The LabPlot Handbook

Stefan Gerlach
Alexander Semke
Yuri Chornoivan
Garvit Khatri

Contents

Abstract

LabPlot is a program for two-dimensional function plotting and data analysis.

Chapter 1

Introduction

LabPlot is a KDE application for interactive graphing and analysis of scientific data. LabPlot provides an easy way to create, manage and edit plots.

Features:

- Project-based management of data
- Project-explorer for management and organization of created objects in different folders and sub-folders
- Spreadsheet with basic functionality for manual data entry or for generation of uniform and non-uniform random numbers
- Import of external ASCII-data into the project for further editing and visualization
- Export of spreadsheet to an ASCII-file
- Worksheet as the main parent object for plots, labels etc., supports different layouts and zooming functions
- Export of worksheet to different formats (pdf, eps, png and svg)
- Great variety of editing capabilities for properties of worksheet and its objects
- Cartesian plots, created either from imported or manually created data sets or via mathematical equation
- Definition of mathematical formulas is supported by syntax-highlighting and completion and by the list of thematicaly grouped mathematical and physical constants and functions
- Investigation of plotted data is supported by many zooming and navigation features
- Several analysis functions and methods for data reduction, differentiation, integration, interpolation, smoothing, (nonlinear) fitting, Fourier filter and Fourier transform
- Linear and non-linear fits to data, several fit-models are predefined and custom models with arbitrary number of parameters can be provided
- Supports many CAS backends like Maxima, Python, KAlgebra, Sage
- Nice Worksheet view for evaluating expressions
- Easy plugin based structure to add different Backends
- Plugin based assistant dialogs for common tasks (like integrating a function or entering a matrix)
- Datapicker for manual or (semi-)automatic data extraction from imported images containing plots and curves.

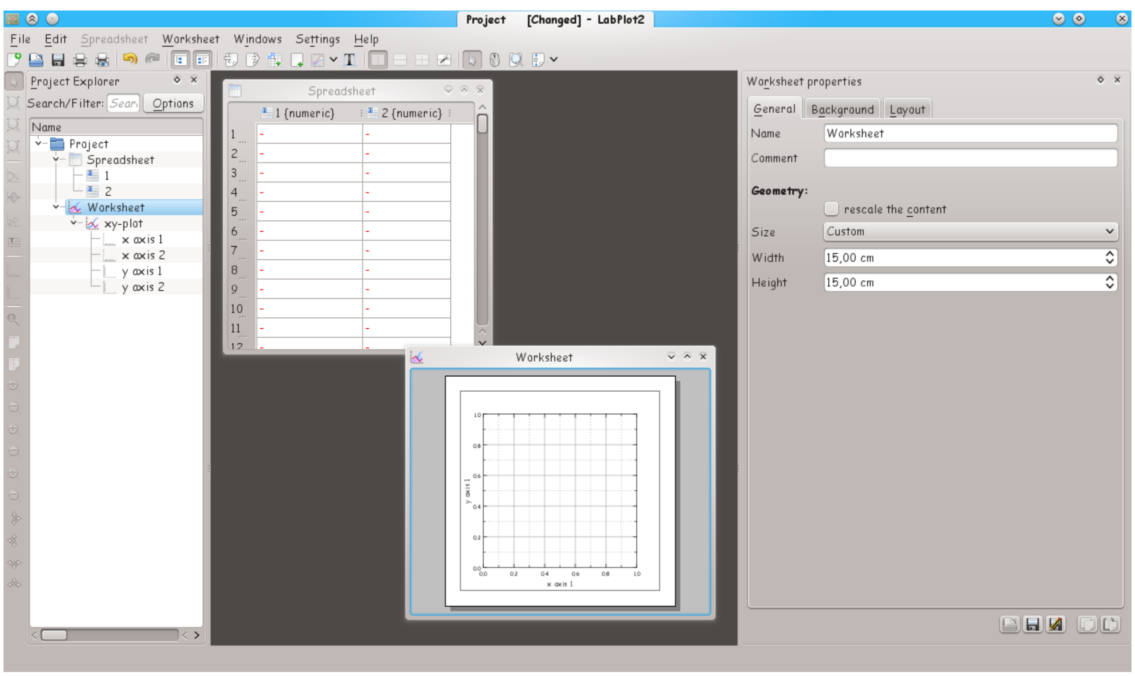
LabPlot can be found on its homepage at kde.org: https://labplot.kde.org/ .

Chapter 2

Using LabPlot

2.1 Interface Overview

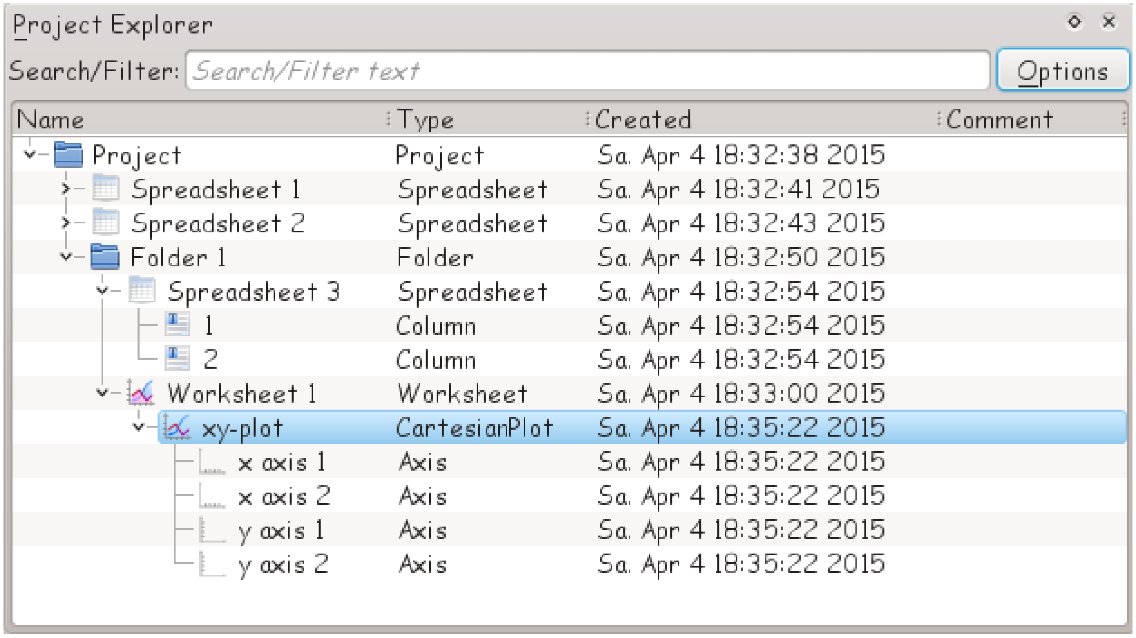
LabPlot follows the MDI (Multiple Document Interface) philosophy - all the created application objects are placed as sub-windows in the Main Area of the application window. The Project Explorer serves as the tool to create and organize those objects in a tree-like structure. The Properties Explorer is used to modify the properties of the currently selected object(s). Many functions are reachable via the main menu and via object specific toolbars and context menus. Additional information and application notifications are shown in the status bar.



2.2 Project Explorer

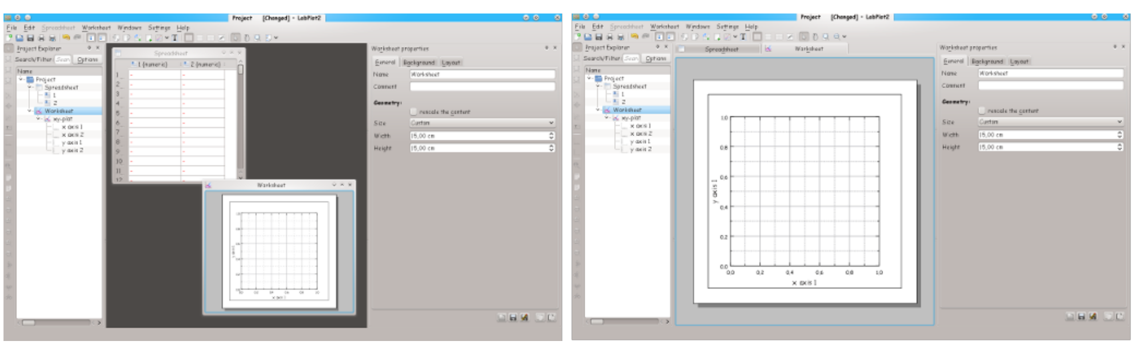
The Project Explorer is the main part of LabPlot aimed to handle its objects. Objects are organized in a tree-like structure representing the parent-child relations between the different objects. Folders and sub-folders can introduce additional grouping for the different objects.

Project explorer is a dockable window and can be placed at an arbitrary place. The user can determine which columns should be shown by selecting/deselecting the columns of interest in the context menu (right mouse button click on an empty place in the tree-view or its header). Furthermore, the list of shown objects can be reduced by providing a filter in the **Search/Filter** text field.



2.3 Main Area

Created objects having a view (like worksheet, spreadsheet etc.) are placed in the main area of the application. Depending on the current setting for the user interface, windows are placed either as independent and freely moveable sub-windows (interface ″Sub-window view″) or as tabs in a tabbed view (interface ″Tabbed view″).



When sub-windows are used, all windows of objects belonging to the currently selected folder only are shown. Alternatively, the visibility of windows can be extended to the currently selected folder and its sub-folders or to all windows in the project. This behaviour is controlled via the parameter ″Window visibility policy″ accessible via the context menu of the project explorer.

2.4 Properties Explorer

Properties explorer allows the user to modify the currently selected object in the project explorer. A great variety of object properties can be edited in undoable/redoable way. Editing of multiple objects of the same time is also possible.

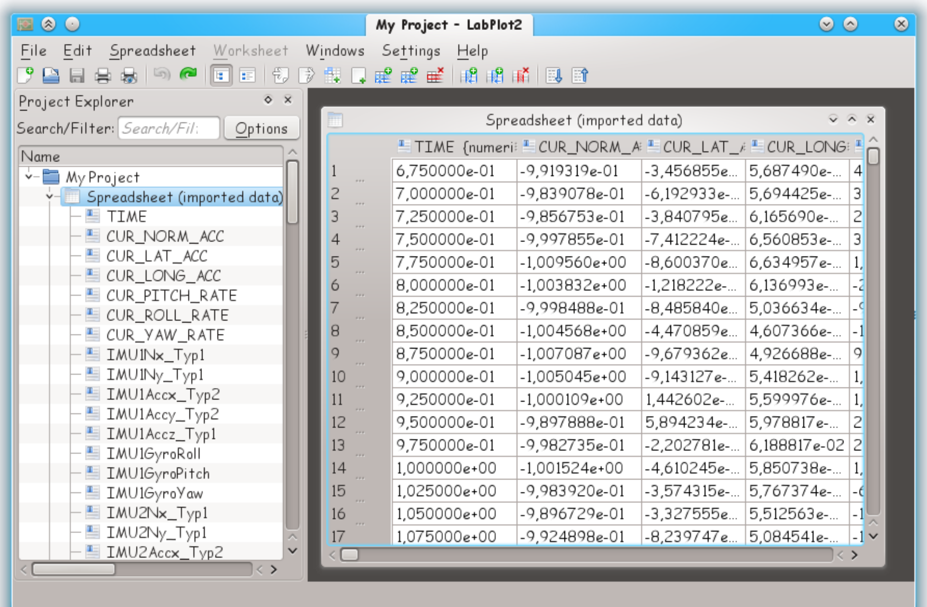
Properties explorer is a dockable window and can be placed at an arbitrary place.

2.5 Spreadsheet

The spreadsheet is the main part of LabPlot when working with data and consists of columns. Column is the basic data set in LabPlot used for plotting and data analysis. Every column of the spreadsheet is specified by its name and the type - numeric, text, month names, day names and date and time. Also, for each type different representation formats can be assigned like decimal or scientific format for numeric columns etc.

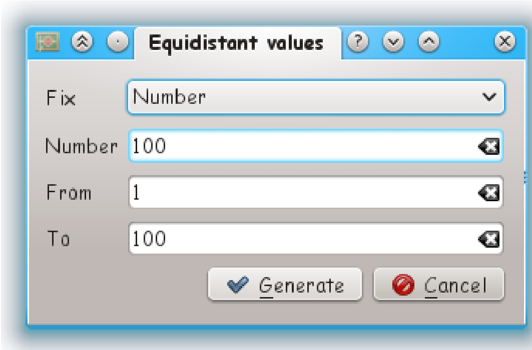
You can mask selected data points in the spreadsheet (**Selection** → **Mask Selection** from the spreadsheet cell context menu). Masked data is not plotted and is also excluded from data analysis functions like fitting etc. Alternatively, you can mask or drop values in a column (**Mask Values** or **Drop Values** from the column context menu) by specifying a range. When specifying which values to mask or to drop, several operators (“equal to”, “greater than”, “lesser than”, etc.) are available. These operations can help to hide or to remove some outliers in the data set prior to, e.g., performing a fit to this data set.

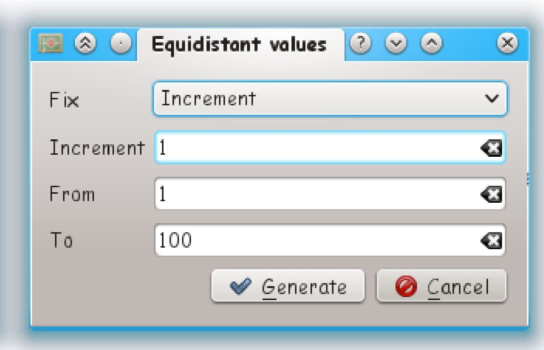
Any spreadsheet function can be reached via the context menu (right mouse button click). You can cut, copy and paste between spreadsheets, generate, normalize and sort data and finally make plots out of your data.



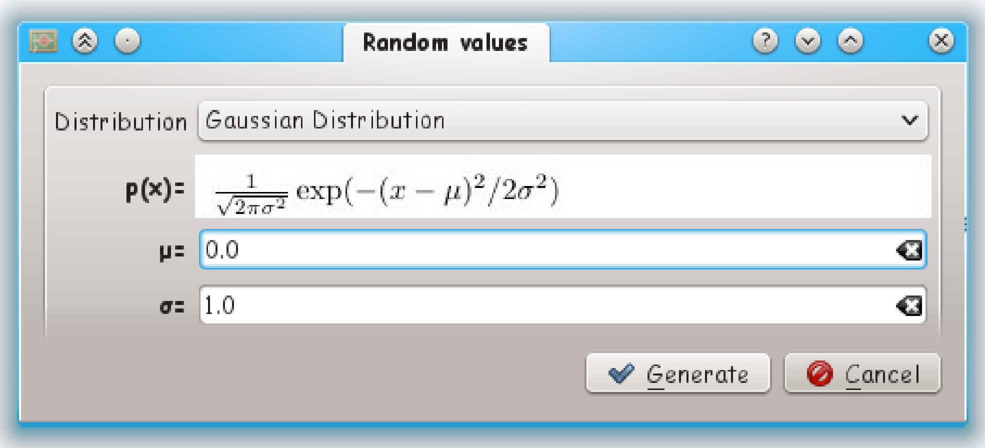
New data can be produced either by entering it manually in the spreadsheet or by generating the data according to a certain prescription. LabPlot provides 5 different methods to generate data, accessible via the context menu of the column:

- Row Numbers - values in the column are set according to its row number, this provide an easy way to quickly create an index.
- Const Values - values in the column are set to a constant value provided by the user.
- Equidistant values (for numeric columns only) - given the minimal and the maximal values, the equidistant values can be either generated by fixing the total number of values in that range or by fixing the increment (distance).



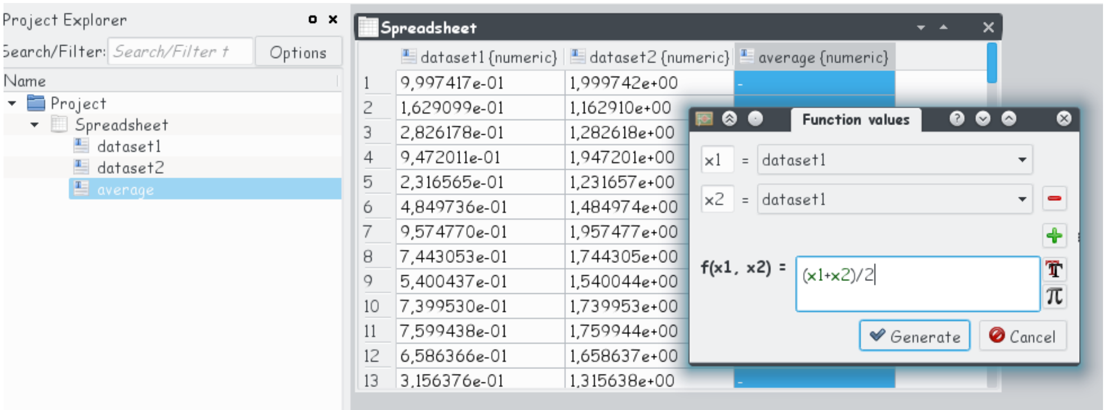


- Random values (for numeric columns only) - values are randomly generated according to the selected distribution. To generate uniformly distributed random numbers, select "Flat" distribution.



In the simplest cases a non-uniform distribution is calculated analytically from the uniform distribution of a random number generator by applying an appropriate transformation. More complicated distributions are created by the acceptance-rejection method, which compares the desired distribution against a distribution which is similar and known analytically.

- Function values (for numeric columns only) - values are generated according to a mathematical function provided by the user, a column (data set) containing the function arguments has to be provided. It is possible to define a multivariant function and to provide a data set (a column in a spreadsheet) for each of the variables. The corresponding dialog supports the creation of arbitrary number of variables.

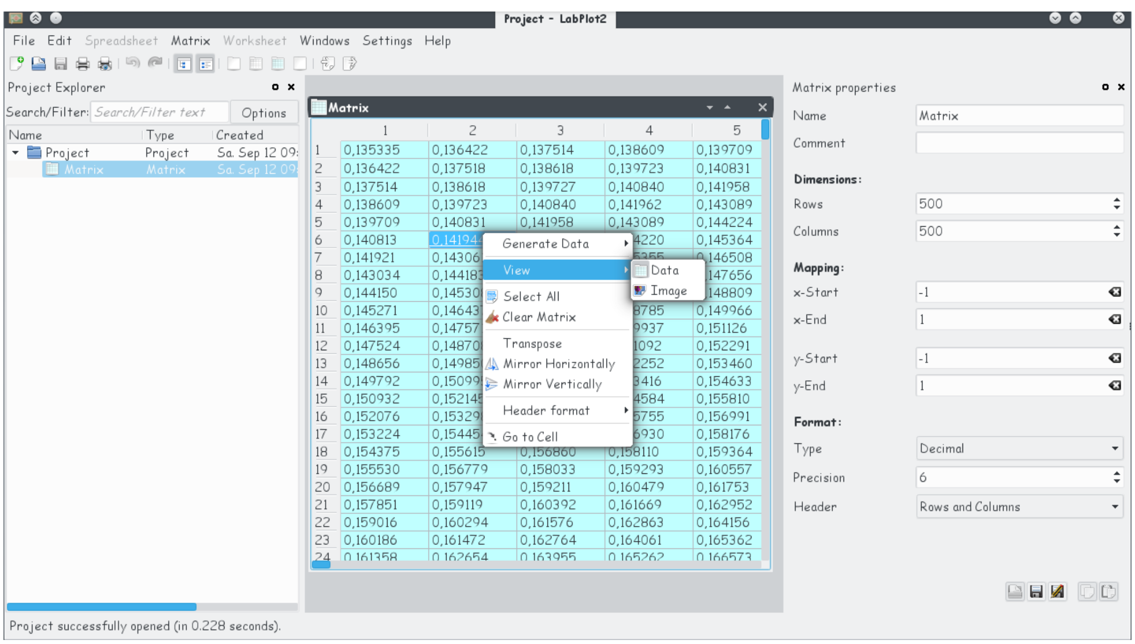


Already existing data can be imported into a spreadsheet from external files via the ”Import Data” dialog. Imported data will be stored in the project file. Changes on data, performed either in the spreadsheet or in the external file after the import, are not synchronized anymore.

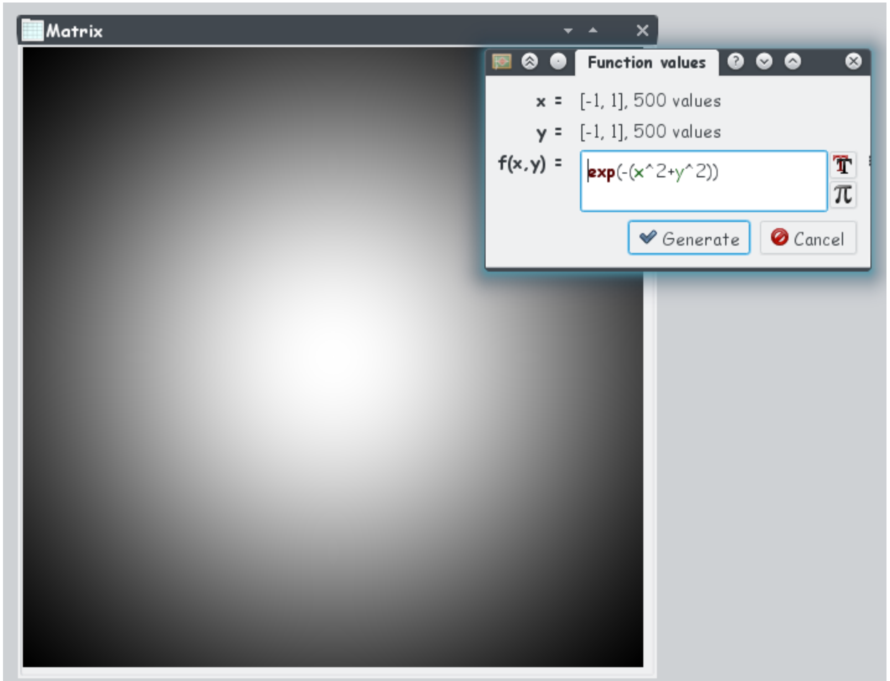
The data in the spreadsheet can be exported to an external file (see Export Dialog).

2.6 Matrix

Matrix is another container for matrix-like data. This container is presented like a table or, alternatively, as a two-dimensional greyscale image. The elements of such a table/matrix can be thought as being the Z-values, Z=Z(X,Y), with X and Y values being the row and column numbers, respectively. The transition from the row and column numbers to the logical coordinates is done via an explicit user-defined mapping of both representations.

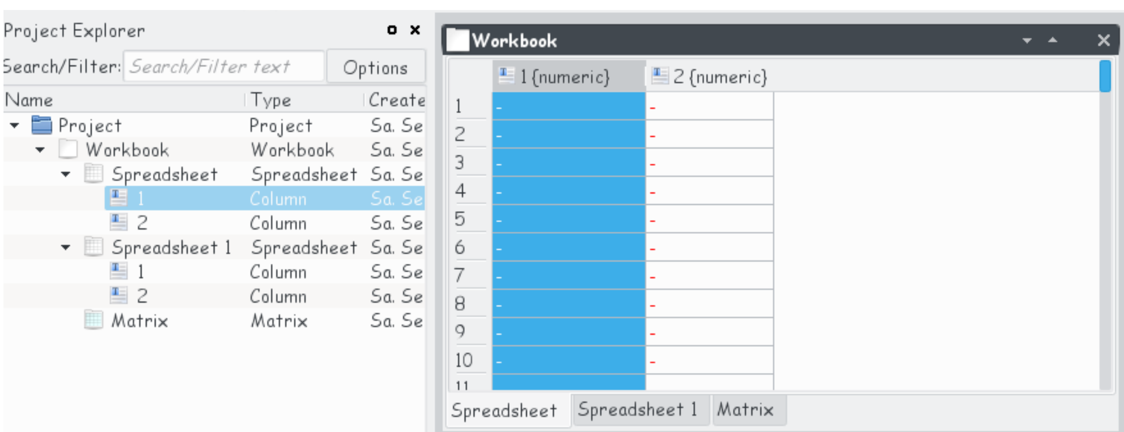


The matrix data can either be entered manually or via an import from an external file. Similar to the data generation for a column in a spreadsheet, the matrix can be filled with constant values or via a formula, too. The screenshot below shows the image view of a matrix together with the formula that was used to generate the matrix elements:



2.7 Workbook

Workbook helps the user to better organize and to group different data containers (Spreadsheet and Matrix). This object serves as the parent container for multiple Spreadsheet- and/or Matrix-objects and puts them together in a view with multiple tabs:



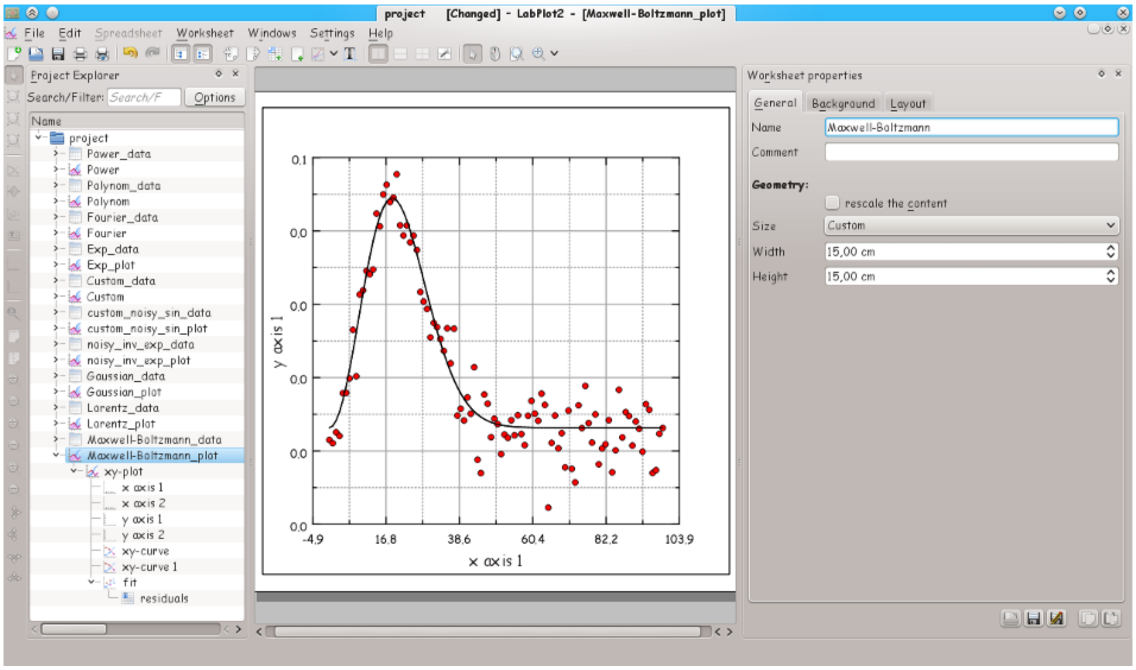
With folders it is already possible to bring some structure in the Project Explorer and to group together several related objects (spreadsheets with data stemming from text files of similar origin, red, green and blue values of an image imported into three different matrices, etc.). With Workbook the user has the possibility for another additional grouping.

2.8 Worksheet

The worksheet is, besides the data containers Spreadsheet and Matrix, another central part of the application and provides an area for showing and grouping together different kinds of worksheet objects - plots, labels etc.

Worksheets can either have a fixed size (a user defined size or one of the predefined sizes like A4, Letter etc.) or they can fill out the complete available area for the worksheet window. Multiple plots can be arranged on the worksheet in a vertical, horizontal or grid layouts.

Many properties of the worksheet like size, background colour and layout settings can be changed in the ″Worksheet properties″ pane.



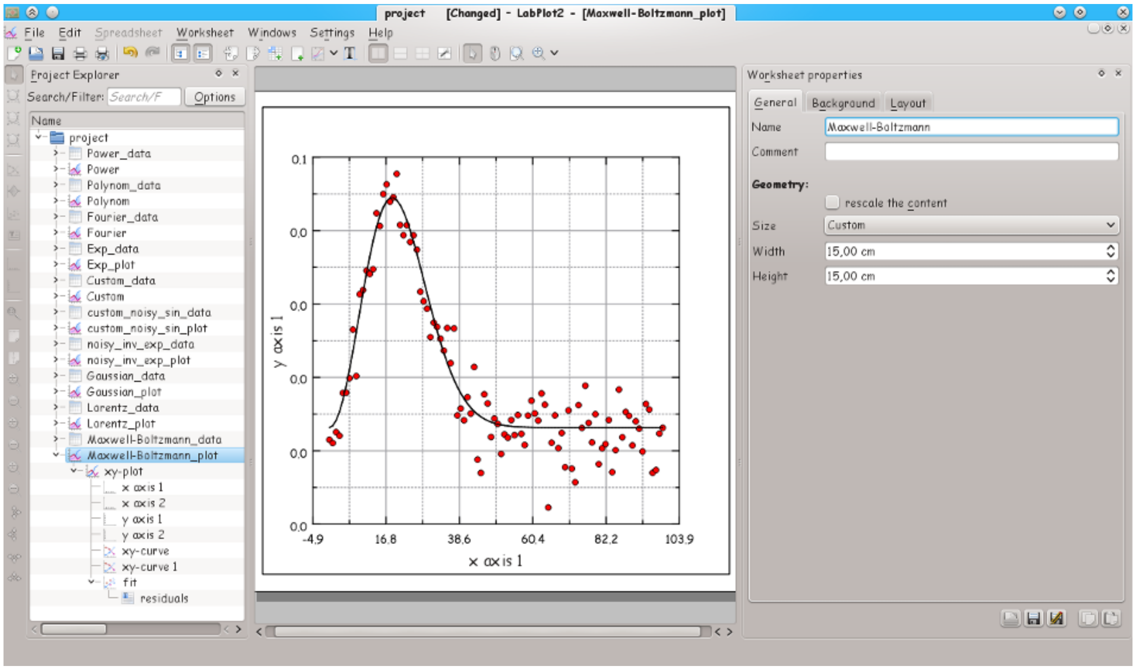
Different worksheet actions dealing with the creation of new objects, changing of the current mouse mode or zooming can be accessed via the toolbar, main menu or the context menu of the worksheet in the project explorer.

The results shown on the worksheet can be exported to different formats via the export dialog.

2.9 CAS Worksheet

The CAS worksheet is, besides the worksheet, the third central part of the application and provides an area to you use your favorite mathematical applications from within an elegant Worksheet Interface.

LabPlot offers you several choices for the backends you wish to use with it. The choice to make depends on what you want to achieve.



Currently the following backends are available:

Sage:

Sage is a free open-source mathematics software system licensed under the GPL. It combines the power of many existing open-source packages, within a common Python-based interface. See http://sagemath.org for more information.

Maxima:

Maxima is a system for the manipulation of symbolic and numeric expressions, including differentiation, integration, Taylor series, Laplace transforms, ordinary differential equations, systems of linear equations, polynomials, sets, lists, vectors, matrices, and tensors. Maxima yields high-precision numeric results by using exact fractions, arbitrary precision integers, and variable precision floating point numbers. Maxima can plot functions and data in two and three dimensions. See http://maxima.sourceforge.net for more information.

R:

R is a language and environment for statistical computing and graphics, similar to the S language and environment. It provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an open-source route to this. See http://www.r-project.org for more information.

KAlgebra:

KAlgebra is a MathML-based graph calculator, that ships with KDE Education project. See http://edu.kde.org/kalgebra/ for more information.

Qalculate!:

Qalculate! is not your regular software replication of the cheapest available calculator. Qalculate! aims to make full use of the superior interface, power and flexibility of modern computers. The center of attention in Qalculate! is the expression entry. Instead of entering each number in a mathematical expression separately, you can directly write the whole expression and later modify it. The interpretation of expressions is flexible and fault tolerant, and if you nevertheless do something wrong, Qalculate! will tell you so. Not fully solvable expressions are however not errors. Qalculate! will simplify as far as it can and answer with an expression. In addition to numbers and arithmetic operators, an expression may contain any combination of variables, units, and functions. See http://qalculate.sourceforge.net/ for more information.

Python2:

Python is a remarkably powerful dynamic programming language that is used in a wide variety of application domains. There are several Python packages to scientific programming.

Python is distributed under Python Software Foundation license (GPL compatible). See the official website for more information.

> NOTE
> This backend adds an additional item to the Cantor's main menu, **Package**. The only item of this menu is **Package** → **Import Package**. This item can be used to import Python packages to the worksheet.

> WARNING
> This backend supports Python 2 only.

Scilab:

Scilab is an free software, cross-platform numerical computational package and a high-level, numerically oriented programming language.

Scilab is distributed under CeCILL license (GPL compatible). See http://www.scilab.org/ for more information.

> WARNING
> You need Scilab version 5.5 or higher to be installed in your system to make this backend usable.

Octave:

GNU Octave is a high-level language, primarily intended for numerical computations. It provides a convenient command line interface for solving linear and nonlinear problems numerically, and for performing other numerical experiments using a language that is mostly compatible with MATLAB. See http://www.gnu.org/software/octave/ for more information.

Lua:

Lua is a fast and lightweight scripting language, with a simple procedural syntax. There are several libraries in Lua aimed at math and science.
See http://www.lua.org/ for more information.

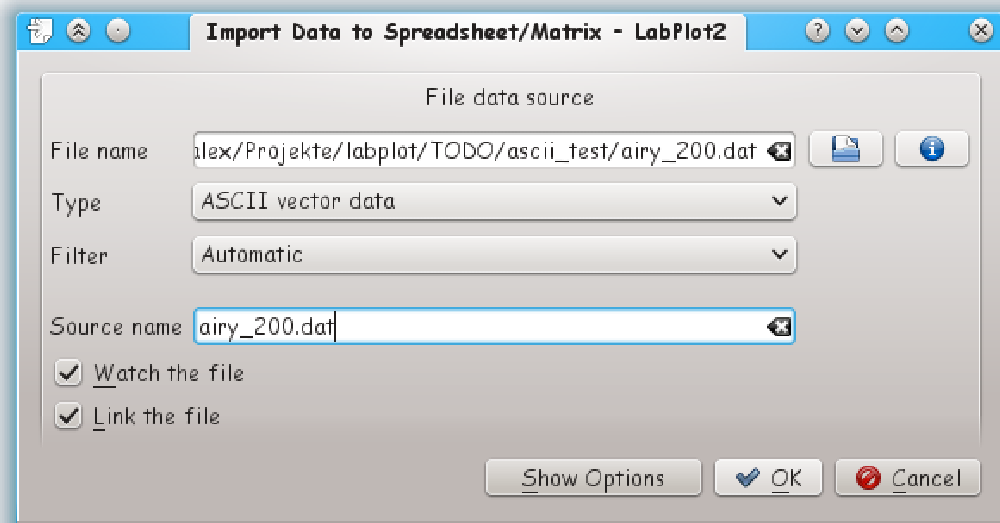
This backend supports luajit 2.

2.10 File Data Source

A file data source is very similar in spirit to a spreadsheet with imported data from an external file. The difference is that the imported data cannot be shown and edited in LabPlot after the import anymore. This can be sufficient e.g. if you only want to plot the data stemming from a calculation in an external program (and exported to an ASCII-file afterwards).

Since no spreadsheet has to be filled with the imported data, the import into a file data source is faster than into a spreadsheet which can be advantageously when dealing with big files.

It is possible to store the link to the external file in the project file only and not its content. Each time the project file is opened in LabPlot, the content is read from the external file again. Also, it is possible to let LabPlot watch the file for changes - the content of the file data source is updated if the external file was changed.



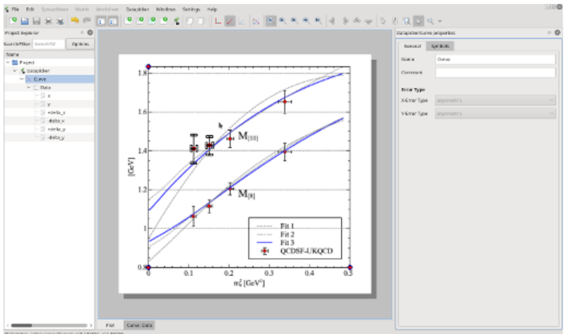
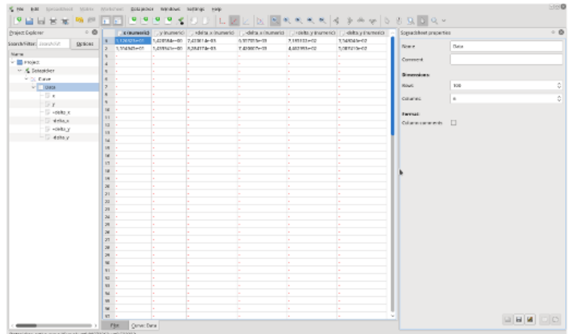
The additional options determining the import of the data are equivalent to those provided in Import Dialog.

2.11 Datapicker

Datapicker is a tool that allows you to easily extract data from image files. The process of extraction consists mainly out of the following steps:

- Import an image containing plots and curves where you want to read the data points from.
- Select the plot type (cartesian, polar, etc.).
- Select tree reference points and provide values for them. With the help of these points the logical coordinate system is determined.
- Create a new datapicker curve and set the type of the error bars.
- Switch to the mouse mode ″Set Curve Points″ and start selecting points on the imported image - the coordinates for the selected points are determined and added to the spreadsheet ″Data″.

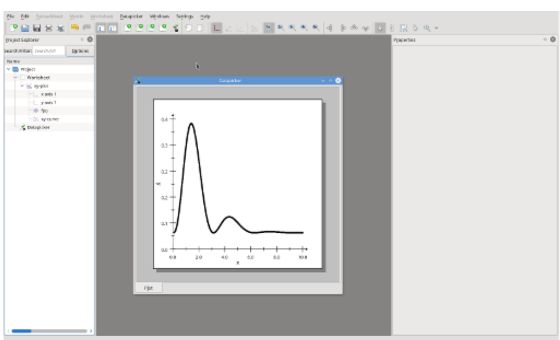
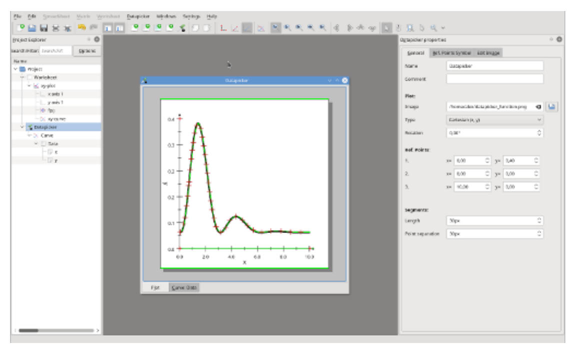
It is possible to add more then one datapicker curve. This is useful in case the imported image contains several curves that need to be digitized. The datapicker curve that is currently being selected in the Project Explorer is the ″active″ one - points clicked on the datapicker image will be calculated and added to its data spreadsheet.

Calculated values are stored in different columns in data spreadsheets in the datapicker. These columns behave exactly the same like other columns in usual spreadsheets and can be directly used as source columns for curves in your own plots.

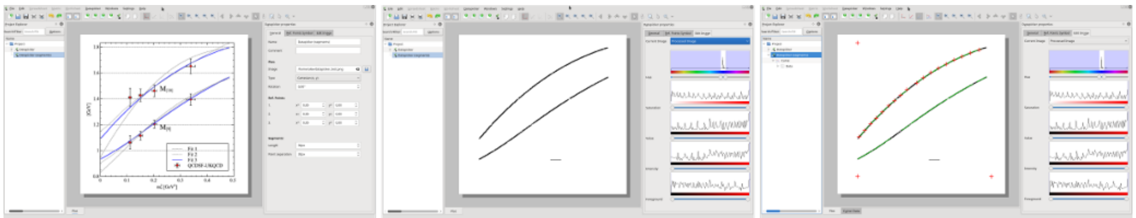
Datapicker supports the process of the data extraction with several helpers. To place the points more precisely, a magnification glass with different magnification levels is available. Also, the last selected point can be shifted with the help of the navigation keys. Furthermore, when reading data points having error bars, datapicker automatically creates bars indicating the end points of the error bars. Those bars can be pulled with the mouse until the required length (the distance to the data point) is reached.

The procedure for the extraction of data from an imported plot as described above is feasible when dealing with a limited number of points. In case the curves in the imported image are given as solid lines, the datapicker tool in LabPlot allows to read them (semi-)automatically. For this, after a new datapicker curve was added as described above, switch to the mouse mode ″Select Curve Segments″. The curves on the plot are recognized and highlighted. By clicking on a highlighted curve (or one of its segments), points along this curve are created. The length of a segment and the density of created points (separation between two points) are adjustable parameters. On the screenshots below, after switching to the segment mode all black lines were highlighted (green colour). In this specific case, the curve was recognized as a single segment and a single mouse-click on this segment is sufficient to digitize this curve and to automatically place points along the curve.

In many cases the plot is not as simple as above (single black curve on white background) and contains grid lines, many curves of different colour and thinness and a non-white background. In such a case the automatic detection fails (too many or no objects are highlighted). To help the datapicker to determine the curve(s) correctly, the user has to limit the allowed ranges in the HSV (or HSI) colour spaces. To subtract the non-white background it is possible to limit the range for the foreground colour, too. Internally, each pixel of the image is converted to black and white where only the points fitting into the user-defined ranges for hue, saturation, value, intensity and foreground are set to black.

On the screenshots below, the blue curves in the original image were projected onto by having appropriately reduced the allowed ranges in the colour space (note the peak for blue in the histogram for the hue). The transformed black and white image contains only the curves the user is interested in and it is now an easy task for the datapicker to determine the curves and to place points on them.



Similar to Worksheet, the currently visible area in the datapicker can be exported. The supported image formats as described in the section Export Dialog.

2.12 Import Dialog

In the import dialog you can import data into one of the available spreadsheets or matrices in LabPlot. The supported data formats are

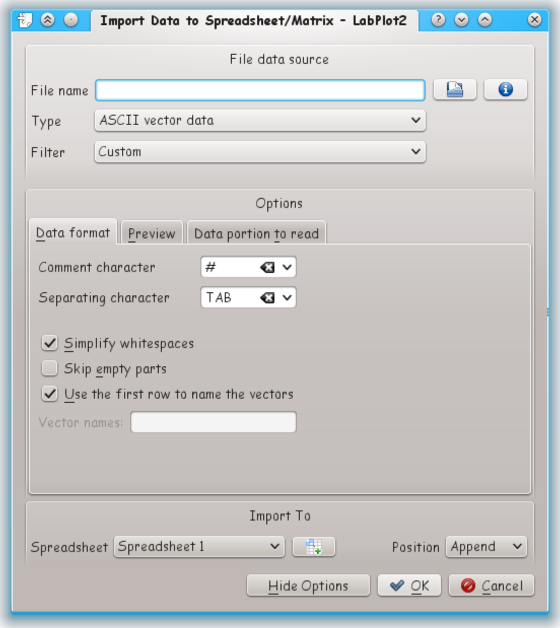
- ASCII
- Binary
- Image
- NetCDF
- HDF5
- FITS

Preview of all supported file types is available in the import dialog. For data formats with complex internal structures (like NetCDF, HDF5 and FITS), the content of the file is presented in a tree view that allows comfortable navigation through the file. A versatile dialog to edit the headers (keywords) of a FITS file is also provided.

Import of ascii and binary data compressed with gzip, bzip2 or xz can be done directly as the decompression happens transparently for the user.

The name of the file containing the data to import has to be provided. The **File Info** button opens a dialog where some information about the selected file is shown. The type of the data can be specified - currently, only ASCII files containing several data sets (vectors) stored as columns are supported. The filter - automatic or custom - determines how the file has to be parsed. Selecting the filter ″custom″, several parameters like separating character etc. can be provided manually in this case.

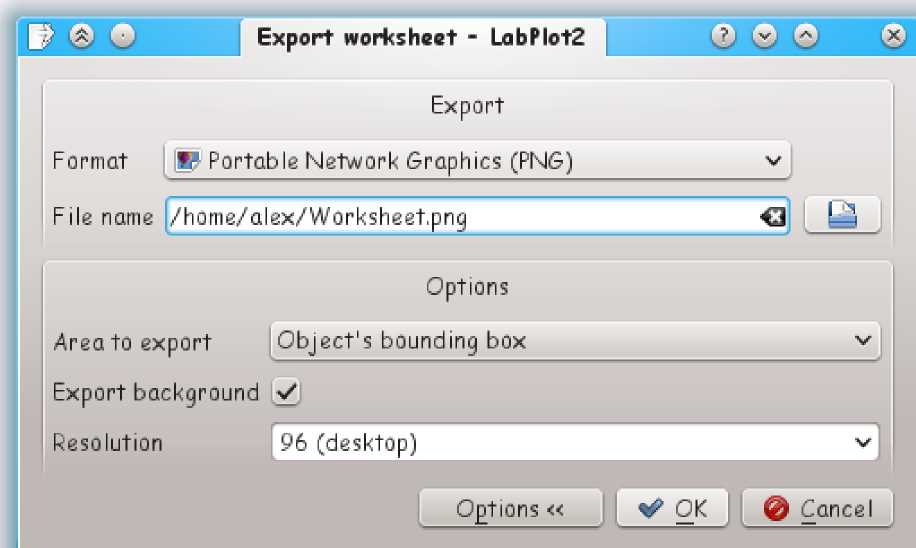
The start and end row to read can be customized using the **Data portion to read** tab. To read all data specify `-1` as an end row or column.



2.13 Export Dialog

A worksheet can be exported to several graphics format (vector and raster). The export is done via the export dialog reachable via the **Export** in the main toolbar or **File** → **Export** in the main menu.

Besides the graphics format, the user can specify which part of the worksheet has to be exported and whether the background has to be exported or not. Also, for raster graphics the image resolution can be provided.



The content of a spreadsheet can be exported to an external text or FITS file. In the export dialog for spreadsheets the user can specify the character separating values of different columns. Optionally, the header of the spreadsheet (names of the columns in the spreadsheet) can be exported.

Chapter 3

Command Reference

3.1 The File Menu

File → New (Ctrl-N)
Creates a new LabPlot project file.
In a project file all settings and all plots are stored in ASCII format.

File → Open (Ctrl-O)
Opens a LabPlot project file.

File → Open Recent
Opens a recent LabPlot project file.
Here the last used 10 project files are listed.

File → Save (Ctrl-S)
Saves the actual project.
If you haven't saved the project before the project is saved under a temporary project file name.

File → Save As
Saves the actual project under a different name.

File → Print (Ctrl-P)
Prints the active plot.
Here a print dialog is opened where you can select the printer, different paper sizes, etc.

File → Print Preview
Open a print preview window. LabPlot allows you to choose print settings using the toolbar of this window and view the result immediately.

File → New → Spreadsheet (Ctrl-=)
Creates a new spreadsheet in the current folder of LabPlot project.

File → New → Worksheet (Alt-X)
Creates a new worksheet in the current folder of LabPlot project.

File → New → Folder
Creates a new spreadsheet in the current folder of LabPlot project.

File → New → File Data Source
Opens **Import data to spreadsheet/matrix** window.

File → Import (Ctrl-Shift-L)
Import data into the active spreadsheet

This item can be used to import data into LabPlot. Please read more in the import dialog section.

File → Export
Saves the active plot as special format.

Currently supported are Encapsulated Postscript (EPS), Portable Document Format (PDF), Scalable Vector Graphics (SVG) and Portable Network Graphics (PNG).

File → Close (Ctrl-W)
Closes the current opened LabPlot project file.

File → Quit (Ctrl-Q)
Quit LabPlot.

3.2 The Edit Menu

Edit → Undo/Redo History
Opens the LabPlot action history window. Select an item in the list to navigate to the corresponding step.

3.3 The Worksheet Menu

This menu contains all the items that can also be found in the context menu (right mouse) of a worksheet. The menu is only available when a worksheet object is selected on the **Project Explorer** panel.

3.4 The Spreadsheet Menu

This menu contains all the items that can also be found in the context menu (right mouse) of a spreadsheet. The menu is only available when a spreadsheet object is selected on the **Project Explorer** panel.

3.5 The CAS Worksheet Menu

This menu contains all the items that can also be found in the context menu (right mouse) of a CAS worksheet. The menu is only available when a worksheet object is selected on the **Project Explorer** panel.

3.6 The Datapicker Menu

This menu contains all the items that can also be found in the context menu (right mouse) of a datapicker. The menu is only available when a datapicker object is selected on the **Project Explorer** panel.

3.7 The Settings Menu

This menu gives you the ability to change user settings.

Apart from the common KDE Settings menu entries described in the Settings Menu chapter of the KDE Fundamentals LabPlot has this application specific menu entry:

Settings → Full Screen Mode (Ctrl-Shift-F)
: Show the workspace in full screen mode.

3.8 The Help Menu

Additionally, LabPlot has the common KDE Help menu items. For more information, read the section about the Help Menu of the KDE Fundamentals.

3.9 Toolbar

The main toolbar contains the main items that you can find in the different menus. More details on this can be found in the KDE Fundamentals manual.

Chapter 4

Plotting

4.1 Plots

Plots can be created inside a worksheet via ″Add new″ in the context menu or in the application menu via ″Worksheet″ by selecting ″xy-plot″ and the type of plot you like to have.

Within this xy-plot you can add a xy-curve containing data to show (again via the context menu or application menu).

The settings of a plot can be changed in the corresponding dock widget. There are general settings like geometry but also the range of the x- and y-axis (including scaling). The plot title can be set in the ″Title″ tab of the dock widget. Background and border styles can be changed in the ″Plot Area″ tab.

4.2 Curves

Curves contain data points that can be shown in a plot. There are three different method to create curves: the standard xy-curve, a xy-curve from a mathematical expression and a xy-curve from a data analysis function.

The standard xy-curve can be filled with values of a spreadsheet by selecting the x-data and y-data as column of the spreadsheet in the xy-curve dock widget. Another method to fill a curve is to use a mathematical expression. Here you can select any mathematical function and range to create the curve. The third method to create a curve is to use a data analysis function. The data and the analysis function can be selected in the dock widget of the analysis function.

For all types of curves the line and symbols styles can be changed in the dock widget. Also annotated values and error bar settings can be changed here.

4.3 Legends

A legend can be easily added to a plot by using the context of application menu. It contains information about all curves in a plot.

The settings of a legend (format and geometry) can be changed in the legend dock widget. Also the legend title settings, the legend background and the layout can be changed in the corresponding tab of the legend dock widget.

Chapter 5

Analysis functions

5.1 Overview

LabPlot supports a wide variety of data analysis functions:

- Data reduction
- Differentiation
- Integration
- Interpolation
- Smoothing
- Nonlinear curve fitting
- Fourier filter
- Fourier transform

All of them can be applied to any data consisting of x- and y-columns. The analysis functions can be accessed using the Analysis menu or the context menu of a worksheet. The newly created curves can be customized (line style, symbol style, etc.) like any other x-y-curve.

5.2 Data reduction

To reduce the number of data points without losing the features of a data set you can apply one of several line simplification algorithm:

- Douglas-Peucker
- Visvalingam-Whyatt
- Reumann-Witkam
- Perpendicular distance simplification
- n-th point simplification
- Radial distance simplification
- Interpolation (nearest neighbor)

- Opheim
- Lang

The desired tolerance is automatically calculated from the data but can also be changed in the dock widget.

5.3 Differentiation

Numerical differentiation of data can be done specifying:

- order of derivation (first to sixth order)
- order of accuracy (up to 4th order, depending on derivation order)

5.4 Integration

Numerical integration of data can be done specifying one of the methods

- rectangle (1-point) rule
- trapezoid (2-point) rule
- Simpson-1/3 (3-point) rule
- Simpson-3/8 (4-point) rule

The default method (trapezoid) should be suitable for most cases. The number of resulting data points is reduced for both Simpson-rules due to the properties of these methods.

5.5 Interpolation

Interpolation of data can be done with several algorithm:

- linear
- polynomial (if number of data points < 100)
- cubic spline
- cubic spline (periodic)
- Akima spline
- Akima spline (periodic)
- Steffen spline (needs GSL ≥ 2.0)
- cosine
- exponential
- piecewise cubic Hermite (finite differences, Catmull-Rom, cardinal, Kochanek-Bartels)
- rational functions

The interpolating function is calculated with the given number n of data points and evaluated as:

- function
- derivative
- second derivative
- integral (starting from zero)

5.6 Smoothing

A number of different smoothing methods are supported:

- Moving average (central)
- Moving average (lagged)
- Percentile filter
- Savitzky-Golay

All smoothing methods support several padding modes (constant, periodic, mirror, nearest, etc.) for the beginning and end of the data set. The moving averages support several weight functions (uniform, triangular, binomial, parabolic, tricubic, etc.) which can be selected to weight the selected data points depending on their distance.

5.7 Curve fitting

Linear and non-linear curve fitting of data can be done with several predefined fit-models (for instance polynomial, exponential, Gaussian or custom) to data consisting of x- and y-columns with an optional weight column. With a custom model any function with unlimited number of parameters can be used for fitting. The results including statistical properties are displayed in the results text.

The start values of the parameter can be set in the parameter dialog. It is also possible to fix any parameter and set lower and upper limits to the values here. Be aware that reducing the parameter space by fixing parameter or specifying limits can slow down convergence or avoid finding a good result. It's always a good idea to remove any parameter limitations when good start values are found.

Following options can be set in the options dialog to optimize the fitting:

- Max. iterations: number of maximum iterations
- Tolerance: desired tolerance for result
- Evaluated points: number of points to evaluate the fit function
- Evaluate full range: evaluate the fit function for the full data range instead of evaluating only for the given x range
- Use results as new start values: results will be the new parameter start values

5.8 Fourier filter

This function can be used to apply a Fourier filter to any data consisting of x- and y-columns. Supported filter types are:

- Low pass
- High pass
- Band pass
- Band reject (band block)

where any of them can have the form

- Ideal
- Butterworth (order 1 to 10)
- Chebyshev type I or II (order 1 to 10)
- Optimal ″L″egendre (order 1 to 10)
- Bessel-Thomson (any order)

The cutoff value(s) can be specified in the units frequency (Hertz), fraction (0.0 to 1.0) or index of the data points.

5.9 Fourier transform

To convert a signal from time to frequency domain or to change between other conjugate variables like position and momentum (k-space) a discrete Fourier transform can be applied. Following options can be used to suite one needs:

- Window function (Welch, Hann, Hamming, etc.) to avoid leakage effects
- Output (magnitude, amplitude, phase, dB, etc.)
- One or two sided spectrum with or without shifting
- X axis scaling to frequency, index or period

Chapter 6

Curve Tracing

6.1 Upload Image

Datapicker can be created inside a project via **Add new** in the context menu of project/folder or in the main toolbar. After that a new image can be added and can be changed via **Plot** in the corresponding dock widget.

After uploading image different zooming options can be used from the context menu/datapicker toolbar to change width and height of image. Image can also be rotated to an angle using **Rotation** in the ″edit″ section of dock widget. After this user have to set axis points.

6.2 Symbols

Symbols are the points that can be drawn over image of datapicker. Symbols can be directly created by mouse right click over the image. Symbols are mainly of two types, with and without error-bar depending on the type of curve they belong.

Every curve of datapicker can have its own symbol style that can be changed in the **Symbols** section of dock widget. ″SelectAndMove″ mouse mode can be used to select multiple points/symbols and can be moved by using navigation keys.

6.3 Axis Points

Axis Points are the set of three reference points over image of datapicker. These points can be set via **Set Axis Points** in the context menu of datapicker. After selecting points over image user have to update their coordinate system type via **Plot Type** and logical positions via **Ref. Points** in the dock widget.

6.4 Datapicker Curve

Datapicker-Curve can be created inside datapicker via **New Curve** in the context menu of datapicker. A curve can have different types of X and Y errors (No-error, symmetric, asymmetric). This depends on the type of errors dock widget of datapicker have at the point of creation.

Every curve object contains all the curve points (hidden) and a spreadsheet that contains logical positions of all its curve points, and provides options to update spreadsheet and to toggle visibility of its curve points using the context menu. Mode **Set Curve Points** in the context menu of datapicker should be selected in order to create curve points.

Multiple curve can be created for same datapicker. The created curve points always correspond to the active curve of datapicker which can be changed via **Active Curve** option in the context menu and dock widget of datapicker. Every curve of datapicker can have its own symbol style that can be changed in the **Symbols** section of dock widget.

6.5 Curve Segments

Curve segment for datapicker can be created over image by switching mode to **Select Curve Segments** in the context menu of datapicker. A segment is a selectable object over image which can be selected by mouse right click over it.

Segments are created by processing of image on the basis range of colour attributes in order to automatically trace curves. To improve results these range and types of colour attributes can be changed in the ″edit″ section of dock-widget. Dock-widget also provides options to switch among processed image and original image, and to set the minimum possible length of segments.

Once a segment is selected it will create curve points over it with a minimum specified distance among them. The minimum specified distance among the points can be changed in the dock widget of datapicker. User might have to select the segments again in order to observe the changes.

Chapter 7

Advanced Topics

Here you will find some explanations of advanced topics.

7.1 Topics

7.1.1 Error bars

If you want to plot data with error bars just import your data with the import dialog into your project. Then use the **Error bars** tab of the curve properties to select **Error type**, choose the error column from the **Data, +-** list. Format of the error bars can be defined using the **Format:** pane.

7.1.2 TeX label

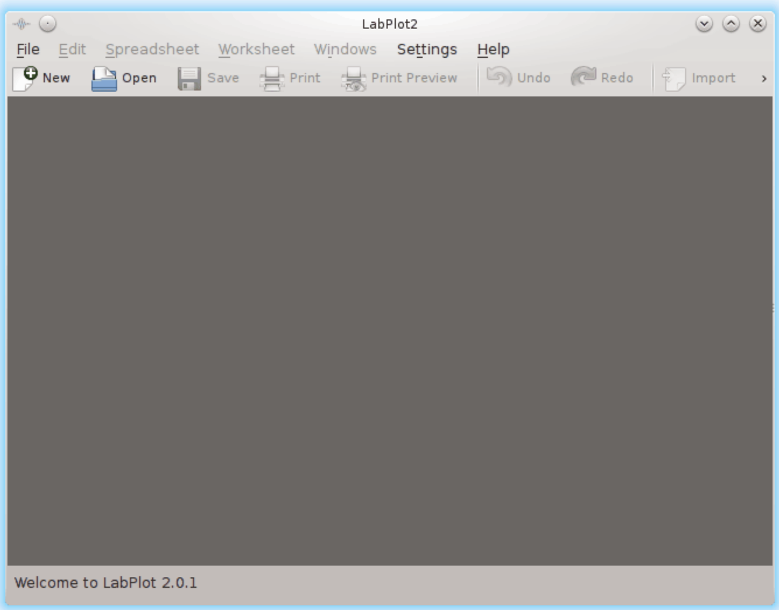
For using TeX label you just have to activate the switch button **TeX** in the **Title** tab. With that every text you enter in the text box is rendered by TeX and plotted accordingly. Since this conversion takes some time you may see a certain delay when redrawing the plot.

Chapter 8

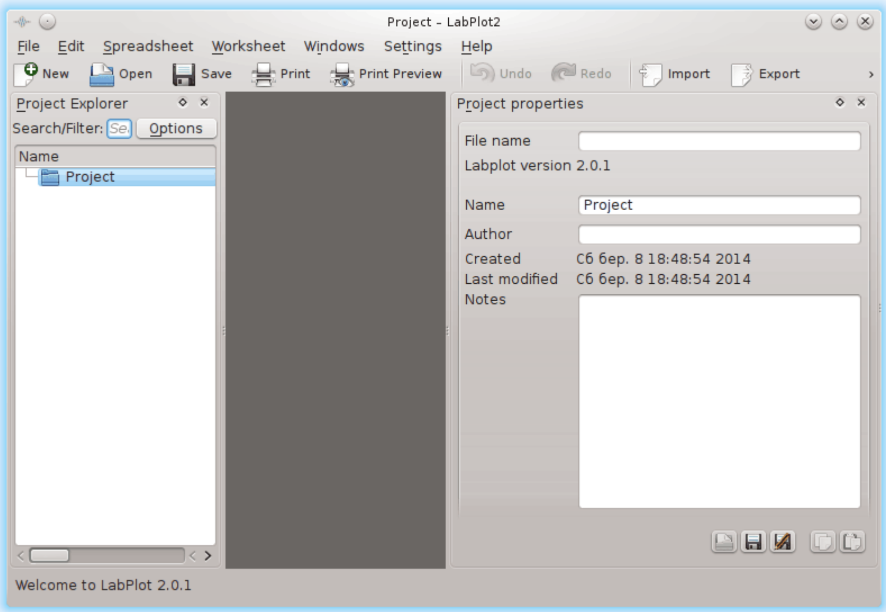
Short Tutorials

8.1 Building a sine graph with LabPlot

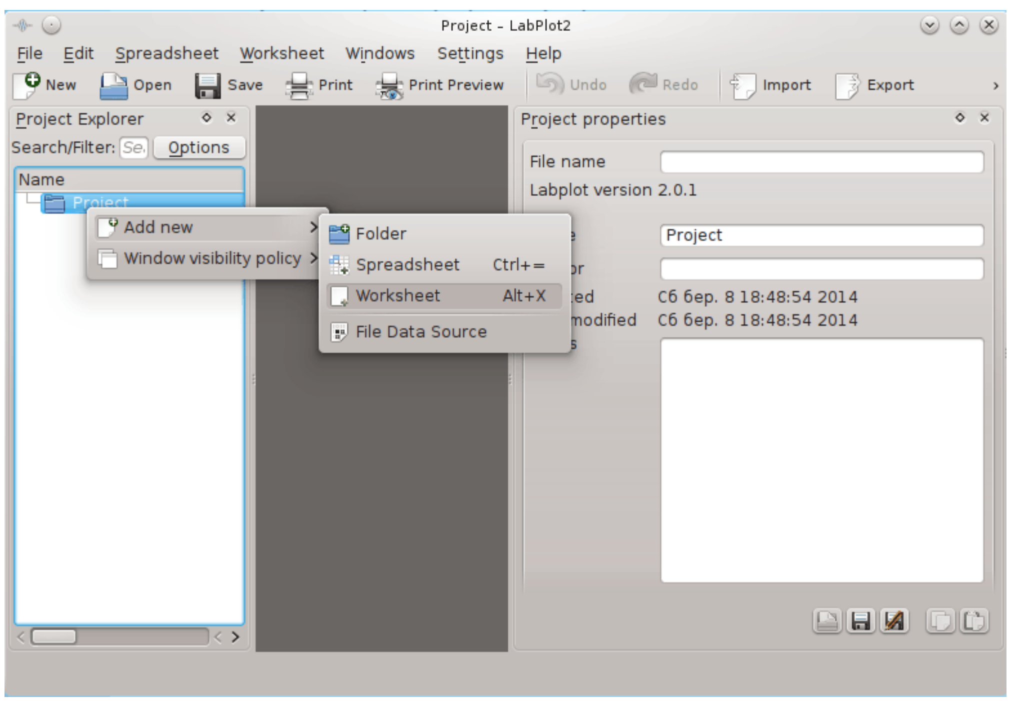
In this chapter you will find explanations on how to build a simple plot for a curve in the Cartesian coordinates from a mathematical equation.



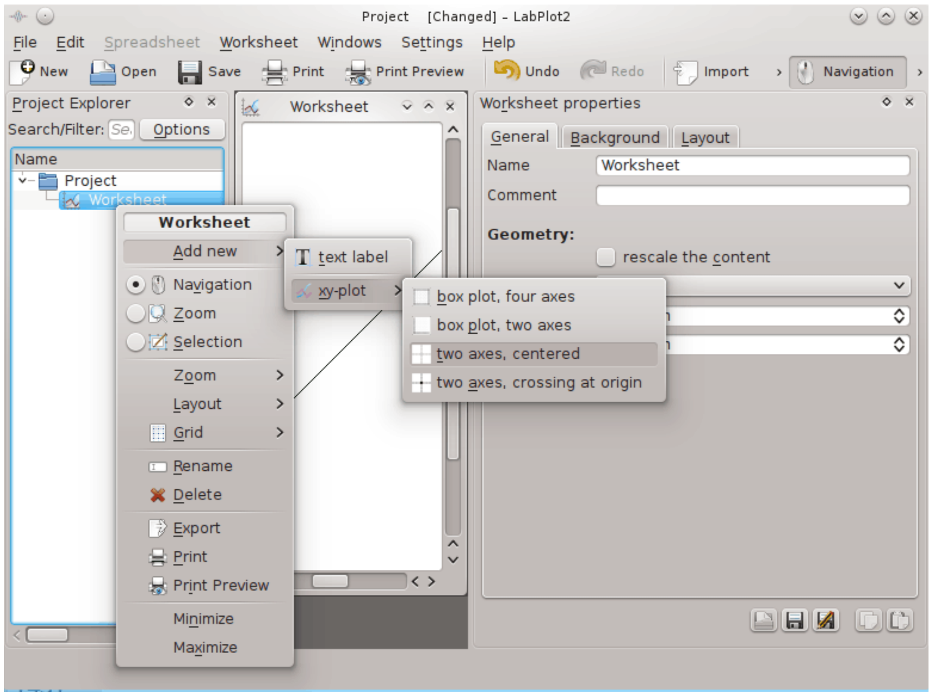
1. Click on the **New** button or press **Ctrl-N** on the keyboard.



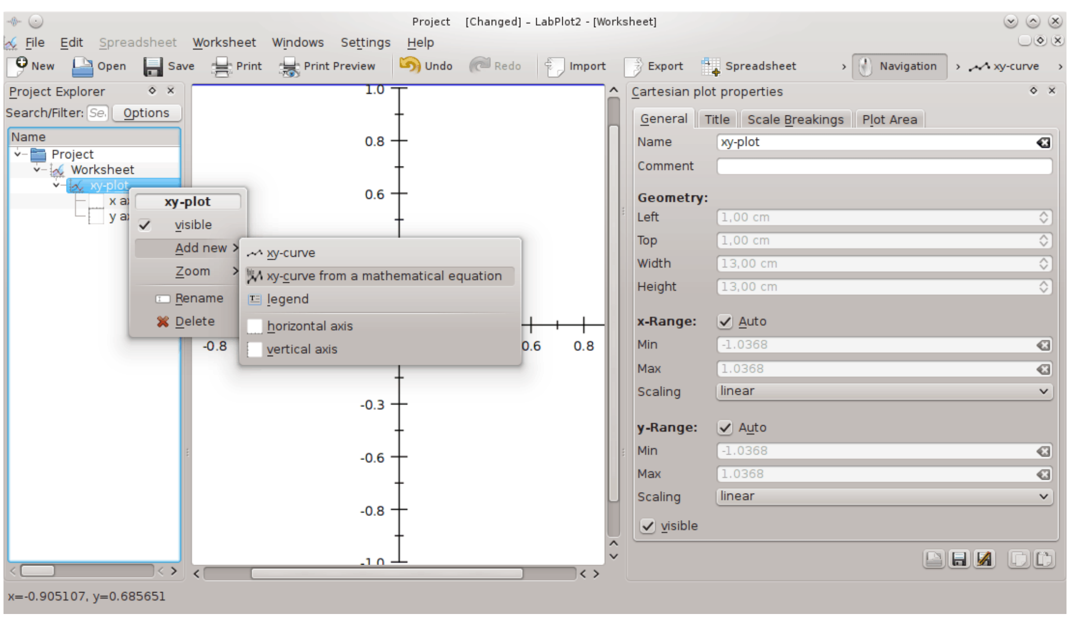
2. Click on the **Project** item on the **Project Explorer** panel with the right mouse button and choose **Add new** → **Worksheet** or press **Alt-X** on the keyboard.



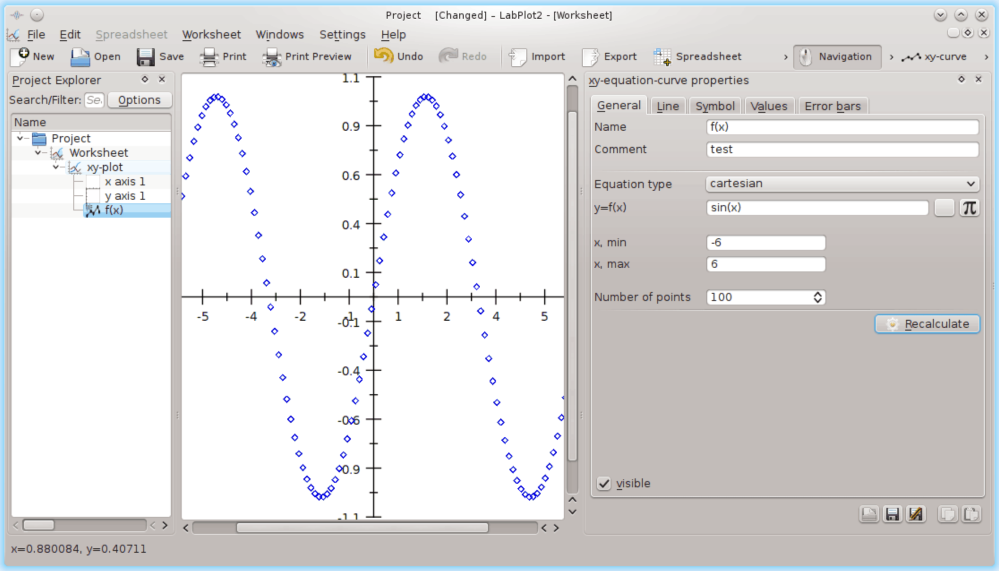
3. Click on the **Worksheet** item on the **Project Explorer** panel with the right mouse button and choose **Add new** → **xy-plot** → **two axes, centered**.



4. Click on the **xy-plot** item on the **Project Explorer** panel with the right mouse button and choose **Add new** → **xy-curve from a mathematical equation**.



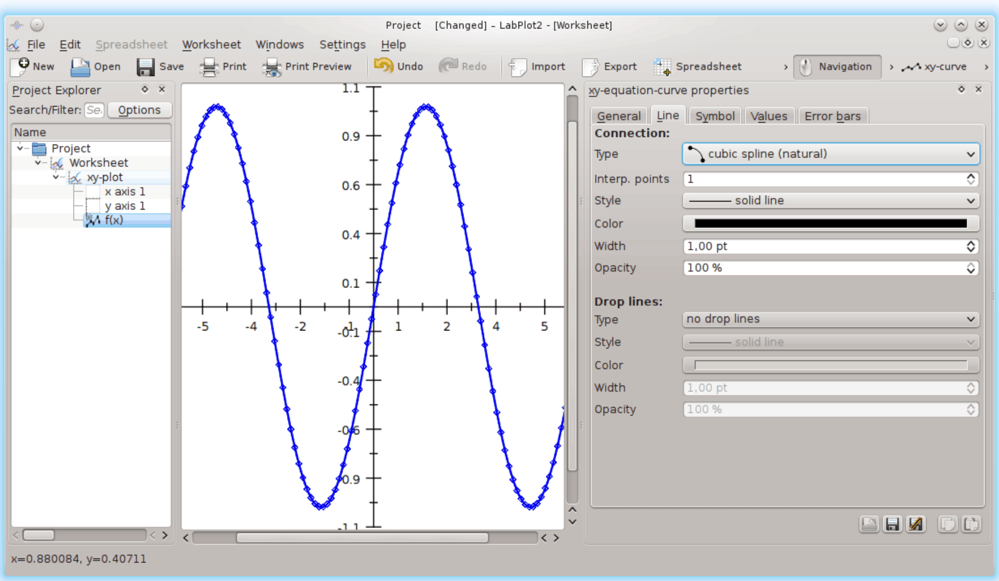
5. Use the **xy-equation-curve properties** pane on the right to enter **`sin(x)`** into the **y=f(x)** field (for the list of available functions please see chapter 10), **−6** into the **x, min** field, **6** into the **x, max** field and click on the **Recalculate** button to see the result.



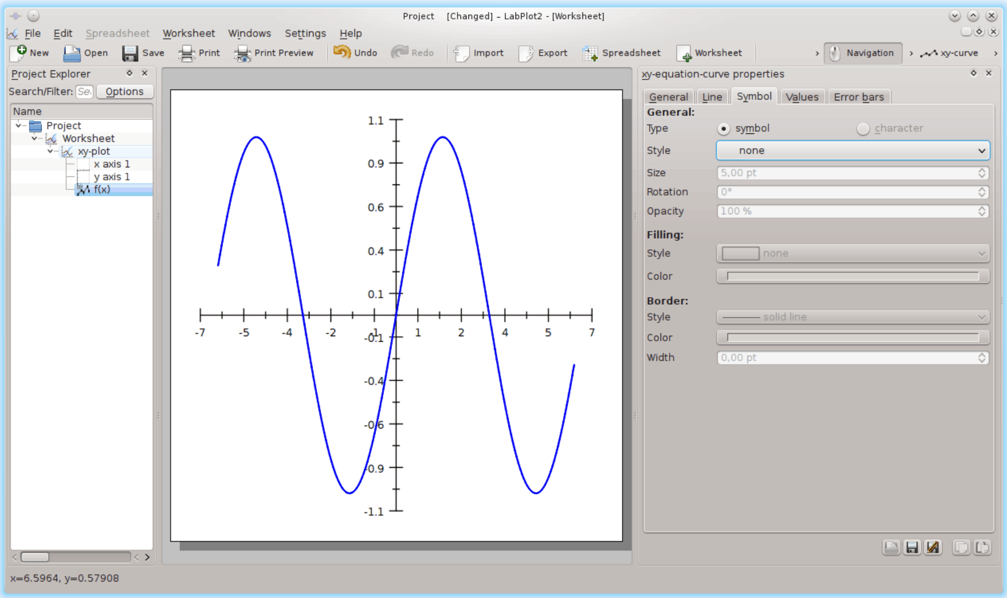
NOTE
LabPlot highlights unknown syntax in the **y=f(x)** field. This is useful to control the correctness of the input.

IMPORTANT
The list of the known functions can be found in corresponding section of this manual.

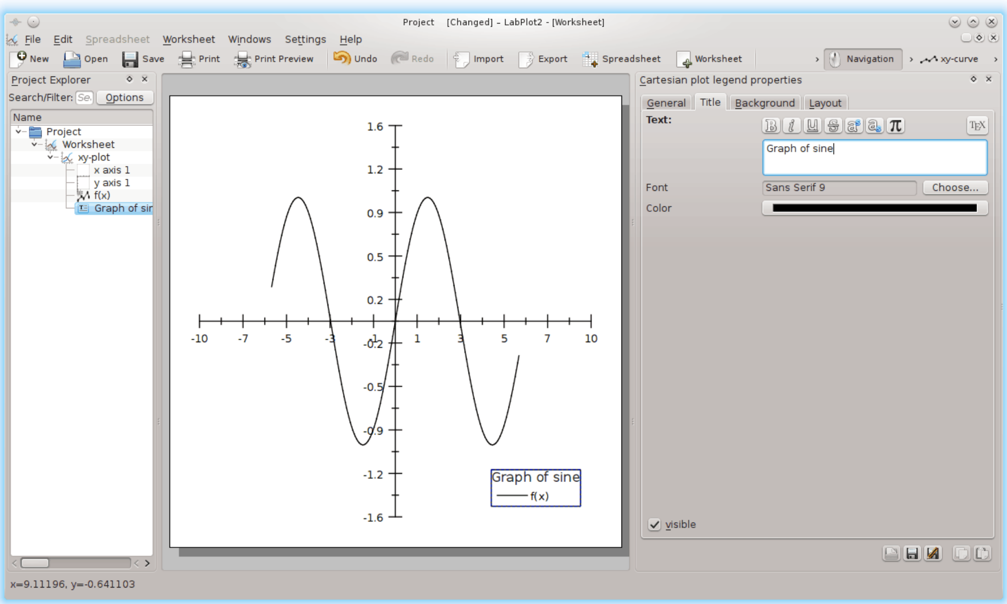
6. Switch to the **Line** tab on the **xy-equation-curve properties** pane and choose **cubic spline (natural)** from the **Type** drop down box.



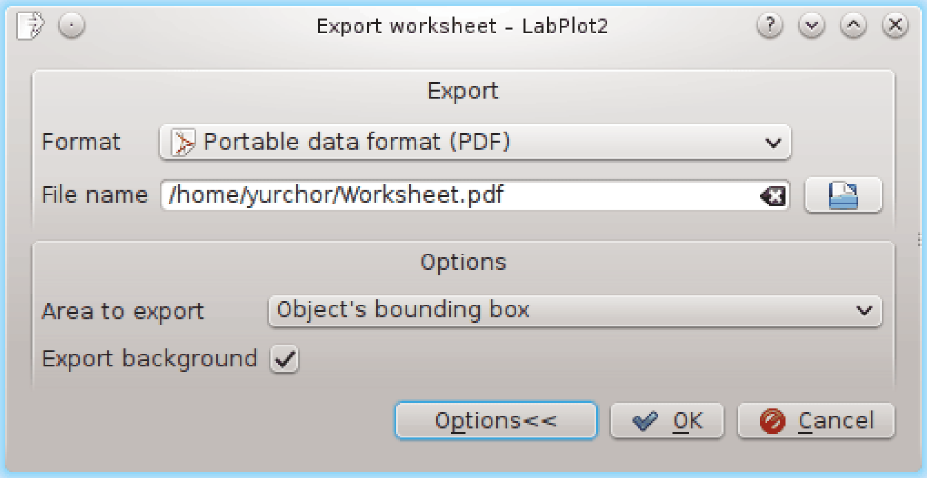
7. Switch to the **Symbol** tab on the **xy-equation-curve properties** pane and choose **none** from the **Style** drop down list.



8. Click on the **xy-plot** item on the **Project Explorer** panel with the right mouse button and choose **Add new** → **legend**. Switch to the **Title** tab on the **Cartesian plot legend properties** pane and enter **`Graph of sine`** into the **Text** field.

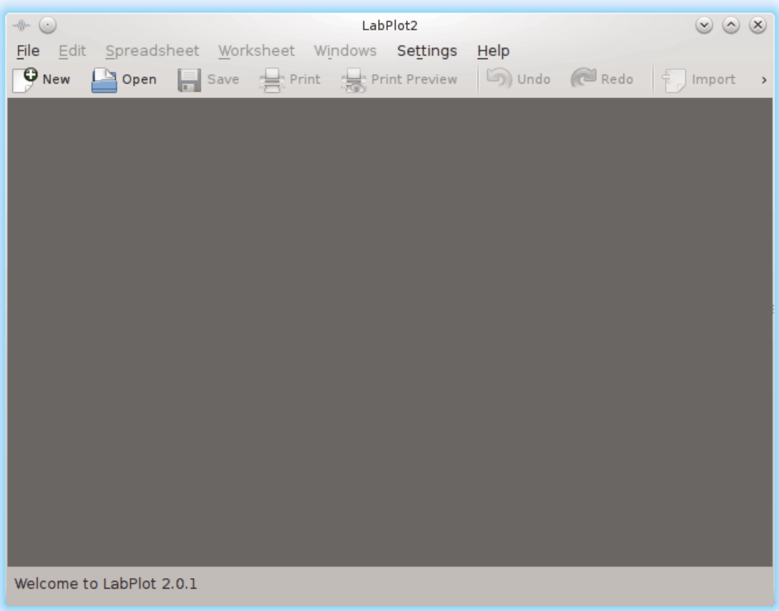


9. Choose **File** → **Export** from the main menu. Select the place and the format to save the plot.

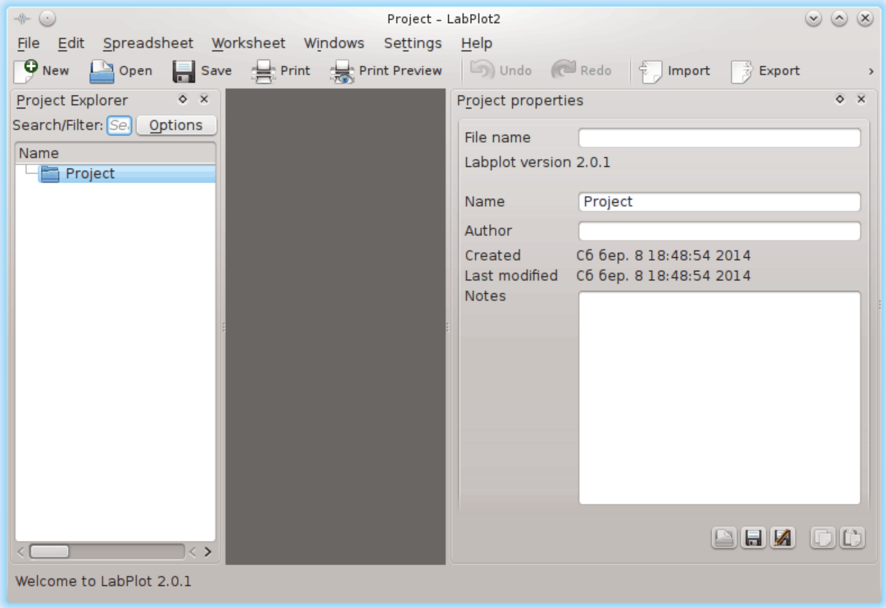


8.2 Building a graph from spreadsheet data with LabPlot

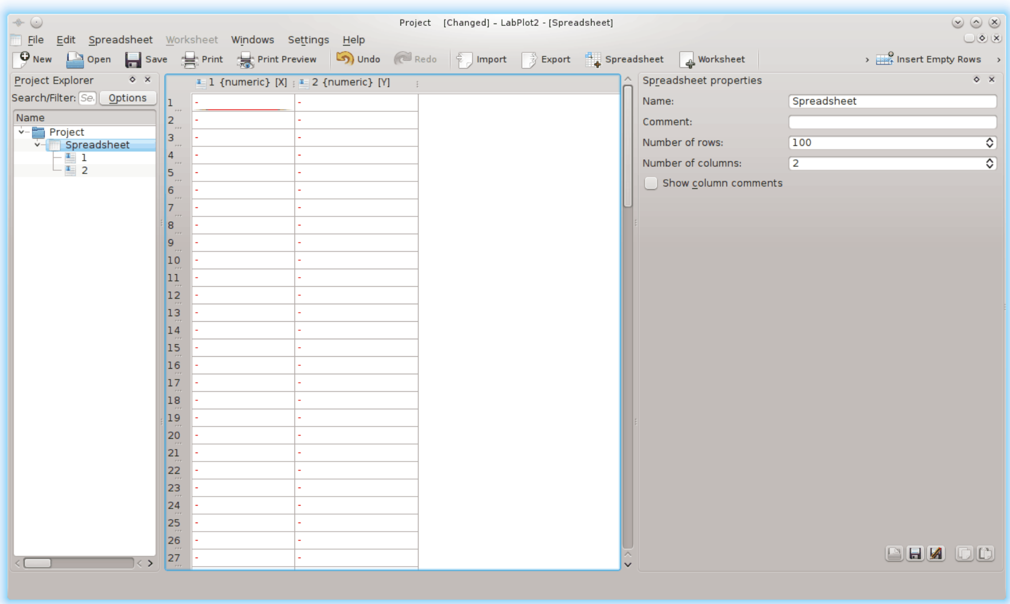
In this chapter you will find explanations on how to build a simple plot from spreadsheet data.



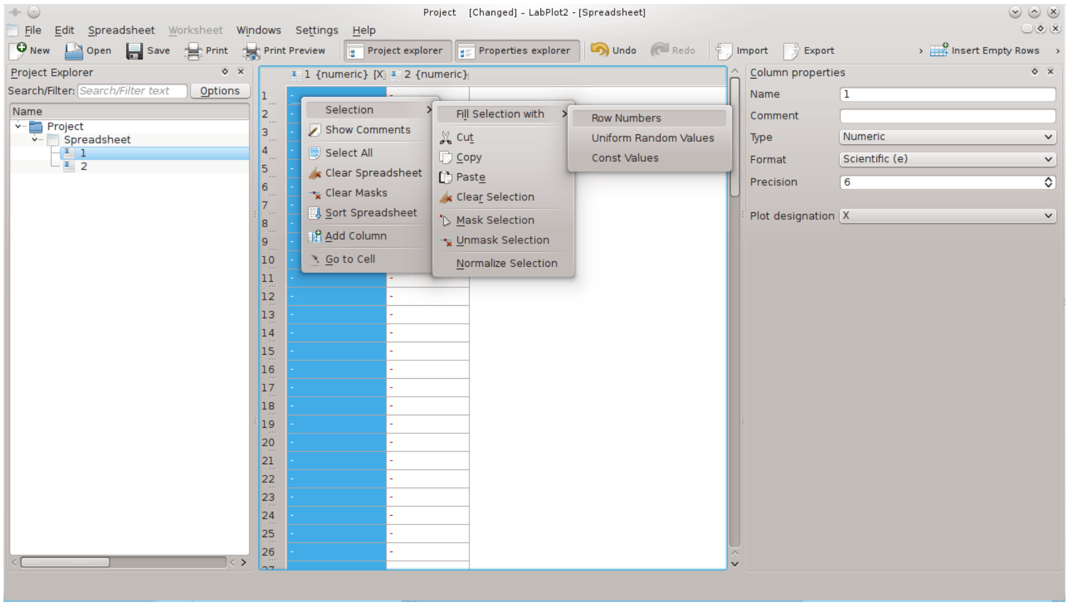
1. Click on the **New** button or press **Ctrl-N** on the keyboard.



2. Click on the **Project** item on the **Project Explorer** panel with the right mouse button and choose **Add new** → **Spreadsheet** or press **Ctrl-=** on the keyboard.

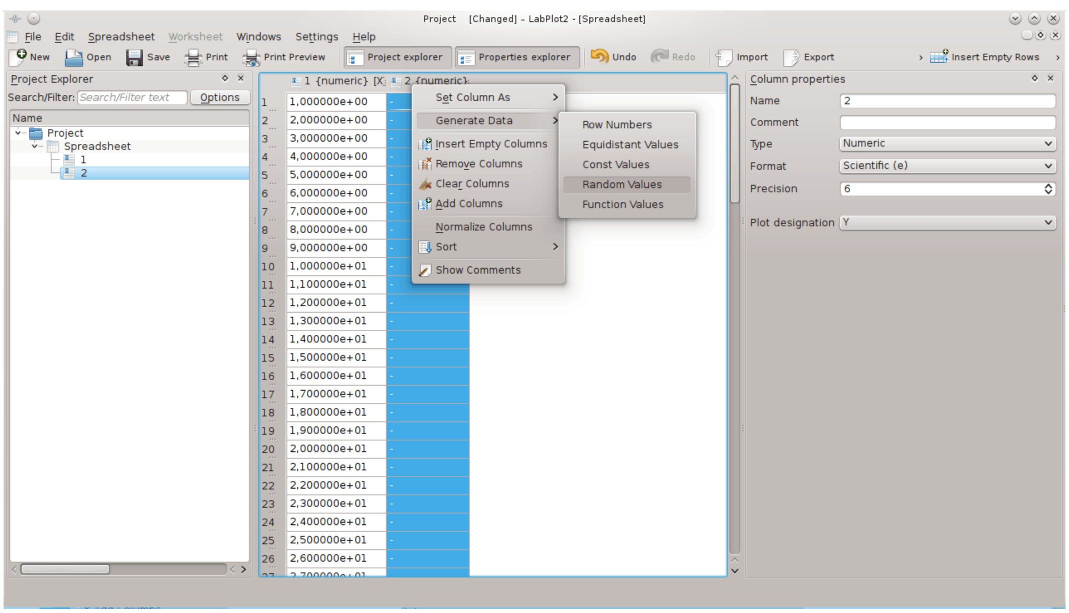


3. Click on the header of the first column of the spreadsheet with the left mouse button then click on any of its cells with right mouse button and choose **Selection** → **Fill Selection with** → **Row Numbers**.

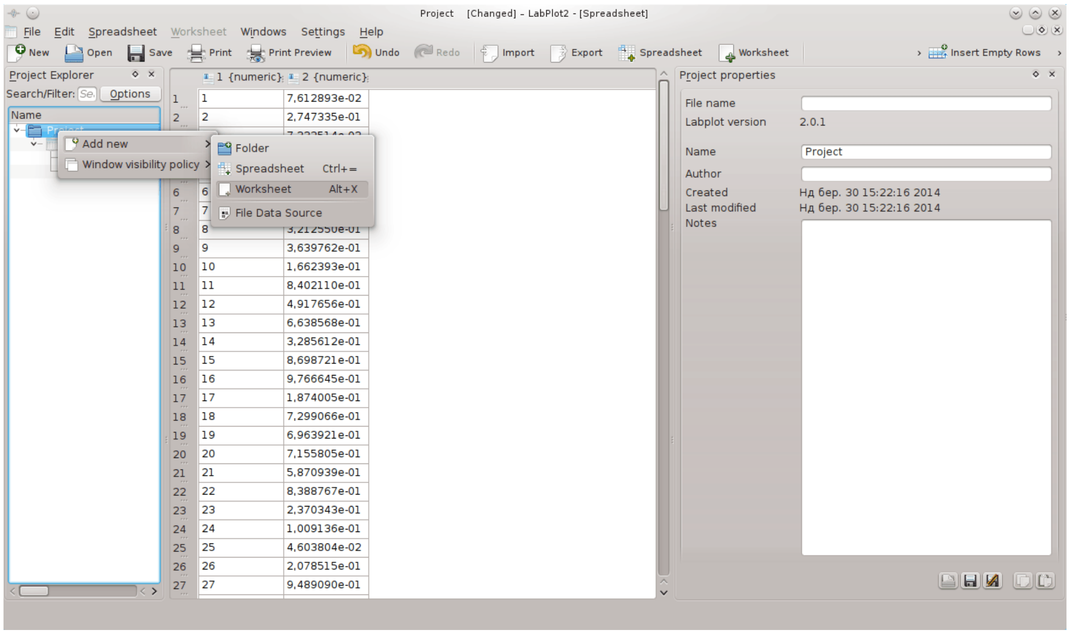


Select **Automatic (g)** from the **Format** drop down box on the **Column properties** right dock to enhance data presentation for the first column.

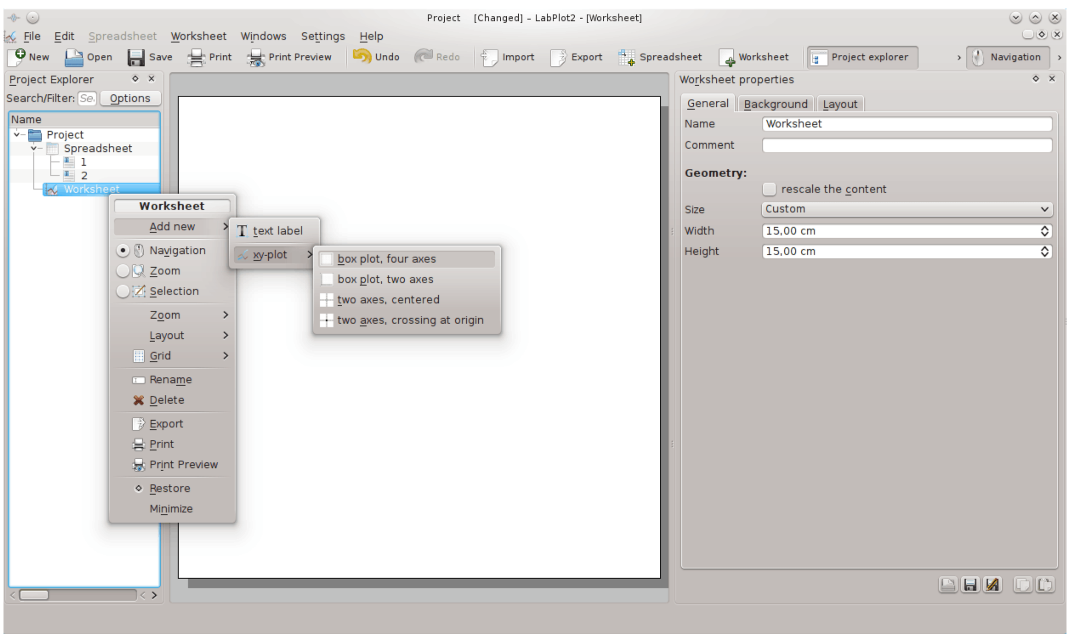
4. Click on the header of the second column of the spreadsheet with the right mouse button and choose **Generate Data** → **Random Values**.



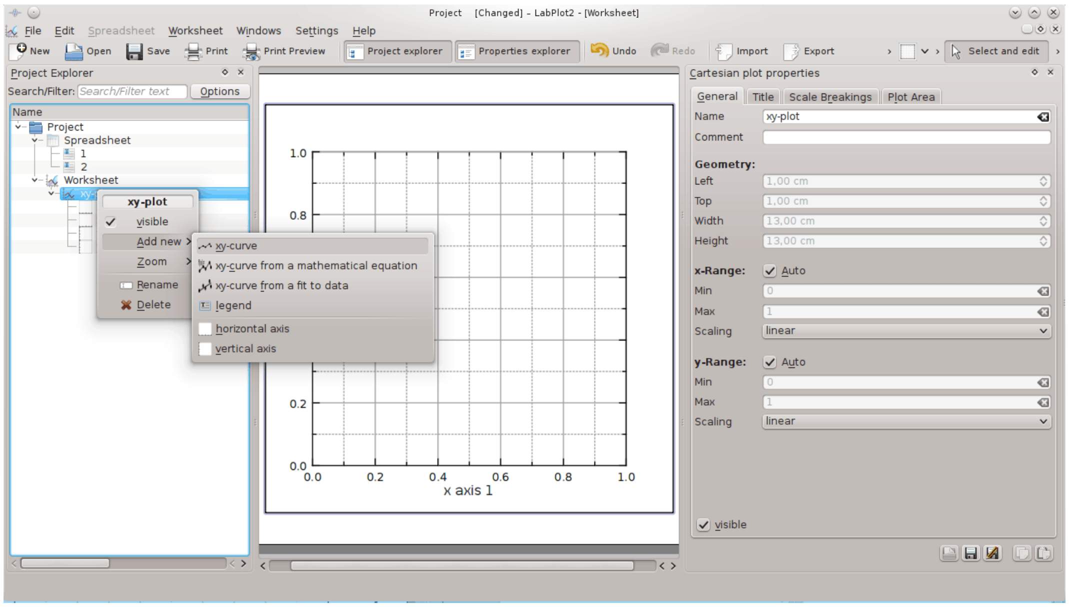
5. Click on the **Project** item on the **Project Explorer** panel with the right mouse button and choose **Add new** → **Worksheet** or press **Alt-X** on the keyboard.



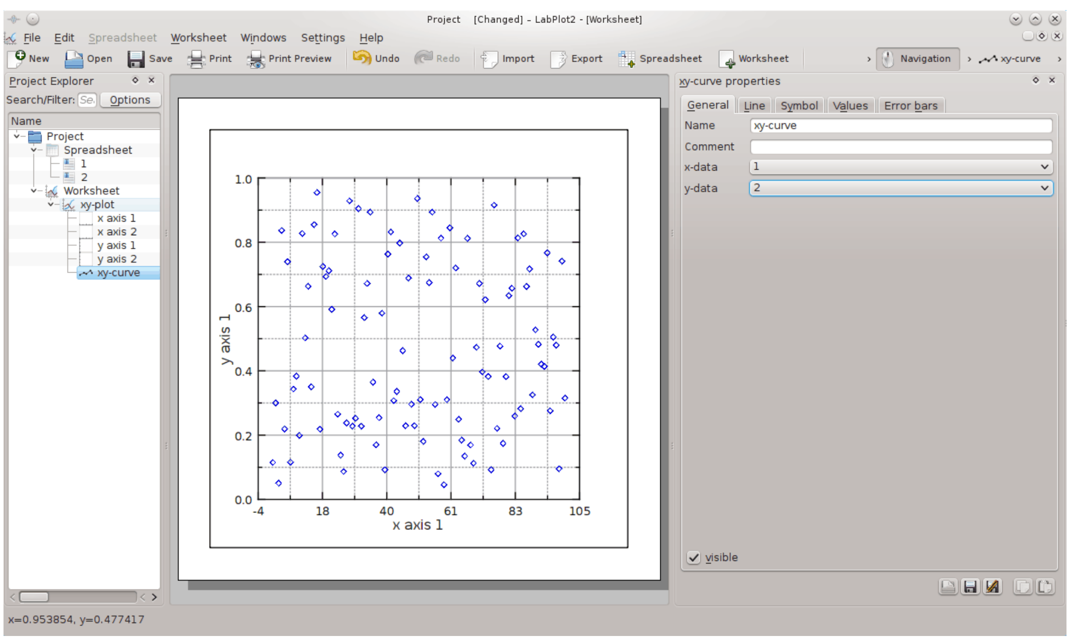
6. Click on the **Worksheet** item on the **Project Explorer** panel with the right mouse button and choose **Add new** → **xy-plot** → **box plot, four axes**.



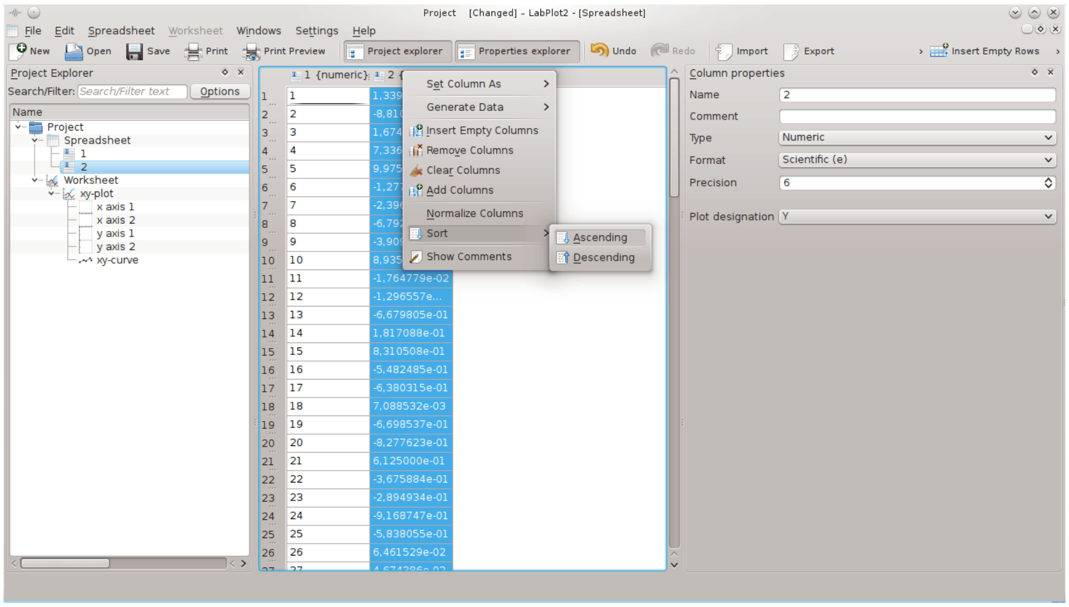
7. Click on the **xy-plot** item on the **Project Explorer** panel with the right mouse button and choose **Add new** → **xy-curve**.



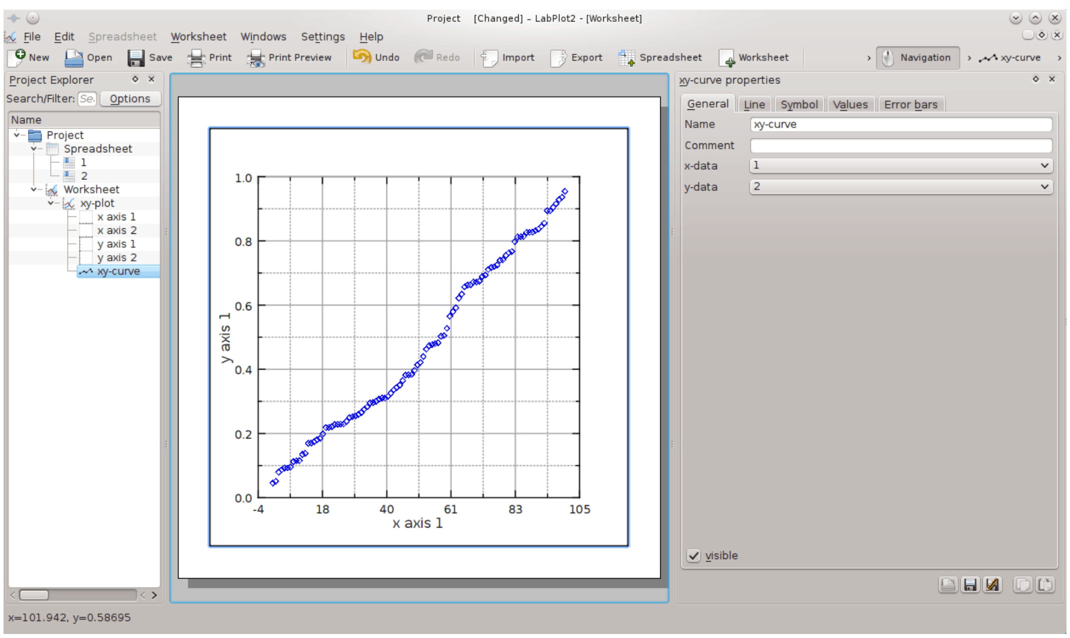
8. Use the **xy-curve properties** pane on the right to select **Project** → **Spreadsheet** → **1** in the **x-data** field (just click on the item and press **Enter**). Use the same procedure to select **2** for the **y-data** field. The results will be shown on the worksheet immediately.



9. Click on the **Spreadsheet** item on the **Project Explorer** panel with the left mouse button then click on the second column header with the right mouse button and choose **Sort** → **Ascending**.



10. Click on the **Worksheet** item on the **Project Explorer** panel with the left mouse button to see the results.



Chapter 9

Examples

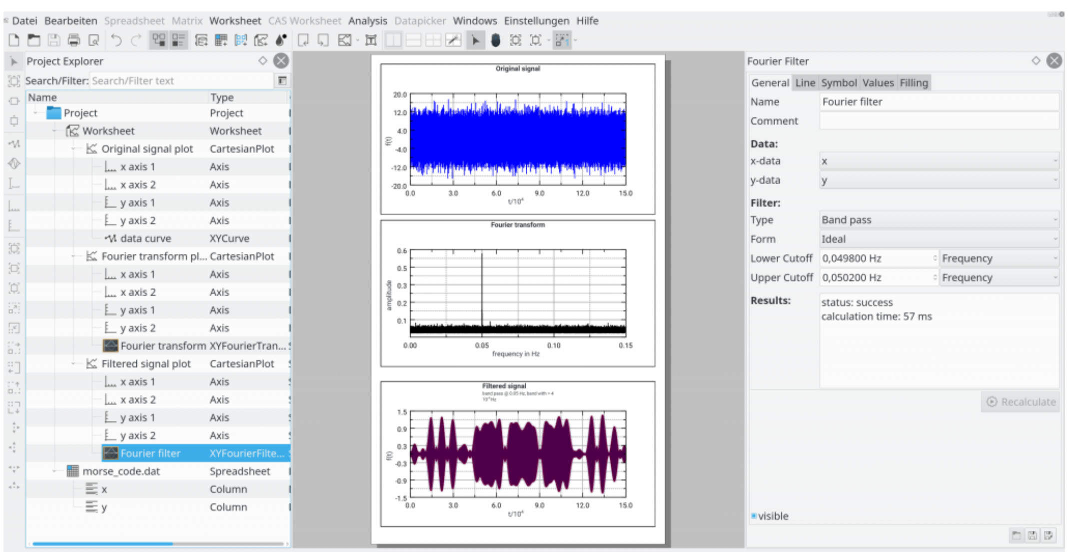
9.1 2D Plotting

Coming soon ...

9.2 Signal processing

Fourier filter

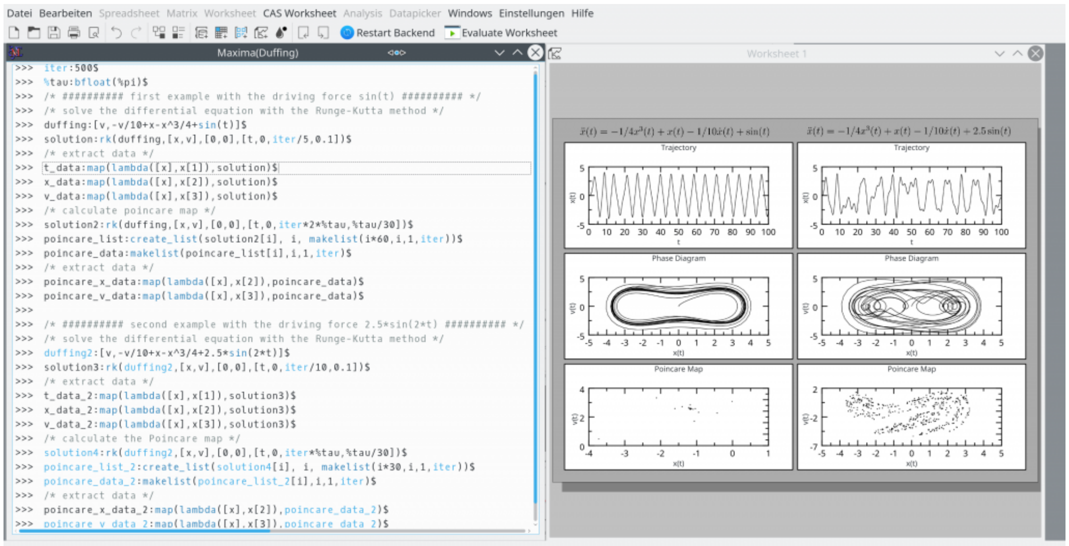
A time signal containing Morse code is Fourier transformed to frequency space to see the main component. By applying a narrow band pass filter the Morse signal is extracted and a nice 'SOS' can be seen:



9.3 Computing

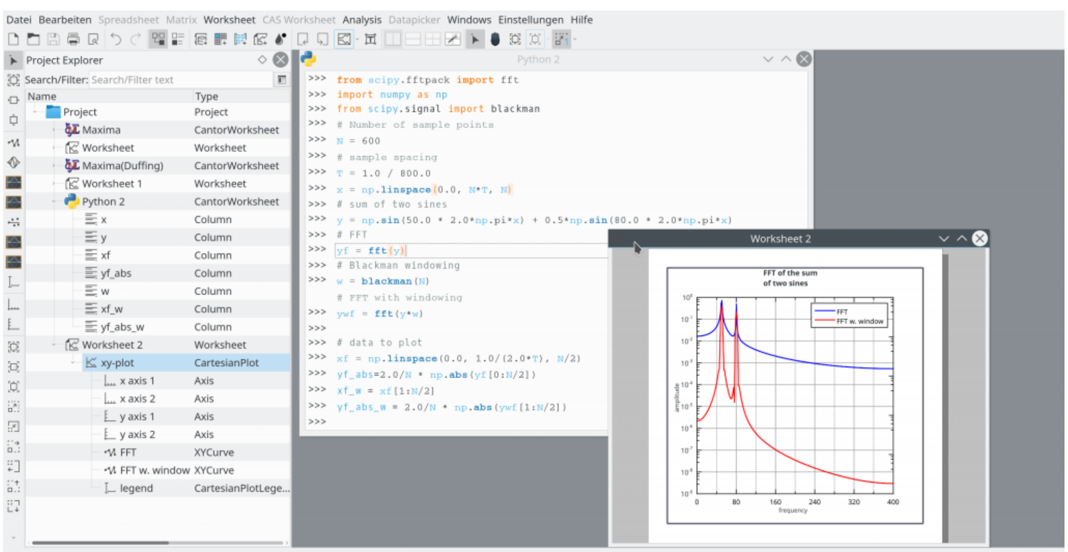
Maxima

Maxima session showing the chaotic dynamics of the Duffing oscillator. The differential equation of the forced oscillator are solved with Maxima. Plots of the trajectory, the phase space of the oscillator and the corresponding Poincaré map are done with LabPlot:



Python

Python session illustrating the effect of Blackman windowing on the Fourier transform:



9.4 Import/Export

Coming soon ...

9.5 Tools

Coming soon ...

Chapter 10

Parser functions

The LabPlot parser allows you to use following functions:

10.1 Standard functions

Function	Description
cbrt(x)	Cube root
ceil(x)	Truncate upward to integer
fabs(x)	Absolute value
gamma(x)	Gamma function
ldexp(x,y)	$x * 2^y$
ln(x)	Logarithm, base e
log(x)	Logarithm, base e
log1p(x)	log(1+x)
log10(x)	Logarithm, base 10
logb(x)	Radix-independent exponent
pow(x,n)	power function x^n
powint(x,n)	integer power function x^n
pow2(x)	power function x^2
pow3(x)	power function x^3
pow4(x)	power function x^4
pow5(x)	power function x^5
pow6(x)	power function x^6
pow7(x)	power function x^7
pow8(x)	power function x^8
pow9(x)	power function x^9
rint(x)	round to nearest integer
round(x)	round to nearest integer
sqrt(x)	Square root
tgamma(x)	Gamma function
trunc(x)	Returns the greatest integer less than or equal to x

10.2 Trigonometric functions

Function	Description
sin(x)	Sine
cos(x)	Cosine
tan(x)	Tangent
asin(x)	Inverse sine
acos(x)	Inverse cosine
atan(x)	Inverse tangent
atan2(y,x)	Inverse tangent function of two variables
sinh(x)	Hyperbolic sine
cosh(x)	Hyperbolic cosine
tanh(x)	Hyperbolic tangent
asinh(x)	Inverse hyperbolic sine
acosh(x)	Inverse hyperbolic cosine
atanh(x)	Inverse hyperbolic tangent
sec(x)	Secant
csc(x)	Cosecant
cot(x)	Cotangent
asec(x)	Inverse secant
acsc(x)	Inverse cosecant
acot(x)	Inverse cotangent
sech(x)	Hyperbolic secant
csch(x)	Hyperbolic cosecant
coth(x)	Hyperbolic cotangent
asech(x)	Inverse hyperbolic secant
acsch(x)	Inverse hyperbolic cosecant
acoth(x)	Inverse hyperbolic cotangent
sinc(x)	Sinc function sin(π x) / (π x)
logsinh(x)	log(sinh(x)) for x > 0
logcosh(x)	log(cosh(x))
hypot(x,y)	Hypotenuse function $\sqrt{}\{x^2 + y^2\}$
hypot3(x,y,z)	$\sqrt{}\{x^2 + y^2 + z^2\}$
anglesymm(α)	force the angle α to lie in the range (-π,π]
anglepos(α)	force the angle α to lie in the range (0,2π]

10.3 Special functions

For more information about the functions see the documentation of GSL.

Function	Description
Ai(x)	Airy function Ai(x)
Bi(x)	Airy function Bi(x)
Ais(x)	scaled version of the Airy function $S_{Ai}(x)$
Bis(x)	scaled version of the Airy function $S_{Bi}(x)$
Aid(x)	Airy function derivative Ai′(x)
Bid(x)	Airy function derivative Bi′(x)
Aids(x)	derivative of the scaled Airy function $S_{Ai}(x)$

Bids(x)	derivative of the scaled Airy function $S_{Bi}(x)$
Ai0(s)	s-th zero of the Airy function Ai(x)
Bi0(s)	s-th zero of the Airy function Bi(x)
Aid0(s)	s-th zero of the Airy function derivative Ai’(x)
Bid0(s)	s-th zero of the Airy function derivative Bi’(x)
J0(x)	regular cylindrical Bessel function of zeroth order, $J_0(x)$
J1(x)	regular cylindrical Bessel function of first order, $J_1(x)$
Jn(n,x)	regular cylindrical Bessel function of order n, $J_n(x)$
Y0(x)	irregular cylindrical Bessel function of zeroth order, $Y_0(x)$
Y1(x)	irregular cylindrical Bessel function of first order, $Y_1(x)$
Yn(n,x)	irregular cylindrical Bessel function of order n, $Y_n(x)$
I0(x)	regular modified cylindrical Bessel function of zeroth order, $I_0(x)$
I1(x)	regular modified cylindrical Bessel function of first order, $I_1(x)$
In(n,x)	regular modified cylindrical Bessel function of order n, $I_n(x)$
I0s(x)	scaled regular modified cylindrical Bessel function of zeroth order, exp $(-\|x\|)$ $I_0(x)$
I1s(x)	scaled regular modified cylindrical Bessel function of first order, exp$(-\|x\|)$ $I_1(x)$
Ins(n,x)	scaled regular modified cylindrical Bessel function of order n, exp$(-\|x\|)$ $I_n(x)$
K0(x)	irregular modified cylindrical Bessel function of zeroth order, $K_0(x)$
K1(x)	irregular modified cylindrical Bessel function of first order, $K_1(x)$
Kn(n,x)	irregular modified cylindrical Bessel function of order n, $K_n(x)$
K0s(x)	scaled irregular modified cylindrical Bessel function of zeroth order, exp(x) $K_0(x)$
K1s(x)	scaled irregular modified cylindrical Bessel function of first order, exp(x) $K_1(x)$
Kns(n,x)	scaled irregular modified cylindrical Bessel function of order n, exp(x) $K_n(x)$
j0(x)	regular spherical Bessel function of zeroth order, $j_0(x)$
j1(x)	regular spherical Bessel function of first order, $j_1(x)$
j2(x)	regular spherical Bessel function of second order, $j_2(x)$
jl(l,x)	regular spherical Bessel function of order l, $j_l(x)$
y0(x)	irregular spherical Bessel function of zeroth order, $y_0(x)$

y1(x)	irregular spherical Bessel function of first order, $y_1(x)$
y2(x)	irregular spherical Bessel function of second order, $y_2(x)$
yl(l,x)	irregular spherical Bessel function of order l, $y_l(x)$
i0s(x)	scaled regular modified spherical Bessel function of zeroth order, $\exp(-\|x\|)\ i_0(x)$
i1s(x)	scaled regular modified spherical Bessel function of first order, $\exp(-\|x\|)\ i_1(x)$
i2s(x)	scaled regular modified spherical Bessel function of second order, $\exp(-\|x\|)\ i_2(x)$
ils(l,x)	scaled regular modified spherical Bessel function of order l, $\exp(-\|x\|)\ i_l(x)$
k0s(x)	scaled irregular modified spherical Bessel function of zeroth order, $\exp(x)\ k_0(x)$
k1s(x)	scaled irregular modified spherical Bessel function of first order, $\exp(x)\ k_1(x)$
k2s(x)	scaled irregular modified spherical Bessel function of second order, $\exp(x)\ k_2(x)$
kls(l,x)	scaled irregular modified spherical Bessel function of order l, $\exp(x)\ k_l(x)$
Jnu(ν,x)	regular cylindrical Bessel function of fractional order ν, $J_\nu(x)$
Ynu(ν,x)	irregular cylindrical Bessel function of fractional order ν, $Y_\nu(x)$
Inu(ν,x)	regular modified Bessel function of fractional order ν, $I_\nu(x)$
Inus(ν,x)	scaled regular modified Bessel function of fractional order ν, $\exp(-\|x\|)\ I_\nu(x)$
Knu(ν,x)	irregular modified Bessel function of fractional order ν, $K_\nu(x)$
lnKnu(ν,x)	logarithm of the irregular modified Bessel function of fractional order ν,$\ln(K_\nu(x))$
Knus(ν,x)	scaled irregular modified Bessel function of fractional order ν, $\exp(\|x\|)\ K_\nu(x)$
J0_0(s)	s-th positive zero of the Bessel function $J_0(x)$
J1_0(s)	s-th positive zero of the Bessel function $J_1(x)$
Jnu_0(nu,s)	s-th positive zero of the Bessel function $J_\nu(x)$
clausen(x)	Clausen integral $Cl_2(x)$
hydrogenicR_1(Z,R)	lowest-order normalized hydrogenic bound state radial wavefunction $R_1 := 2Z \sqrt{Z} \exp(-Z r)$
hydrogenicR(n,l,Z,R)	n-th normalized hydrogenic bound state radial wavefunction
dawson(x)	Dawson’s integral
D1(x)	first-order Debye function $D_1(x) = (1/x) \int_0^x (t/(e^t - 1))\, dt$
D2(x)	second-order Debye function $D_2(x) = (2/x^2) \int_0^x (t^2/(e^t - 1))\, dt$
D3(x)	third-order Debye function $D_3(x) = (3/x^3) \int_0^x (t^3/(e^t - 1))\, dt$
D4(x)	fourth-order Debye function $D_4(x) = (4/x^4) \int_0^x (t^4/(e^t - 1))\, dt$

D5(x)	fifth-order Debye function $D_5(x) = (5/x^5) \int_0^x (t^5/(e^t - 1))\, dt$
D6(x)	sixth-order Debye function $D_6(x) = (6/x^6) \int_0^x (t^6/(e^t - 1))\, dt$
Li2(x)	dilogarithm
Kc(k)	complete elliptic integral K(k)
Ec(k)	complete elliptic integral E(k)
F(phi,k)	incomplete elliptic integral F(phi,k)
E(phi,k)	incomplete elliptic integral E(phi,k)
P(phi,k,n)	incomplete elliptic integral P(phi,k,n)
D(phi,k,n)	incomplete elliptic integral D(phi,k,n)
RC(x,y)	incomplete elliptic integral RC(x,y)
RD(x,y,z)	incomplete elliptic integral RD(x,y,z)
RF(x,y,z)	incomplete elliptic integral RF(x,y,z)
RJ(x,y,z)	incomplete elliptic integral RJ(x,y,z,p)
erf(x)	error function $\mathrm{erf}(x) = 2/\sqrt{\pi} \int_0^x \exp(-t^2)\, dt$
erfc(x)	complementary error function $\mathrm{erfc}(x) = 1 - \mathrm{erf}(x) = 2/\sqrt{\pi} \int_x^\infty \exp(-t^2)\, dt$
log_erfc(x)	logarithm of the complementary error function log(erfc(x))
erf_Z(x)	Gaussian probability function $Z(x) = (1/(2\pi)) \exp(-x^2/2)$
erf_Q(x)	upper tail of the Gaussian probability function $Q(x) = (1/(2\pi)) \int_x^\infty \exp(-t^2/2)\, dt$
hazard(x)	hazard function for the normal distribution
exp(x)	Exponential, base e
expm1(x)	exp(x)-1
exp_mult(x,y)	exponentiate x and multiply by the factor y to return the product y exp(x)
exprel(x)	(exp(x)-1)/x using an algorithm that is accurate for small x
exprel2(x)	$2(\exp(x)-1-x)/x^2$ using an algorithm that is accurate for small x
expreln(n,x)	n-relative exponential, which is the n-th generalization of the functions ‘exprel’
E1(x)	exponential integral $E_1(x)$, $E_1(x) := \mathrm{Re} \int_1^\infty \exp(-xt)/t\, dt$
E2(x)	second-order exponential integral $E_2(x)$, $E_2(x) := \mathrm{Re} \int_1^\infty \exp(-xt)/t^2\, dt$
En(x)	exponential integral E_n(x) of order n, $E_n(x) := \mathrm{Re} \int_1^\infty \exp(-xt)/t^n\, dt)$
Ei(x)	exponential integral E_i(x), $Ei(x) := PV(\int_{-x}^\infty \exp(-t)/t\, dt)$
shi(x)	$Shi(x) = \int_0^x \sinh(t)/t\, dt$
chi(x)	integral $Chi(x) := \mathrm{Re}[\gamma_E + \log(x) + \int_0^x (\cosh[t]-1)/t\, dt]$
Ei3(x)	exponential integral $Ei_3(x) = \int_0^x \exp(-t^3)\, dt$ for x >= 0
si(x)	Sine integral $Si(x) = \int_0^x \sin(t)/t\, dt$
ci(x)	Cosine integral $Ci(x) = -\int_x^\infty \cos(t)/t\, dt$ for x > 0
atanint(x)	Arctangent integral $AtanInt(x) = \int_0^x \arctan(t)/t\, dt$

Fm1(x)	complete Fermi-Dirac integral with an index of -1, $F_{-1}(x) = e^x / (1 + e^x)$
F0(x)	complete Fermi-Dirac integral with an index of 0, $F_0(x) = \ln(1 + e^x)$
F1(x)	complete Fermi-Dirac integral with an index of 1, $F_1(x) = \int_0^\infty (t /(\exp(t-x)+1))\, dt$
F2(x)	complete Fermi-Dirac integral with an index of 2, $F_2(x) = (1/2) \int_0^\infty (t^2 /(\exp(t-x)+1))\, dt$
Fj(j,x)	complete Fermi-Dirac integral with an index of j, $F_j(x) = (1/\Gamma(j+1)) \int_0^\infty (t^j /(\exp(t-x)+1))\, dt$
Fmhalf(x)	complete Fermi-Dirac integral $F_{-1/2}(x)$
Fhalf(x)	complete Fermi-Dirac integral $F_{1/2}(x)$
F3half(x)	complete Fermi-Dirac integral $F_{3/2}(x)$
Finc0(x,b)	incomplete Fermi-Dirac integral with an index of zero, $F_0(x,b) = \ln(1 + e^{b-x}) - (b-x)$
lngamma(x)	logarithm of the Gamma function
gammastar(x)	regulated Gamma Function $\Gamma^*(x)$ for $x > 0$
gammainv(x)	reciprocal of the gamma function, $1/\Gamma(x)$ using the real Lanczos method.
fact(n)	factorial n!
doublefact(n)	double factorial n!! = n(n-2)(n-4)...
lnfact(n)	logarithm of the factorial of n, log(n!)
lndoublefact(n)	logarithm of the double factorial log(n!!)
choose(n,m)	combinatorial factor ‘n choose m’ = n!/(m!(n-m)!)
lnchoose(n,m)	logarithm of ‘n choose m’
taylor(n,x)	Taylor coefficient x^n / n! for x >= 0, n >= 0
poch(a,x)	Pochhammer symbol $(a)_x := \Gamma(a + x)/\Gamma(x)$
lnpoch(a,x)	logarithm of the Pochhammer symbol $(a)_x := \Gamma(a + x)/\Gamma(x)$
pochrel(a,x)	relative Pochhammer symbol ((a,x) - 1)/x where $(a,x) = (a)_x := \Gamma(a + x)/\Gamma(a)$
gammainc(a,x)	incomplete Gamma Function $\Gamma(a,x) = \int_x^\infty t^{a-1} \exp(-t)\, dt$ for $a > 0$, $x >= 0$
gammaincQ(a,x)	normalized incomplete Gamma Function $P(a,x) = 1/\Gamma(a) \int_x^\infty t^{a-1} \exp(-t)\, dt$ for $a > 0$, $x >= 0$
gammaincP(a,x)	complementary normalized incomplete Gamma Function $P(a,x) = 1/\Gamma(a) \int_0^x t^{a-1} \exp(-t)\, dt$ for $a > 0$, $x >= 0$
beta(a,b)	Beta Function, $B(a,b) = \Gamma(a)\, \Gamma(b)/\Gamma(a+b)$ for $a > 0$, $b > 0$
lnbeta(a,b)	logarithm of the Beta Function, log(B(a,b)) for $a > 0$, $b > 0$
betainc(a,b,x)	normalize incomplete Beta function B_x(a,b)/B(a,b) for $a > 0$, $b > 0$
C1(λ,x)	Gegenbauer polynomial $C^\lambda_1(x)$
C2(λ,x)	Gegenbauer polynomial $C^\lambda_2(x)$
C3(λ,x)	Gegenbauer polynomial $C^\lambda_3(x)$
Cn(n,λ,x)	Gegenbauer polynomial $C^\lambda_n(x)$
hyperg_0F1(c,x)	hypergeometric function ${}_0F_1(c,x)$

hyperg_1F1i(m,n,x)	confluent hypergeometric function $_1F_1(m,n,x) = M(m,n,x)$ for integer parameters m, n
hyperg_1F1(a,b,x)	confluent hypergeometric function $_1F_1(a,b,x) = M(a,b,x)$ for general parameters a,b
hyperg_Ui(m,n,x)	confluent hypergeometric function U(m,n,x) for integer parameters m,n
hyperg_U(a,b,x)	confluent hypergeometric function U(a,b,x)
hyperg_2F1(a,b,c,x)	Gauss hypergeometric function $_2F_1(a,b,c,x)$
hyperg_2F1c(a_R,a_I,c,x)	Gauss hypergeometric function $_2F_1(a_R + i a_I, a_R - i a_I, c, x)$ with complex parameters
hyperg_2F1r(a_R,a_I,c,x)	renormalized Gauss hypergeometric function $_2F_1(a,b,c,x) / \Gamma(c)$
hyperg_2F1cr(a_R,a_I,c,x)	renormalized Gauss hypergeometric function $_2F_1(a_R + i a_I, a_R - i a_I, c, x) / \Gamma(c)$
hyperg_2F0(a,b,x)	hypergeometric function $_2F_0(a,b,x)$
L1(a,x)	generalized Laguerre polynomials $L^a{}_1(x)$
L2(a,x)	generalized Laguerre polynomials $L^a{}_2(x)$
L3(a,x)	generalized Laguerre polynomials $L^a{}_3(x)$
W0(x)	principal branch of the Lambert W function, $W_0(x)$
Wm1(x)	secondary real-valued branch of the Lambert W function, $W_{-1}(x)$
P1(x)	Legendre polynomials $P_1(x)$
P2(x)	Legendre polynomials $P_2(x)$
P3(x)	Legendre polynomials $P_3(x)$
Pl(l,x)	Legendre polynomials $P_l(x)$
Q0(x)	Legendre polynomials $Q_0(x)$
Q1(x)	Legendre polynomials $Q_1(x)$
Ql(l,x)	Legendre polynomials $Q_l(x)$
Plm(l,m,x)	associated Legendre polynomial $P_l{}^m(x)$
Pslm(l,m,x)	normalized associated Legendre polynomial $\sqrt{(2l+1)/(4\pi)} \sqrt{(l-m)!/(l+m)!} P_l{}^m(x)$ suitable for use in spherical harmonics
Phalf(λ,x)	irregular Spherical Conical Function $P^{1/2}{}_{-1/2 + i \lambda}(x)$ for $x > -1$
Pmhalf(λ,x)	regular Spherical Conical Function $P^{-1/2}{}_{-1/2 + i \lambda}(x)$ for $x > -1$
Pc0(λ,x)	conical function $P^0{}_{-1/2 + i \lambda}(x)$ for $x > -1$
Pc1(λ,x)	conical function $P^1{}_{-1/2 + i \lambda}(x)$ for $x > -1$
Psr(l,λ,x)	Regular Spherical Conical Function $P^{-1/2-l}{}_{-1/2 + i \lambda}(x)$ for $x > -1$, $l >= -1$
Pcr(l,λ,x)	Regular Cylindrical Conical Function $P^{-m}{}_{-1/2 + i \lambda}(x)$ for $x > -1$, $m >= -1$
H3d0(λ,η)	zeroth radial eigenfunction of the Laplacian on the 3-dimensional hyperbolic space, $L^{H3d}{}_0(\lambda,,\eta) := \sin(\lambda \eta)/(\lambda \sinh(\eta))$ for $\eta >= 0$
H3d1(λ,η)	zeroth radial eigenfunction of the Laplacian on the 3-dimensional hyperbolic space, $L^{H3d}{}_1(\lambda,\eta) := 1/\sqrt{\lambda^2 + 1} \sin(\lambda \eta)/(\lambda \sinh(\eta)) (\coth(\eta) - \lambda \cot(\lambda \eta))$ for $\eta >= 0$

H3d(l,λ,η)	L'th radial eigenfunction of the Laplacian on the 3-dimensional hyperbolic space eta >= 0, l >= 0
logabs(x)	logarithm of the magnitude of X, log(|x|)
logp(x)	log(1 + x) for x > -1 using an algorithm that is accurate for small x
logm(x)	log(1 + x) - x for x > -1 using an algorithm that is accurate for small x
psiint(n)	digamma function ψ(n) for positive integer n
psi(x)	digamma function ψ(n) for general x
psi1piy(y)	real part of the digamma function on the line 1+i y, Re[ψ(1 + i y)]
psi1int(n)	Trigamma function ψ'(n) for positive integer n
psi1(n)	Trigamma function ψ'(x) for general x
psin(m,x)	polygamma function $\psi^{(m)}(x)$ for m >= 0, x > 0
synchrotron1(x)	first synchrotron function $x \int_x^\infty K_{5/3}(t)\,dt$ for x >= 0
synchrotron2(x)	second synchrotron function $x K_{2/3}(x)$ for x >= 0
J2(x)	transport function J(2,x)
J3(x)	transport function J(3,x)
J4(x)	transport function J(4,x)
J5(x)	transport function J(5,x)
zetaint(n)	Riemann zeta function ζ(n) for integer n
zeta(s)	Riemann zeta function ζ(s) for arbitrary s
zetam1int(n)	Riemann ζ function minus 1 for integer n
zetam1(s)	Riemann ζ function minus 1
zetaintm1(s)	Riemann ζ function for integer n minus 1
hzeta(s,q)	Hurwitz zeta function ζ(s,q) for s > 1, q > 0
etaint(n)	eta function η(n) for integer n
eta(s)	eta function η(s) for arbitrary s

10.4 Random number distributions

For more information about the functions see the documentation of GSL.

Function	**Description**
gaussian(x,σ)	probability density p(x) for a Gaussian distribution with standard deviation σ
ugaussian(x)	unit Gaussian distribution. They are equivalent to the functions above with a standard deviation of σ = 1
gaussianP(x,σ)	cumulative distribution functions P(x) for the Gaussian distribution with standard deviation σ

gaussianQ(x,σ)	cumulative distribution functions Q(x) for the Gaussian distribution with standard deviation σ
gaussianPinv(P,σ)	inverse cumulative distribution functions P(x) for the Gaussian distribution with standard deviation σ
gaussianQinv(Q,σ)	inverse cumulative distribution functions Q(x) for the Gaussian distribution with standard deviation σ
ugaussianP(x)	cumulative distribution function P(x) for the unit Gaussian distribution
ugaussianQ(x)	cumulative distribution function Q(x) for the unit Gaussian distribution
ugaussianPinv(P)	inverse cumulative distribution function P(x) for the unit Gaussian distribution
ugaussianQinv(Q)	inverse cumulative distribution function Q(x) for the unit Gaussian distribution
gaussiantail(x,a,σ)	probability density p(x) for a Gaussian tail distribution with standard deviation σ and lower limit a
ugaussiantail(x,a)	tail of a unit Gaussian distribution. They are equivalent to the functions above with a standard deviation of σ = 1
gaussianbi(x,y,σ_x,σ_y,ρ)	probability density p(x,y) for a bivariate gaussian distribution with standard deviations σ_x, σ_y and correlation coefficient ρ
exponential(x,μ)	probability density p(x) for an exponential distribution with mean μ
exponentialP(x,μ)	cumulative distribution function P(x) for an exponential distribution with mean μ
exponentialQ(x,μ)	cumulative distribution function Q(x) for an exponential distribution with mean μ
exponentialPinv(P,μ)	inverse cumulative distribution function P(x) for an exponential distribution with mean μ
exponentialQinv(Q,μ)	inverse cumulative distribution function Q(x) for an exponential distribution with mean μ
laplace(x,a)	probability density p(x) for a Laplace distribution with width a
laplaceP(x,a)	cumulative distribution function P(x) for a Laplace distribution with width a
laplaceQ(x,a)	cumulative distribution function Q(x) for a Laplace distribution with width a
laplacePinv(P,a)	inverse cumulative distribution function P(x) for an Laplace distribution with width a
laplaceQinv(Q,a)	inverse cumulative distribution function Q(x) for an Laplace distribution with width a
exppow(x,a,b)	probability density p(x) for an exponential power distribution with scale parameter a and exponent b

exppowP(x,a,b)	cumulative probability density P(x) for an exponential power distribution with scale parameter a and exponent b
exppowQ(x,a,b)	cumulative probability density Q(x) for an exponential power distribution with scale parameter a and exponent b
cauchy(x,a)	probability density p(x) for a Cauchy (Lorentz) distribution with scale parameter a
cauchyP(x,a)	cumulative distribution function P(x) for a Cauchy distribution with scale parameter a
cauchyQ(x,a)	cumulative distribution function Q(x) for a Cauchy distribution with scale parameter a
cauchyPinv(P,a)	inverse cumulative distribution function P(x) for a Cauchy distribution with scale parameter a
cauchyQinv(Q,a)	inverse cumulative distribution function Q(x) for a Cauchy distribution with scale parameter a
rayleigh(x,σ)	probability density p(x) for a Rayleigh distribution with scale parameter σ
rayleighP(x,σ)	cumulative distribution function P(x) for a Rayleigh distribution with scale parameter σ
rayleighQ(x,σ)	cumulative distribution function Q(x) for a Rayleigh distribution with scale parameter σ
rayleighPinv(P,σ)	inverse cumulative distribution function P(x) for a Rayleigh distribution with scale parameter σ
rayleighQinv(Q,σ)	inverse cumulative distribution function Q(x) for a Rayleigh distribution with scale parameter σ
rayleigh_tail(x,a,σ)	probability density p(x) for a Rayleigh tail distribution with scale parameter σ and lower limit a
landau(x)	probability density p(x) for the Landau distribution
gammapdf(x,a,b)	probability density p(x) for a gamma distribution with parameters a and b
gammaP(x,a,b)	cumulative distribution function P(x) for a gamma distribution with parameters a and b
gammaQ(x,a,b)	cumulative distribution function Q(x) for a gamma distribution with parameters a and b
gammaPinv(P,a,b)	inverse cumulative distribution function P(x) for a gamma distribution with parameters a and b
gammaQinv(Q,a,b)	inverse cumulative distribution function Q(x) for a gamma distribution with parameters a and b

flat(x,a,b)	probability density p(x) for a uniform distribution from a to b
flatP(x,a,b)	cumulative distribution function P(x) for a uniform distribution from a to b
flatQ(x,a,b)	cumulative distribution function Q(x) for a uniform distribution from a to b
flatPinv(P,a,b)	inverse cumulative distribution function P(x) for a uniform distribution from a to b
flatQinv(Q,a,b)	inverse cumulative distribution function Q(x) for a uniform distribution from a to b
lognormal(x,ζ,σ)	probability density p(x) for a lognormal distribution with parameters ζ and σ
lognormalP(x,ζ,σ)	cumulative distribution function P(x) for a lognormal distribution with parameters ζ and σ
lognormalQ(x,ζ,σ)	cumulative distribution function Q(x) for a lognormal distribution with parameters ζ and σ
lognormalPinv(P,ζ,σ)	inverse cumulative distribution function P(x) for a lognormal distribution with parameters ζ and σ
lognormalQinv(Q,ζ,σ)	inverse cumulative distribution function Q(x) for a lognormal distribution with parameters ζ and σ
chisq(x,ν)	probability density p(x) for a χ^2 distribution with ν degrees of freedom
chisqP(x,ν)	cumulative distribution function P(x) for a χ^2 distribution with ν degrees of freedom
chisqQ(x,ν)	cumulative distribution function Q(x) for a χ^2 distribution with ν degrees of freedom
chisqPinv(P,ν)	inverse cumulative distribution function P(x) for a χ^2 distribution with ν degrees of freedom
chisqQinv(Q,ν)	inverse cumulative distribution function Q(x) for a χ^2 distribution with ν degrees of freedom
fdist(x,ν_1,ν_2)	probability density p(x) for an F-distribution with ν_1 and ν_2 degrees of freedom
fdistP(x,ν_1,ν_2)	cumulative distribution function P(x) for an F-distribution with ν_1 and ν_2 degrees of freedom
fdistQ(x,ν_1,ν_2)	cumulative distribution function Q(x) for an F-distribution with ν_1 and ν_2 degrees of freedom
fdistPinv(P,ν_1,ν_2)	inverse cumulative distribution function P(x) for an F-distribution with ν_1 and ν_2 degrees of freedom
fdistQinv(Q,ν_1,ν_2)	inverse cumulative distribution function Q(x) for an F-distribution with ν_1 and ν_2 degrees of freedom
tdist(x,ν)	probability density p(x) for a t-distribution with ν degrees of freedom

tdistP(x,ν)	cumulative distribution function P(x) for a t-distribution with ν degrees of freedom
tdistQ(x,ν)	cumulative distribution function Q(x) for a t-distribution with ν degrees of freedom
tdistPinv(P,ν)	inverse cumulative distribution function P(x) for a t-distribution with ν degrees of freedom
tdistQinv(Q,ν)	inverse cumulative distribution function Q(x) for a t-distribution with ν degrees of freedom
betapdf(x,a,b)	probability density p(x) for a beta distribution with parameters a and b
betaP(x,a,b)	cumulative distribution function P(x) for a beta distribution with parameters a and b
betaQ(x,a,b)	cumulative distribution function Q(x) for a beta distribution with parameters a and b
betaPinv(P,a,b)	inverse cumulative distribution function P(x) for a beta distribution with parameters a and b
betaQinv(Q,a,b)	inverse cumulative distribution function Q(x) for a beta distribution with parameters a and b
logistic(x,a)	probability density p(x) for a logistic distribution with scale parameter a
logisticP(x,a)	cumulative distribution function P(x) for a logistic distribution with scale parameter a
logisticQ(x,a)	cumulative distribution function Q(x) for a logistic distribution with scale parameter a
logisticPinv(P,a)	inverse cumulative distribution function P(x) for a logistic distribution with scale parameter a
logisticQinv(Q,a)	inverse cumulative distribution function Q(x) for a logistic distribution with scale parameter a
pareto(x,a,b)	probability density p(x) for a Pareto distribution with exponent a and scale b
paretoP(x,a,b)	cumulative distribution function P(x) for a Pareto distribution with exponent a and scale b
paretoQ(x,a,b)	cumulative distribution function Q(x) for a Pareto distribution with exponent a and scale b
paretoPinv(P,a,b)	inverse cumulative distribution function P(x) for a Pareto distribution with exponent a and scale b
paretoQinv(Q,a,b)	inverse cumulative distribution function Q(x) for a Pareto distribution with exponent a and scale b
weibull(x,a,b)	probability density p(x) for a Weibull distribution with scale a and exponent b
weibullP(x,a,b)	cumulative distribution function P(x) for a Weibull distribution with scale a and exponent b

weibullQ(x,a,b)	cumulative distribution function Q(x) for a Weibull distribution with scale a and exponent b
weibullPinv(P,a,b)	inverse cumulative distribution function P(x) for a Weibull distribution with scale a and exponent b
weibullQinv(Q,a,b)	inverse cumulative distribution function Q(x) for a Weibull distribution with scale a and exponent b
gumbel1(x,a,b)	probability density p(x) for a Type-1 Gumbel distribution with parameters a and b
gumbel1P(x,a,b)	cumulative distribution function P(x) for a Type-1 Gumbel distribution with parameters a and b
gumbel1Q(x,a,b)	cumulative distribution function Q(x) for a Type-1 Gumbel distribution with parameters a and b
gumbel1Pinv(P,a,b)	inverse cumulative distribution function P(x) for a Type-1 Gumbel distribution with parameters a and b
gumbel1Qinv(Q,a,b)	inverse cumulative distribution function Q(x) for a Type-1 Gumbel distribution with parameters a and b
gumbel2(x,a,b)	probability density p(x) at X for a Type-2 Gumbel distribution with parameters A and B
gumbel2P(x,a,b)	cumulative distribution function P(x) for a Type-2 Gumbel distribution with parameters a and b
gumbel2Q(x,a,b)	cumulative distribution function Q(x) for a Type-2 Gumbel distribution with parameters a and b
gumbel2Pinv(P,a,b)	inverse cumulative distribution function P(x) for a Type-2 Gumbel distribution with parameters a and b
gumbel2Qinv(Q,a,b)	inverse cumulative distribution function Q(x) for a Type-2 Gumbel distribution with parameters a and b
poisson(k,μ)	probability p(k) of obtaining k from a Poisson distribution with mean μ
poissonP(k,μ)	cumulative distribution functions P(k) for a Poisson distribution with mean μ
poissonQ(k,μ)	cumulative distribution functions Q(k) for a Poisson distribution with mean μ
bernoulli(k,p)	probability p(k) of obtaining k from a Bernoulli distribution with probability parameter p
binomial(k,p,n)	probability p(k) of obtaining p from a binomial distribution with parameters p and n
binomialP(k,p,n)	cumulative distribution functions P(k) for a binomial distribution with parameters p and n

binomialQ(k,p,n)	cumulative distribution functions Q(k) for a binomial distribution with parameters p and n
nbinomial(k,p,n)	probability p(k) of obtaining k from a negative binomial distribution with parameters p and n
nbinomialP(k,p,n)	cumulative distribution functions P(k) for a negative binomial distribution with parameters p and n
nbinomialQ(k,p,n)	cumulative distribution functions Q(k) for a negative binomial distribution with parameters p and n
pascal(k,p,n)	probability p(k) of obtaining k from a Pascal distribution with parameters p and n
pascalP(k,p,n)	cumulative distribution functions P(k) for a Pascal distribution with parameters p and n
pascalQ(k,p,n)	cumulative distribution functions Q(k) for a Pascal distribution with parameters p and n
geometric(k,p)	probability p(k) of obtaining k from a geometric distribution with probability parameter p
geometricP(k,p)	cumulative distribution functions P(k) for a geometric distribution with parameter p
geometricQ(k,p)	cumulative distribution functions Q(k) for a geometric distribution with parameter p
hypergeometric(k,n_1,n_2,t)	probability p(k) of obtaining k from a hypergeometric distribution with parameters n_1, n_2, t
hypergeometricP(k,n_1,n_2,t)	cumulative distribution function P(k) for a hypergeometric distribution with parameters n_1, n_2, t
hypergeometricQ(k,n_1,n_2,t)	cumulative distribution function Q(k) for a hypergeometric distribution with parameters n_1, n_2, t
logarithmic(k,p)	probability p(k) of obtaining K from a logarithmic distribution with probability parameter p

10.5 Constants

Constant	**Description**
e	The base of natural logarithms
pi	π

10.6 GSL constants

For more information about this constants see the documentation of GSL.

Constant	Description
c	The speed of light in vacuum
mu0	The permeability of free space
e0	The permittivity of free space
h	The Planck constant h
hbar	The reduced Planck constant ℏ
na	Avogadro's number
f	The molar charge of 1 Faraday
k	The Boltzmann constant
r0	The molar gas constant
v0	The standard gas volume
sigma	The Stefan–Boltzmann constant
gauss	The magnetic field of 1 Gauss
au	The length of 1 astronomical unit (mean earth-sun distance)
G	The gravitational constant
ly	The distance of 1 light-year
pc	The distance of 1 parsec
gg	The standard gravitational acceleration on Earth
ms	The mass of the Sun
ee	The charge of the electron
eV	The energy of 1 electron volt
amu	The unified atomic mass
me	The mass of the electron
mmu	The mass of the muon
mp	The mass of the proton
mn	The mass of the neutron
alpha	The electromagnetic fine structure constant
ry	The Rydberg constant
a0	The Bohr radius
a	The length of 1 angstrom
barn	The area of 1 barn
muB	The Bohr Magneton
mun	The Nuclear Magneton
mue	The magnetic moment of the electron
mup	The magnetic moment of the proton
sigmaT	The Thomson cross section for an electron
pD	The debye
min	The number of seconds in 1 minute
h	The number of seconds in 1 hour
d	The number of seconds in 1 day
week	The number of seconds in 1 week
in	The length of 1 inch
ft	The length of 1 foot
yard	The length of 1 yard
mil	The length of 1 mil (1/1000th of an inch)
v_km_per_h	The speed of 1 kilometer per hour
v_mile_per_h	The speed of 1 mile per hour

nmile	The length of 1 nautical mile
fathom	The length of 1 fathom
knot	The speed of 1 knot
pt	The length of 1 printer’s point (1/72 inch)
texpt	The length of 1 TeX point (1/72.27 inch)
micron	The length of 1 micrometre
hectare	The area of 1 hectare
acre	The area of 1 acre
liter	The volume of 1 liter
us_gallon	The volume of 1 US gallon
can_gallon	The volume of 1 Canadian gallon
uk_gallon	The volume of 1 UK gallon
quart	The volume of 1 quart
pint	The volume of 1 pint
pound	The mass of 1 pound
ounce	The mass of 1 ounce
ton	The mass of 1 ton
mton	The mass of 1 metric ton (1000 kg)
uk_ton	The mass of 1 UK ton
troy_ounce	The mass of 1 troy ounce
carat	The mass of 1 carat
gram_force	The force of 1 gram weight
pound_force	The force of 1 pound weight
kilepound_force	The force of 1 kilopound weight
poundal	The force of 1 poundal
cal	The energy of 1 calorie
btu	The energy of 1 British Thermal Unit
therm	The energy of 1 Therm
hp	The power of 1 horsepower
bar	The pressure of 1 bar
atm	The pressure of 1 standard atmosphere
torr	The pressure of 1 torr
mhg	The pressure of 1 meter of mercury
inhg	The pressure of 1 inch of mercury
inh2o	The pressure of 1 inch of water
psi	The pressure of 1 pound per square inch
poise	The dynamic viscosity of 1 poise
stokes	The kinematic viscosity of 1 stokes
stilb	The luminance of 1 stilb
lumen	The luminous flux of 1 lumen
lux	The illuminance of 1 lux
phot	The illuminance of 1 phot
ftcandle	The illuminance of 1 footcandle
lambert	The luminance of 1 lambert
ftlambert	The luminance of 1 footlambert
curie	The activity of 1 curie
roentgen	The exposure of 1 roentgen
rad	The absorbed dose of 1 rad
N	The force of 1 newton
dyne	The force of 1 dyne
J	The energy of 1 joule
erg	The energy of 1 erg

Chapter 11

Questions and Answers

1. *For which platforms is LabPlot available?*

   LabPlot is developed for Unix platforms and uses the Qt™ toolkit and KDE Frameworks. Normally you can expect LabPlot to build and run on every platform KDE Frameworks supports. A recent list of supported platforms and tips for compiling and running LabPlot can be found on http://labplot.wiki.sourceforge.net/Download.

2. *How do I export the active worksheet as image?*

   The standard way is to use **File** → **Export**. All Qt™ supported image formats are allowed. Just select the desired format and the active worksheet is exported.

3. *How do I use Greek letters for title, axes label, etc.?*

   Use π button to open character selector window or **TEX** to generate Greek letters and other symbols using LATEX.

4. *I miss an important feature. What can I do?*

   Please take a look at the TODO file in the documentation of LabPlot. Here, all planned features are listed in more or less sorted order which I will implement in future releases of LabPlot. If you like to have additional features or like to have a listed feature soon, mail me your wishes and, if possible, send me example data or a short description of what you like to do. It is not unlikely that your feature will appear in the next stable release of LabPlot :-)

5. *Many Analysis functions are disabled. What can I do?*

   It looks like your LabPlot package was compiled without GSL (GNU Scientific Library) support. LabPlot was designed to even work on systems that are missing most of the standard libraries. Many distributions are shipping LabPlot packages without this additional functionality. In this case some functions are not available. Fortunately some programs (like pstoedit or texvc) can be added without recompiling LabPlot. You can always check your system environment in the help menu of LabPlot.

   The packages provided on the official download page are always built with the standard libraries (GSL, etc.). You should use them to have all the features.

6. *I want to help. How can I contribute to LabPlot?*

   Yes, of course. There are a lot things to do. Even if you don't know anything about programming we always need people to find bugs, test things and make suggestions. Also the translation and documentation always needs a lot of work.

Chapter 12

License

LabPlot

IMPORTANT

LabPlot is still under development. There is a long list of missing features that will be implemented in later versions of LabPlot.

Because there are a lot things to do, developers need every help you can give. Any contribution like wishes, corrections, patches, bug reports or screen shots is welcome.