**

Density:  $f_{\text{stu}}(x,r)$  for  $r$  degrees of freedom.

**tan**

tan x computes a tangent of x

**tanh**

$\tanh x$  computes a hyperbolic tangent of  $x$ .

## **Temporary Variable**

Denoted as t, this variable is used to store the result of the last operation, shown in the Result Display. Even if the result is explicitly stored in another variable (A..Z), it will be stored in t as well.



**ten**

ten x computes 10 to the power of x

**two**

two x computes two to the power of x

## **User functions**

An expression can be used to define a new function of a single argument,  $t$ . This new function can then be used in other expressions.

**VX**

Variance of x in the Data Buffer

**vy**

Variance of y in the Data Buffer

## **Wrotniak**

Ex-physicist, formerly specializing in computer Monte Carlo techniques applied to cosmic ray physics. Taught physics and computer science at University of Lodz, Poland. In the U.S. since 1984, makes his life as a software engineer. Currently the Chief Scientist at Aerospace Engineering & Research in MD, helping to bring the air traffic control software out of its 19th century state. One day may get a real job and a haircut.

**x#**

x#i returns the i-th x-value from the Data Buffer.

**x1**

The smallest value of x in the Data Buffer



**x2**

The largest value of x in the Data Buffer

**y#**

y#i returns the i-th y-value from the Data Buffer in the XY mode.

**y1**

The smallest value of y in the Data Buffer

**y2**

The largest value of y in the Data Buffer



