Csvgraph – a graph plotter for csv format files

Manual for Version 2.0 17/2/2021

Introduction

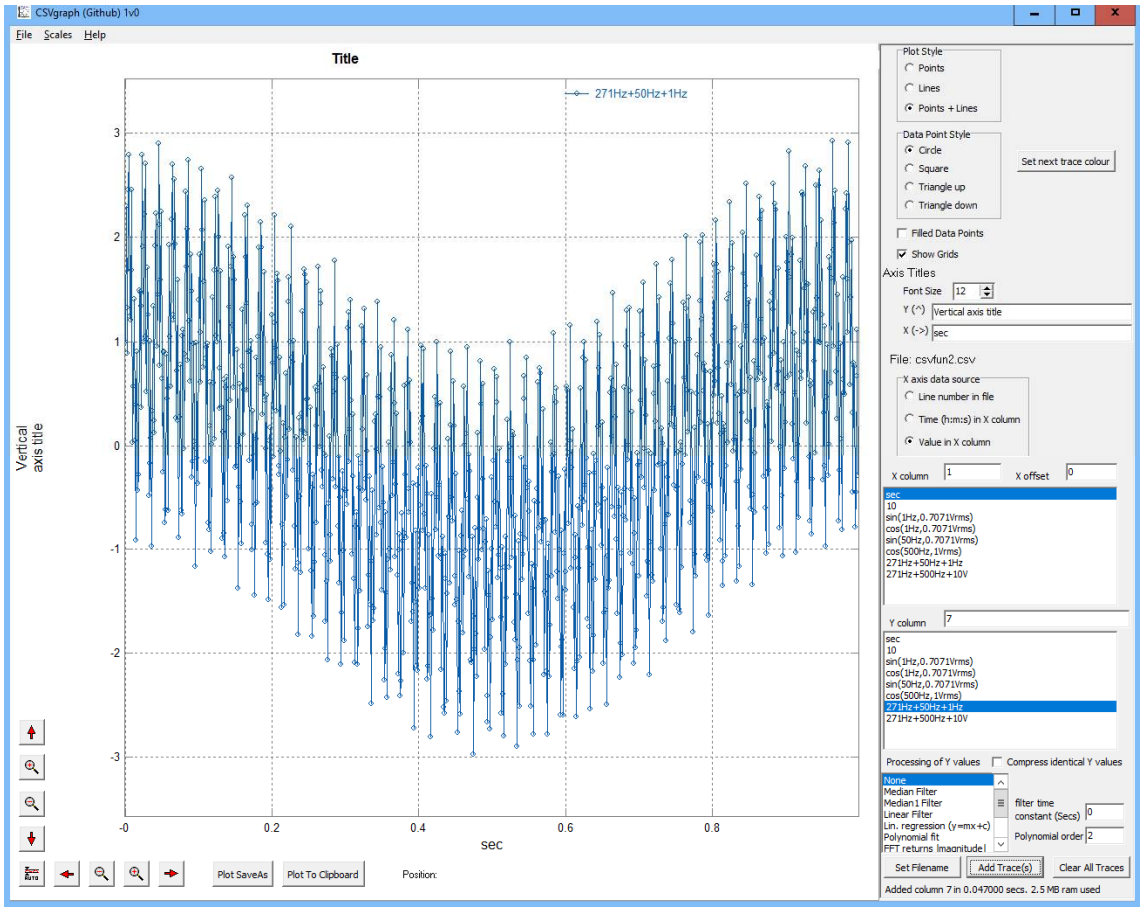
Csvgraph is designed to allow quick viewing of graphs of potentially very large (GB) csv files (for comparison most spreadsheets are limited to 1,048,576 rows). Csvgraph has no built-in limits, but ultimately it is limited by your available RAM (it will use up to 4GB of RAM if its available). Even with extremely large files reading is fast and zooming is normally instantaneous.

These csv files are assumed to have column headers on their first line so a typical csv file would start:

```
"Time(sec)","Col-2","Col-3","Col-4","Col-5"
99950,20,0,20,20
99950.1,10,1,11,12
```

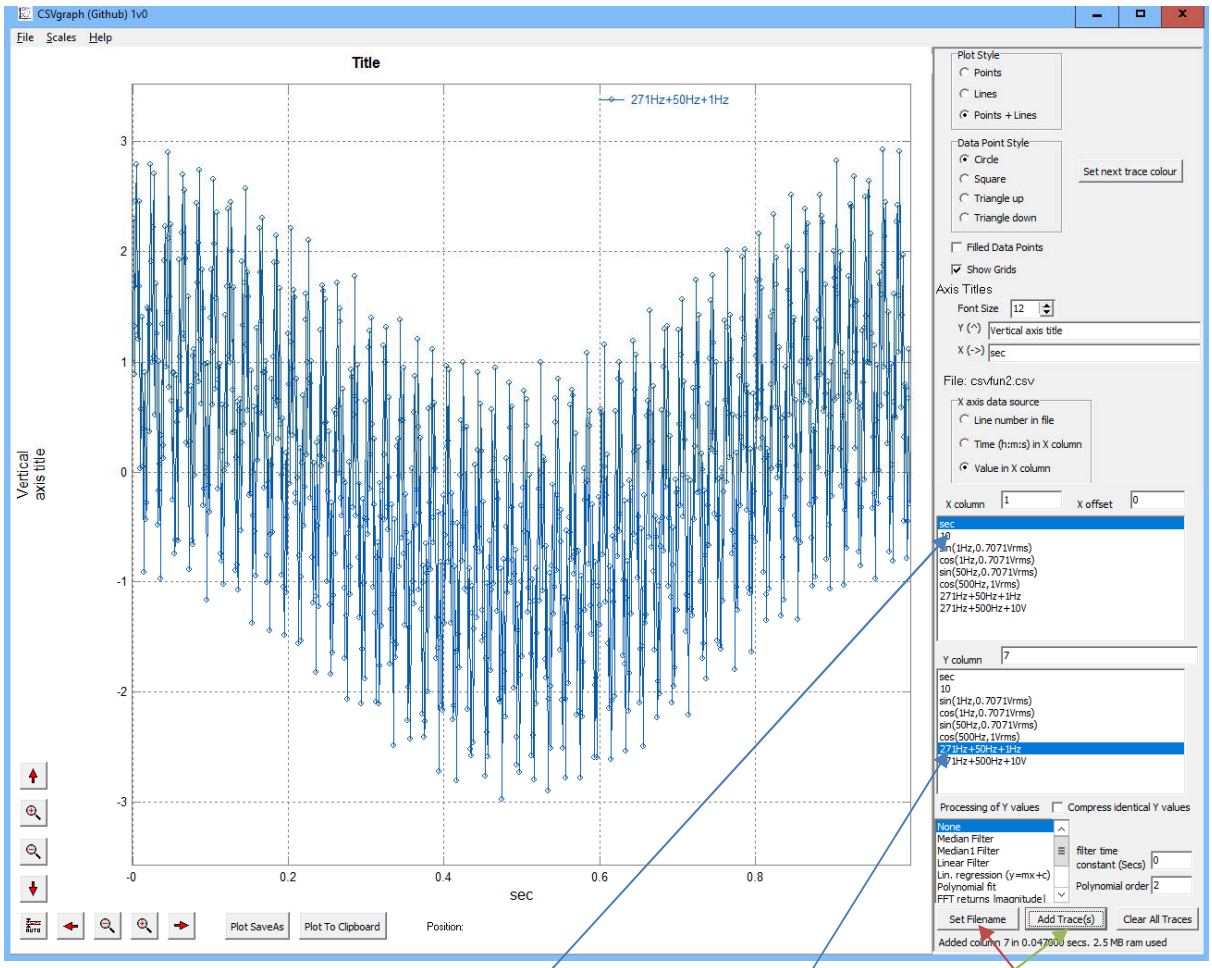
Values are read as floating-point numbers so are restricted to numbers between +/-3.4e+38 and the smallest non-zero number is approximately 1.4e-45, with approximately 7 significant digits.

The X values are assumed to be monotonically increasing, if they are not in the csv file then the X values (together with the corresponding Y value) are automatically sorted before they are displayed.



Use

Run csvgraph.exe by double clicking on it.

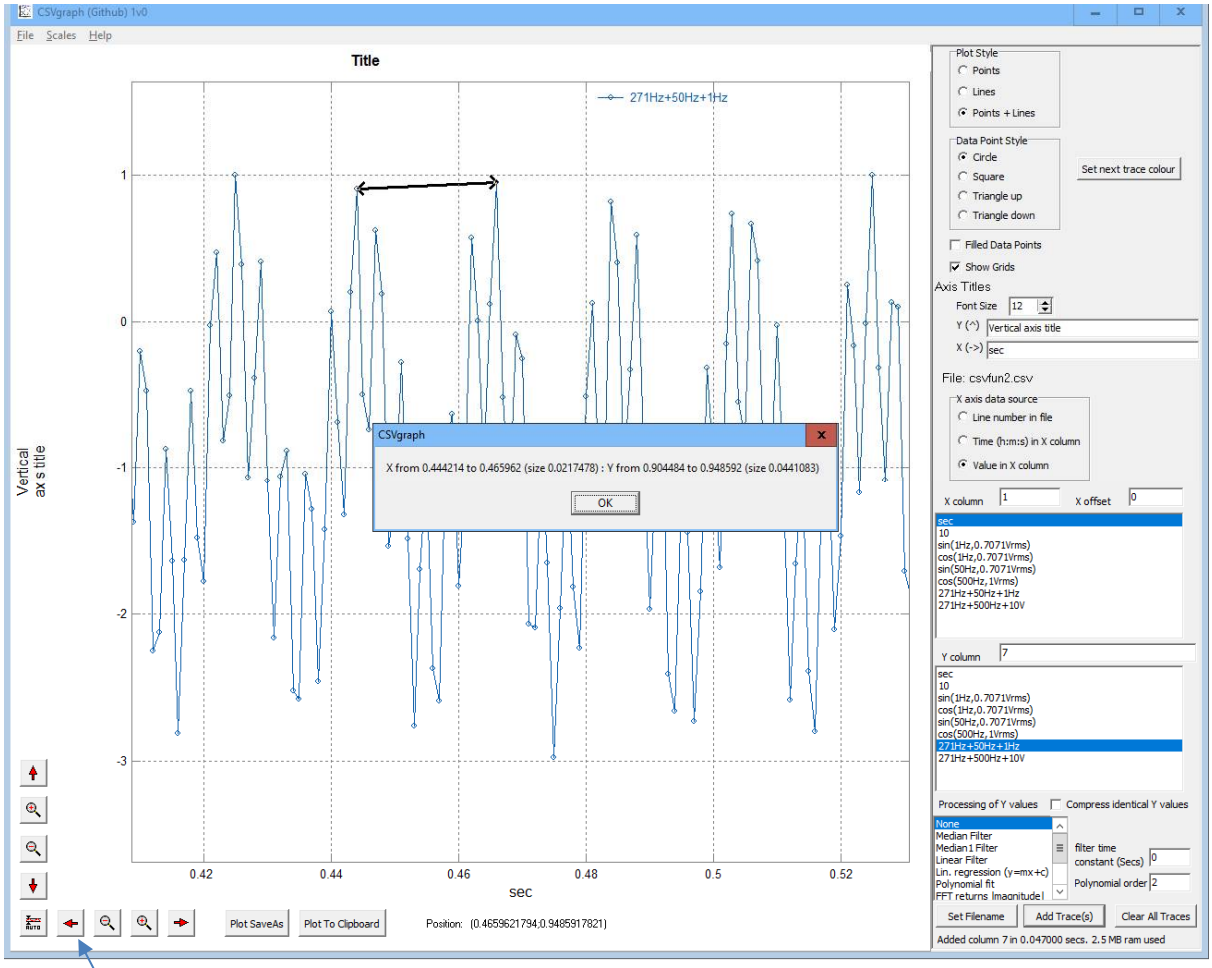


Select a suitable csv file (there are few examples in the archive and csvfun2.csv was used for the screenshot) and drag the file onto csvgraph (or select the file by pressing the "Set filename" button bottom right or use the menu, File, Open).

This should populate the Xcolumn (horizontal axis) and Ycolumn (vertical axis) boxes from the header row of the csv file as shown in the screen shot above. To add a trace to the graph, select one x column and one or more y columns and press the "Add Trace(s)" button on the bottom left.

Multiple traces can be added, and you can change the filename between traces if required by pressing the "Set Filename" button again before selecting X and Y columns and pressing the "Add Trace(s)" button.

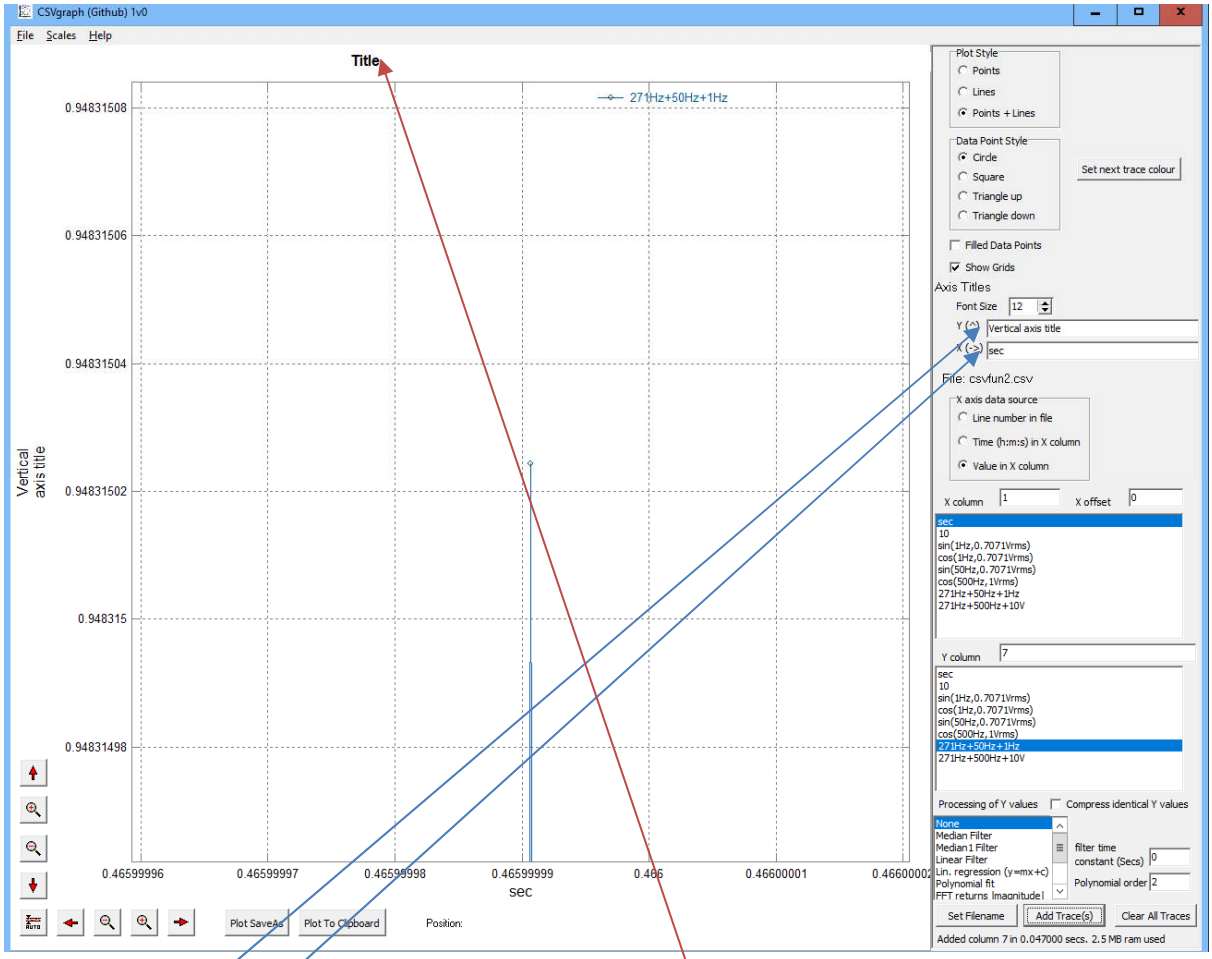
When a graph is displayed move the mouse over it and press the left mouse button, while keeping this button pressed move the mouse to select an area of the graph – when the mouse button is released the graph will zoom into the selected area. To restore the original view, press the middle mouse button. The right mouse button allows for measurements to be taken from the graph as shown below:



The mouse scroll wheel also allows quick zooming in and out on the graph.

The menu Scales option allows a specific area of the graph to be easily viewed, while the buttons on the bottom left allow the X and Y axes to be moved and zoomed independently.

There is no limit to zooming within the number range specified in the introduction – the graph below shows zooming in on a single point on the graph:



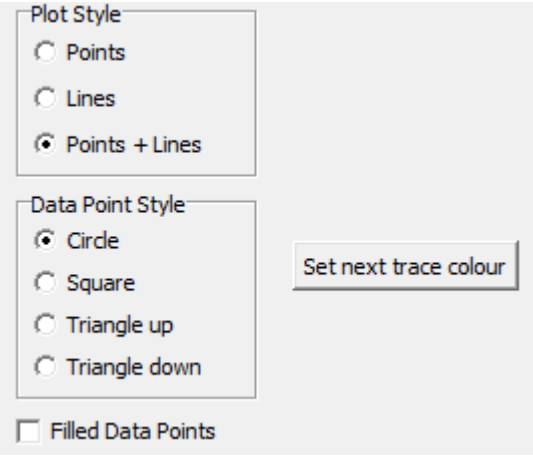
Adding Legends to a graph

A title can just be typed into the area by default labelled “Title”.

The y and x axis titles are entered into the boxes in the column on the right. Just above these the font size for these legends can be set.

The y (vertical) title can be split into 2 lines by including \n at the end of the first line e.g., “vertical\naxis title”.

Changing the format of the graph



By default, the graphs consist of points joined by lines; at the top of the right-hand column the plot style can be changed to display just points or just lines and the shape of the points and filled/not selected.

By default, additional traces will be given different colours automatically, but the colour can be set by pressing the “Set next trace colour” button if required.

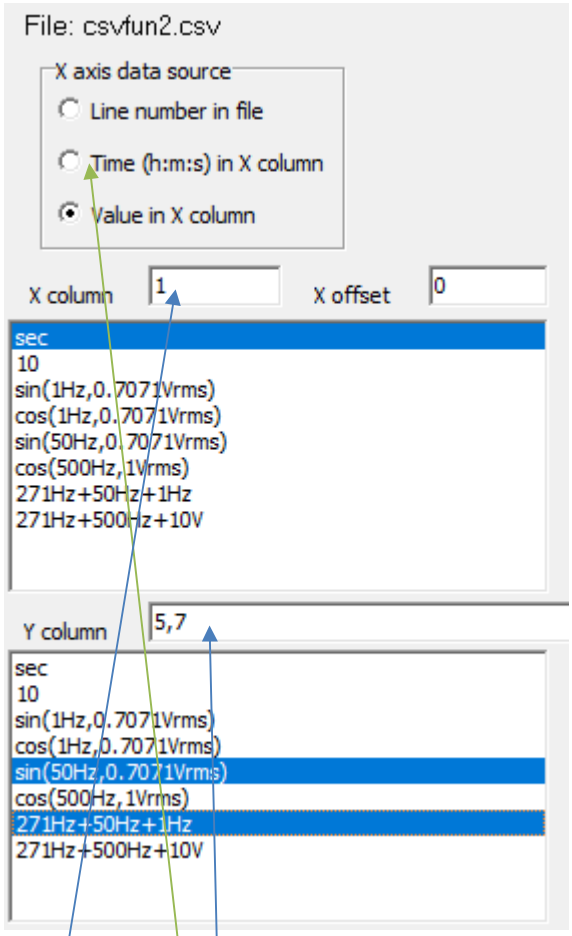
Saving graphs

Graphs can be copied to the clipboard by pressing the “Plot to Clipboard” button along the bottom of the screen, or by using the menu File/Save/Plot to clipboard.

Graphs can be saved to files by pressing the “Plot saveas” button on the bottom of the screen or using the menu File/Save/Save plot as. Plots can be saved as bmp, jpg, gif or png files. Note the graphs title is not saved (or copied to the clipboard) as in most uses this will be added as a caption to the graphic.

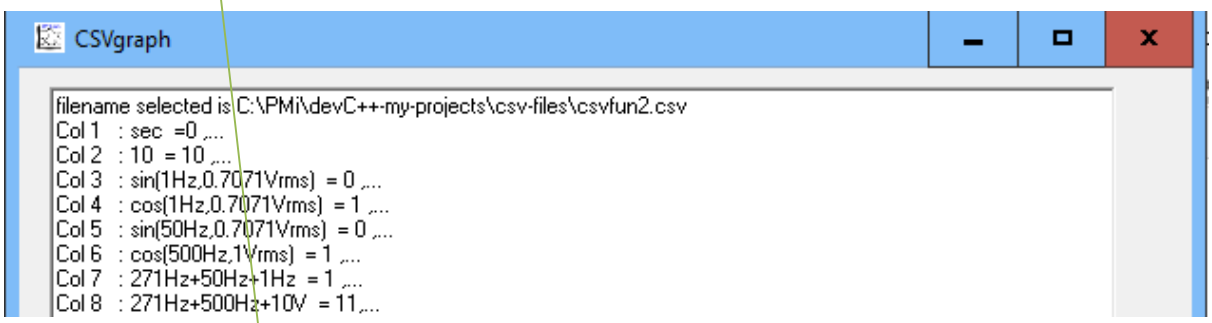
The data within a graph can be saved (which is useful if this has been calculated or filtered – see later for how to do this) by using the menu File/Save/Save Data as csv.

Selecting columns of the csv file



As seen previously the easiest way to select traces is to click on the names of the columns (shown in blue above). Multiple Y columns may be selected using shift and a left click or control and a left click of the mouse.

Alternatively, the columns may be selected by typing numbers into the areas to the right of the X column and Y column legends. As shown above multiple Y column numbers are separated by commas. The 1st column in the file is numbered 1, the second 2 etc – and these can be found from the 2nd csvgraph window as shown below:



It is also possible to select the x axis value to be the line number in the file, or to specify the value in the column is a time (h:m:s with an optional leading date e.g., 1/1/2020). If "time" is selected then time on the x-axis will start at zero and be in seconds (i.e., the first value read will be used as an offset for all future values). If the time increments past 23:59:59.9999 to 0:0:0 the x axis value will be

86400 (24 hours in seconds) rather than rolling back to zero so times longer than 1 day are automatically supported. Because of this action the actual date [if present] is ignored (that's partly because there are a large number of possible date formats in common use e.g., MM/DD/YY, DD/MM/YY, YY/MM/DD, Mon-DD-YYYY, DD-Mon-YYYY, etc and it's not possible for csvgraph to accurately guess which one has been used).

Note that numbers or times can be within double quotes in the csvfile (e.g., "12" will be read as 12).

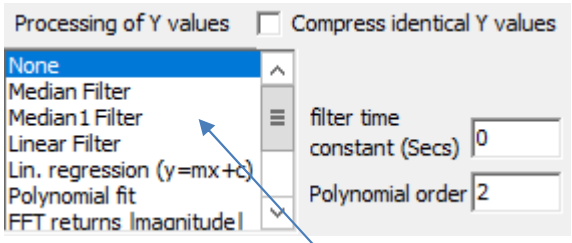
The y column value may also be described as an expression potentially combining the values from multiple columns e.g., $2-$3 would plot the difference between the 2nd and 3rd columns.

Allowable expressions are described in appendix A.

The Xoffset value (to the right of the X column) allows the x values of the most recently added trace to be moved left and right compared to previous traces (+positive numbers move right, negative move left). This can be useful to align traces.

Filtering

By default, traces are added without any processing (i.e., exactly as in the csv file), but csvgraph offers a number of options to "filter" the data before displaying it.



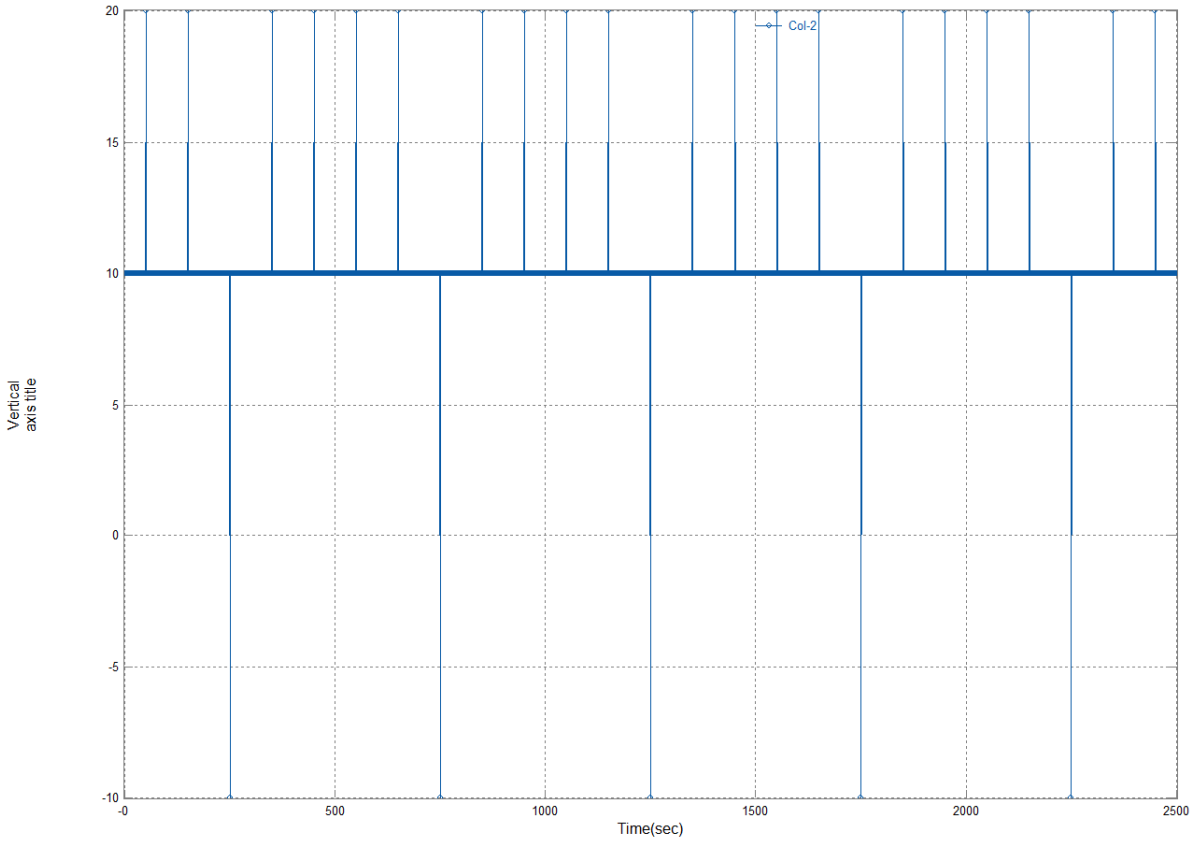
The simplest option which is unlikely to be needed unless you need to view extremely large files or have limited RAM is "Compress identical Y values". If this is ticked then sequences with identical Y values will be compressed (just the first and last point in a sequence kept) – the line graph will be identical with this option ticked. Note this is not true if filtering is applied as the filter values will only be calculated at stored points (csvgraph will remind you if you select "compress" and a filter).

Underneath this a range of filters can be selected and these are described below.

None

No filtering is done (this is the default) – the data is displayed exactly as it is in the csv file.

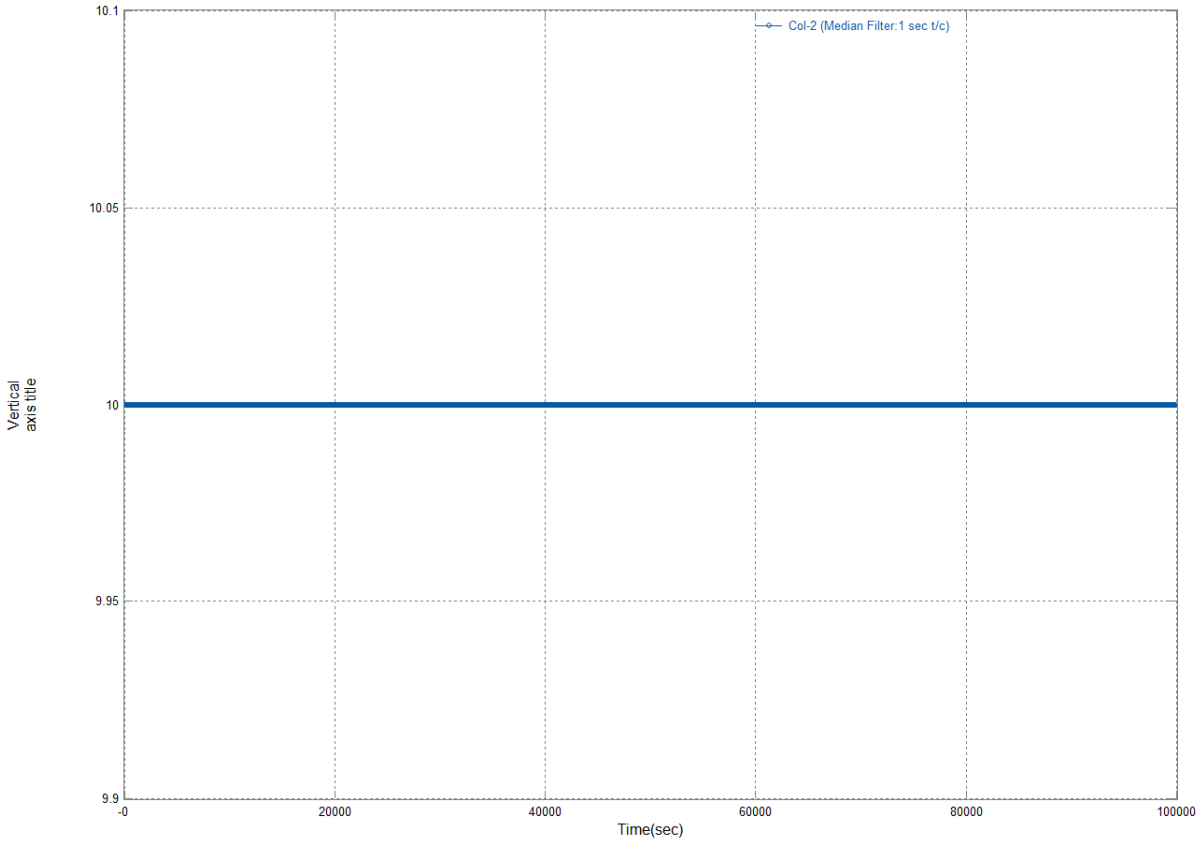
Looking at the example file demo1M.csv scaled with x from 0 to 2500 and Y from -10 to +20 (using menu/Scales to set these limits) gives:



Median/Median1 Filter

This applies an approximate median filter to each point which (approximately) takes the median of points +/- the specified filter time constant either side and plots this.

Using the same example data/scale as using in the “None” example above, with a 2 sec time constant, this gives:



If the spikes on the original were noise then this filter has completely removed them without impacting any of the other values.

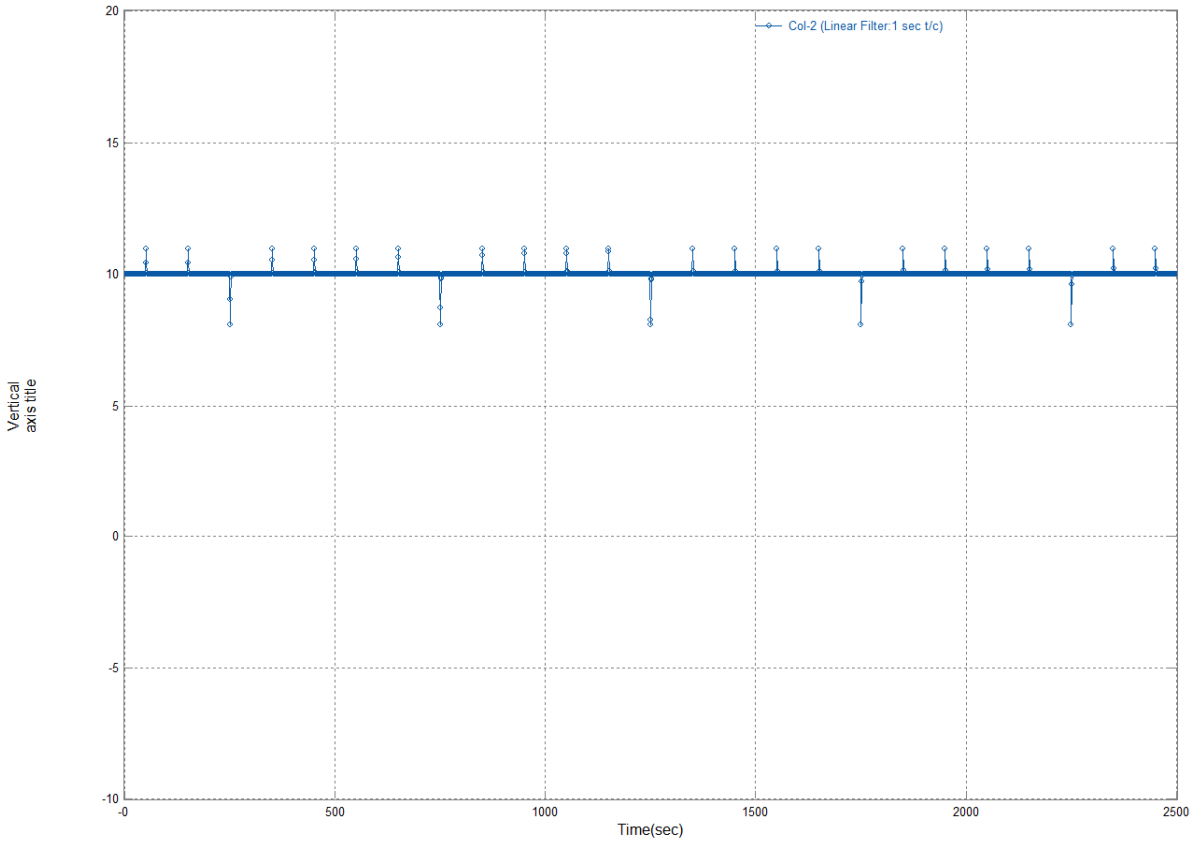
The difference between the Median and Median 1 filters is more subtle (they both give the same result in this example), the Median 1 filter gives a smoother change when the underlying data changes. Note there is no “lag” with a median filter (compare this to the linear filter below).

The algorithm used is very efficient so large filter times can be used if required, the (exact) median of the whole data can be found by supplying a very large value for the time constant.

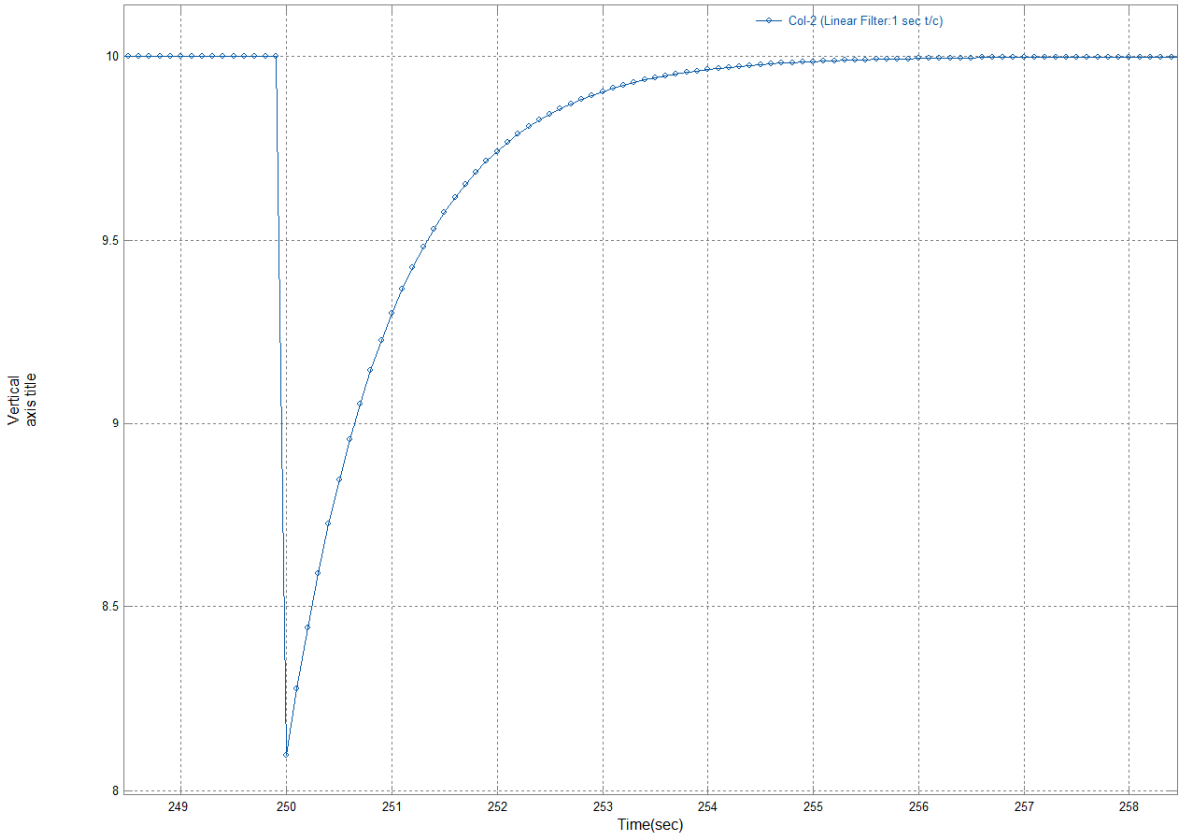
Linear Filter

This is a first order low pass filter with the time constant specified.

Using the same data as the Median filter above (2 sec time constant) gives:



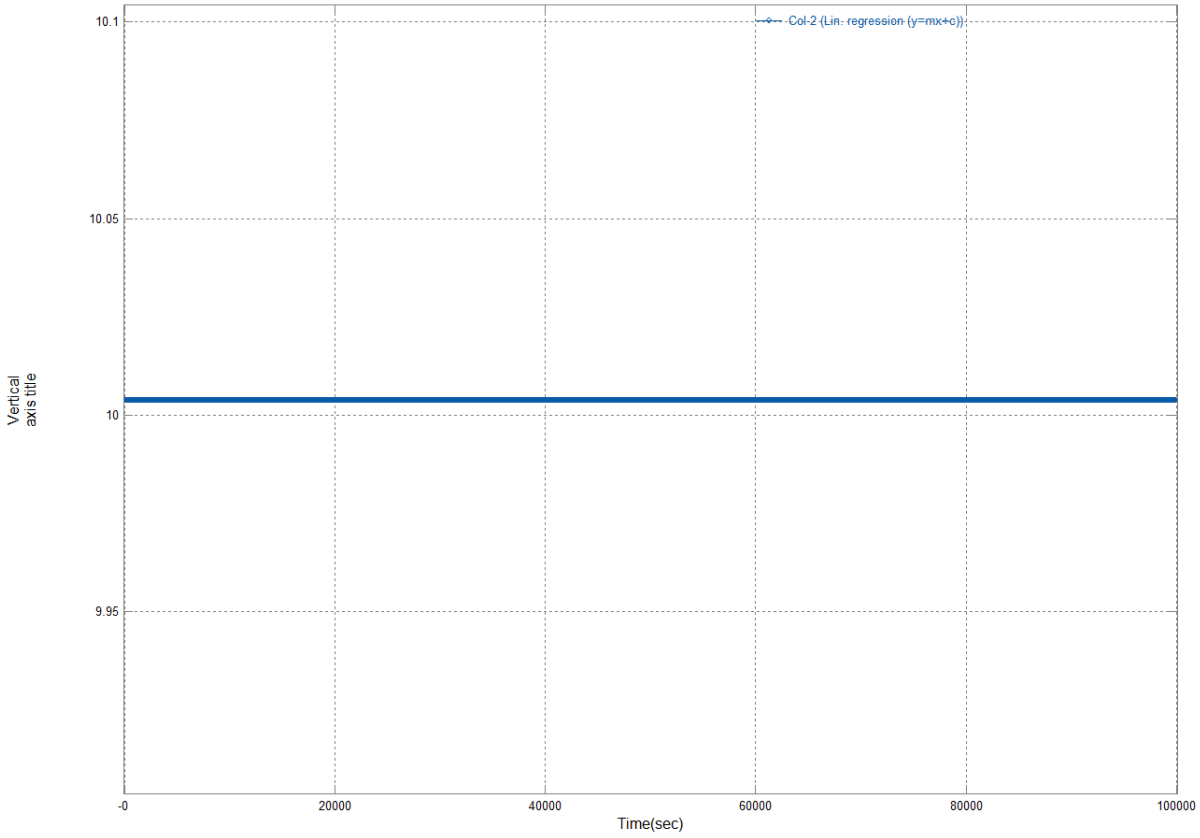
Zooming into one of the spikes shows that while this filter has reduced the height of the spikes it has also extended their duration:


Col-2 (Linear Filter:1 sec t/c)
Vertical axis title
10
9.5
9
8.5
8
249
250
251
252
253
254
255
256
257
258
Time(sec)

Linear Regression

Selecting this filter will result in a straight line which is the best (least squares) fit to the input data.

In the example above this gives:



If you look at the other csvgraph window you will see the text:

> Adding trace of Col-2 (col 2)
> vs Time(sec) (col 1)
> 1000000 lines read from csv file
> Best Least squares straight line is Y=2.38342e-13*X+10.004 which has an R^2 of 2.95898e-16
> Maximum value = 10.004 found on trace 0 (Col-2 (Lin. regression (y=mx+c))) at X=0
> Minimum value = 10.004 found on trace 0 (Col-2 (Lin. regression (y=mx+c))) at X=0
> Added column 2 in 0.750000 secs. 22.0 MB ram used

This gives the equation for the line (Y=2.38342e-13*X+10.004) and the R^2 value which varies between zero (meaning a poor match to the underlying data, which is the case here) and 1 (a very good match to the data).

In this case it has removed the spikes in a similar way as the median filter, but the resultant y value is a little different (here it is the average Y value (10.004) rather than the median (10.000)).

In version 1v3 and above there are two versions of Linear Regression, y=mx and y=mx+c.

The first (y=mx) forces the line to pass through the origin (x=0,y=0) the second y=mx+c is the more general version.

Least squares linear regression minimises the sum of squared errors between the measurements and the fitted straight line. So, if the correct value is 10 and the estimated value is 12 the squared error is 4.

GMR regression (y=mx+c)

This is another technique to fit a straight line to the supplied data.

Geometric Mean Regression (GMR) is also called Triangular regression. This method is less sensitive to outliers in the data than the (least squares) Linear regression method above.

This method minimises the sum of the areas of the right-angle triangles between the measurements and the fitted straight line.

Minimum absolute error for y=mx+c

This is another method to fit a straight line to data, it minimises the maximum absolute error and as a secondary function when the minimum error is reached it then minimises the sum of the absolute errors (this means the best line is normally unique; if the maximum absolute error only was minimised then typically a large set of lines would have the same maximum absolute error). For example, if the correct value is 10 and the estimated value is 12 the absolute error is 2.

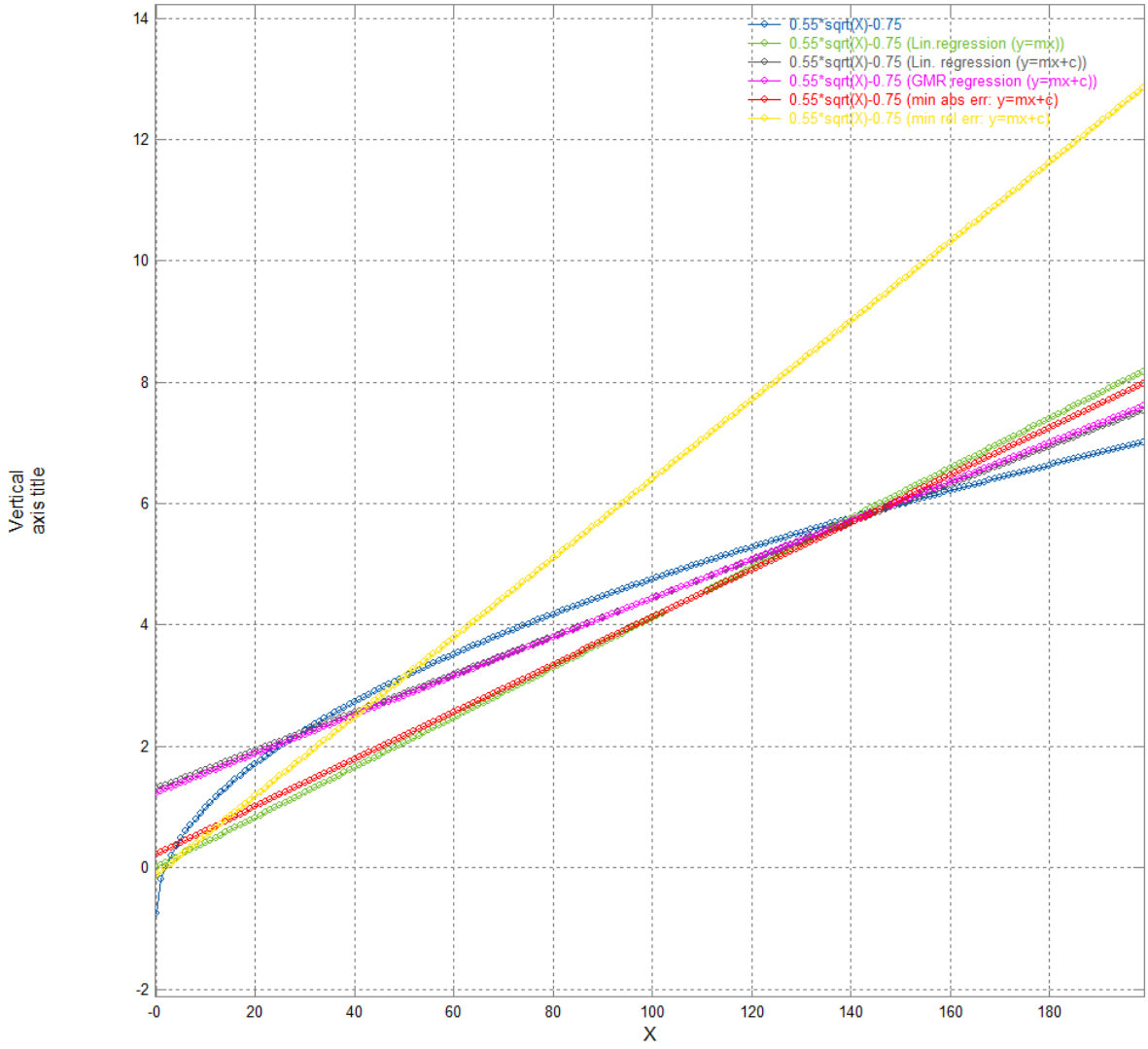
Minimum relative error for y=mx+c

This is another method to fit a straight line to data, it minimises the maximum relative error and as a secondary function when the minimum error is reached it then minimises the sum of the absolute relative errors. For example, if the correct value is 10 and the estimated value is 12 the relative error is 2/10 (0.2 or 20%). Relative error may give a better fit if the y values change a lot, for example if y values range from 1 to 1000 then an absolute error of 1 is 100% of the lowest y value (1) but only 0.1% of the largest y value (1000). Using relative errors than a relative error of 1/100 (1%) would be an absolute error 0.01 when y is 1 and an absolute error of 10 when y is 1000.

Which straight line fit should I use?

Each method has its place which is why csvgraph supports 5 ways to fit a straight line (4 techniques and one option to force the line to pass through the origin).

The plot below shows an example with all 5 options:



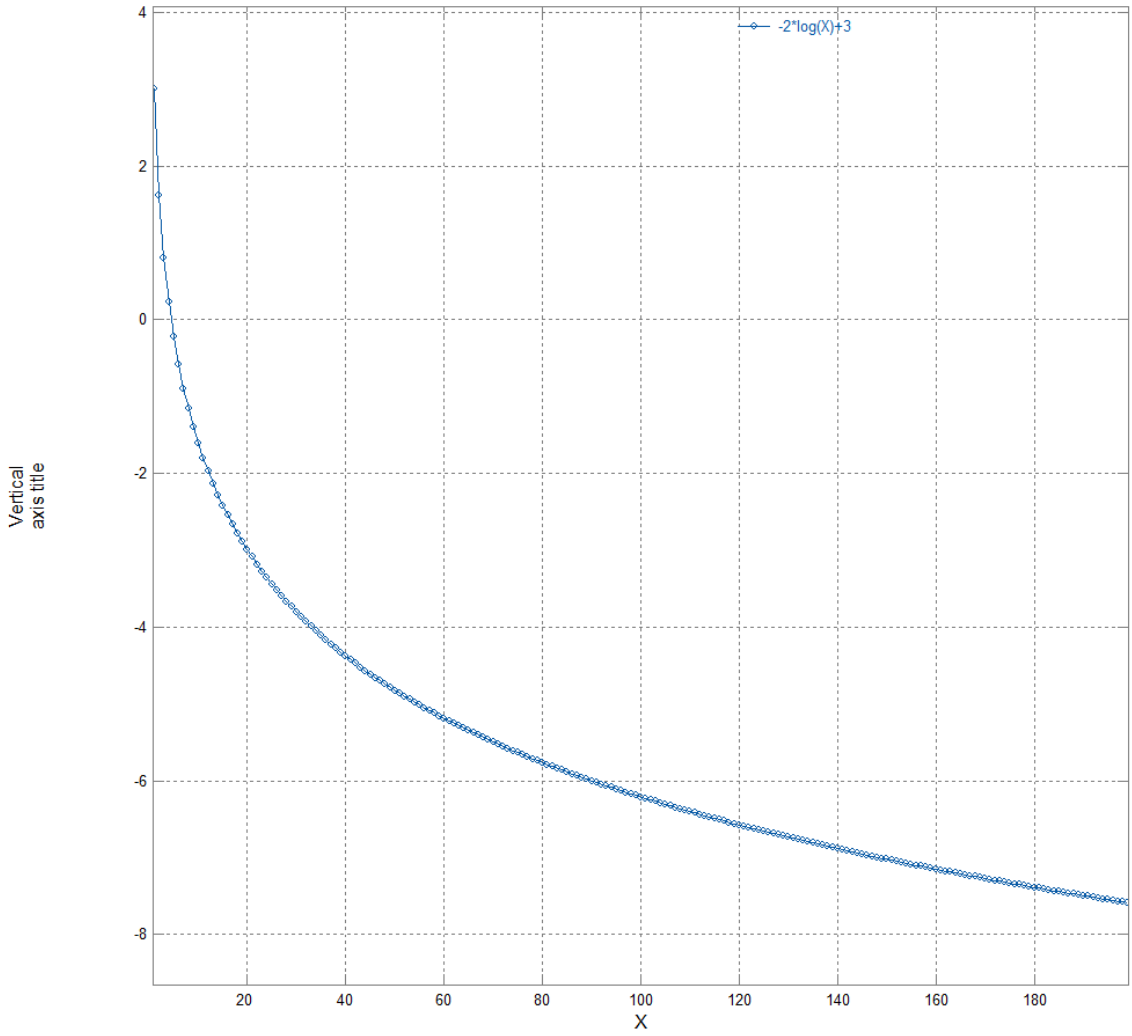
The blue curve on the graph above is what the other lines are trying to approximate. The minimum absolute error line (red) normally has its maximum absolute error at both ends and "the middle", while the minimum relative error line (yellow) will have a steadily increasing absolute error (as it is trying to keep the relative error constant). The GMR line (purple) is normally reasonably similar to the linear regression line (least squares) (black) and that's true here – and both will follow the curve reasonably closely giving a compromise between the absolute and relative errors. For this example, in terms of reducing maximum absolute errors, the minimum relative error line gives 5.8, the linear regression (y=mx+c) gives 2.05, GMR gives 1.98, linear regression (y=mx) gives 1.17 and the minimum absolute error line gives 0.97.

Logarithmic Regression

Fits the equation y=m*log(x)+c using a best (least squares) fit to the input data.

In the same way as linear regression (above), the 2nd csvgraph windows gives the fitted equation and the corresponding R^2 value.

The supplied data file csvfun3.csv includes an example, see graph below.



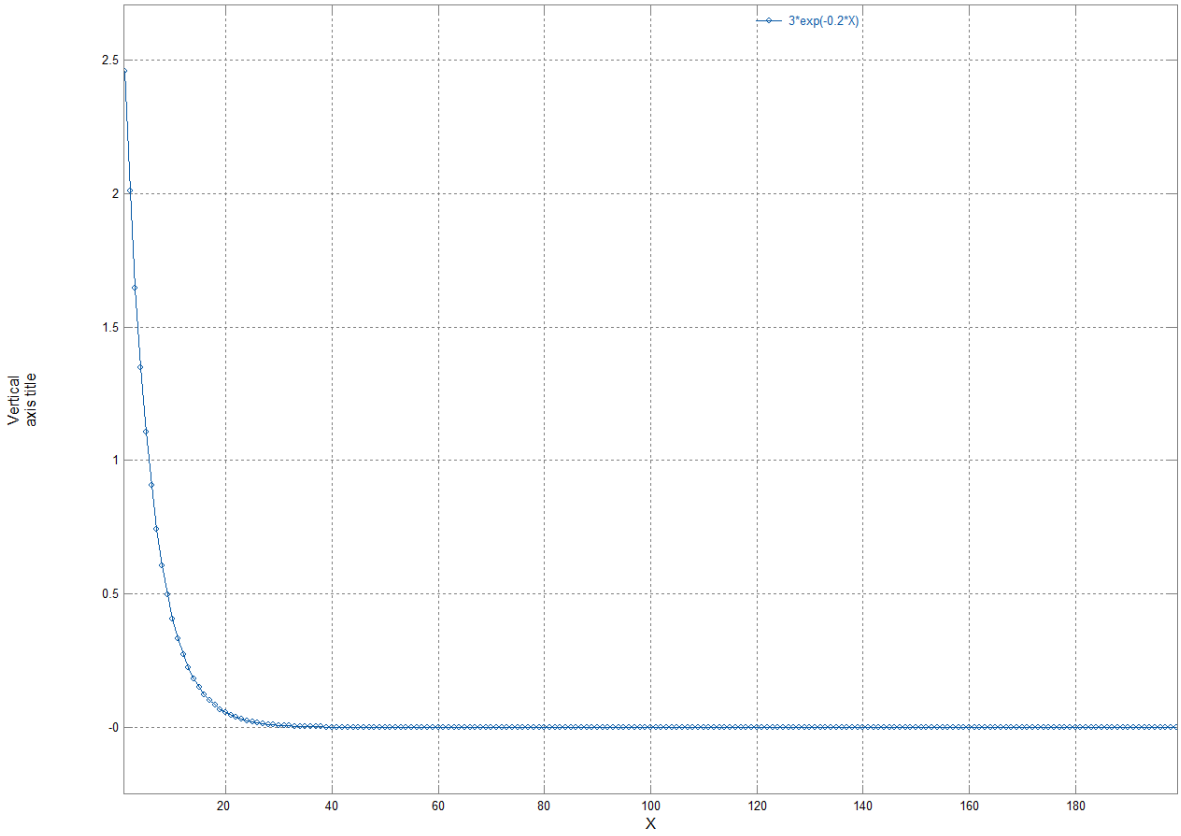
In the 2nd csvgraph window you will see the fit (which in this case is exact):

Best Least squares line is Y=-2*log(X)+3 which has an R^2 of 1.

Exponential Regression

Fits the equation $y=c*e^{m*x}$ using a best (least squares) fit to the input data.

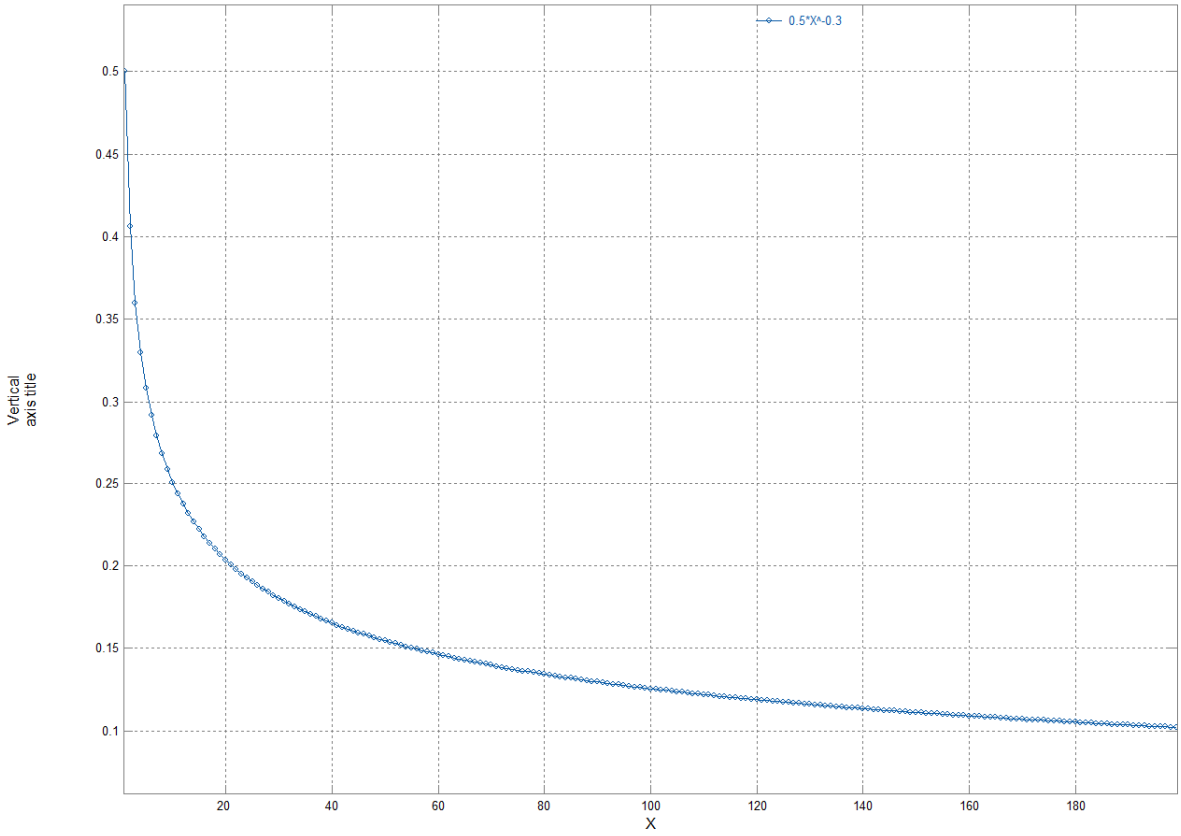
The supplied data file csvfun3.csv includes an example, see graph below.



Power Regression

Fits the equation $y=c*x^m$ using a best (least squares) fit to the input data.

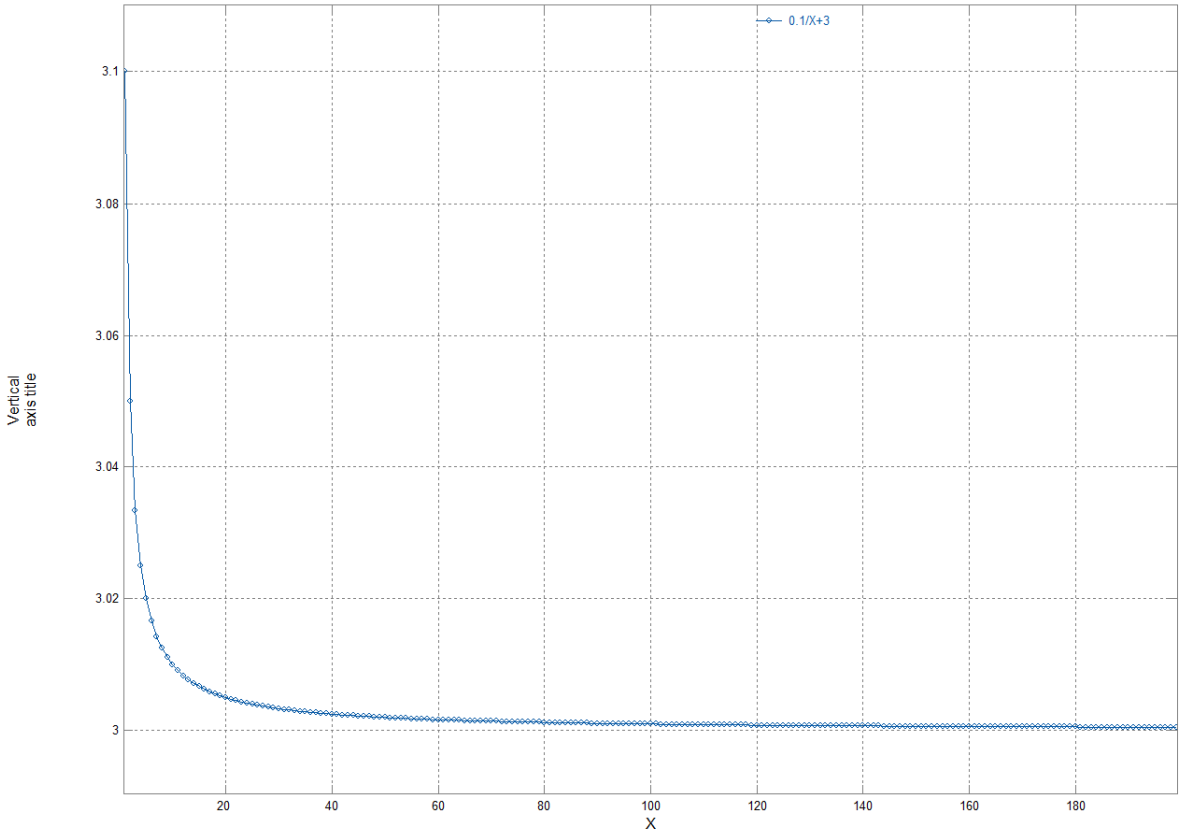
The supplied data file csvfun3.csv includes an example, see graph below.



Reciprocal Regression

Fits the equation y= (m/x)+c using a best (least squares) fit to the input data.

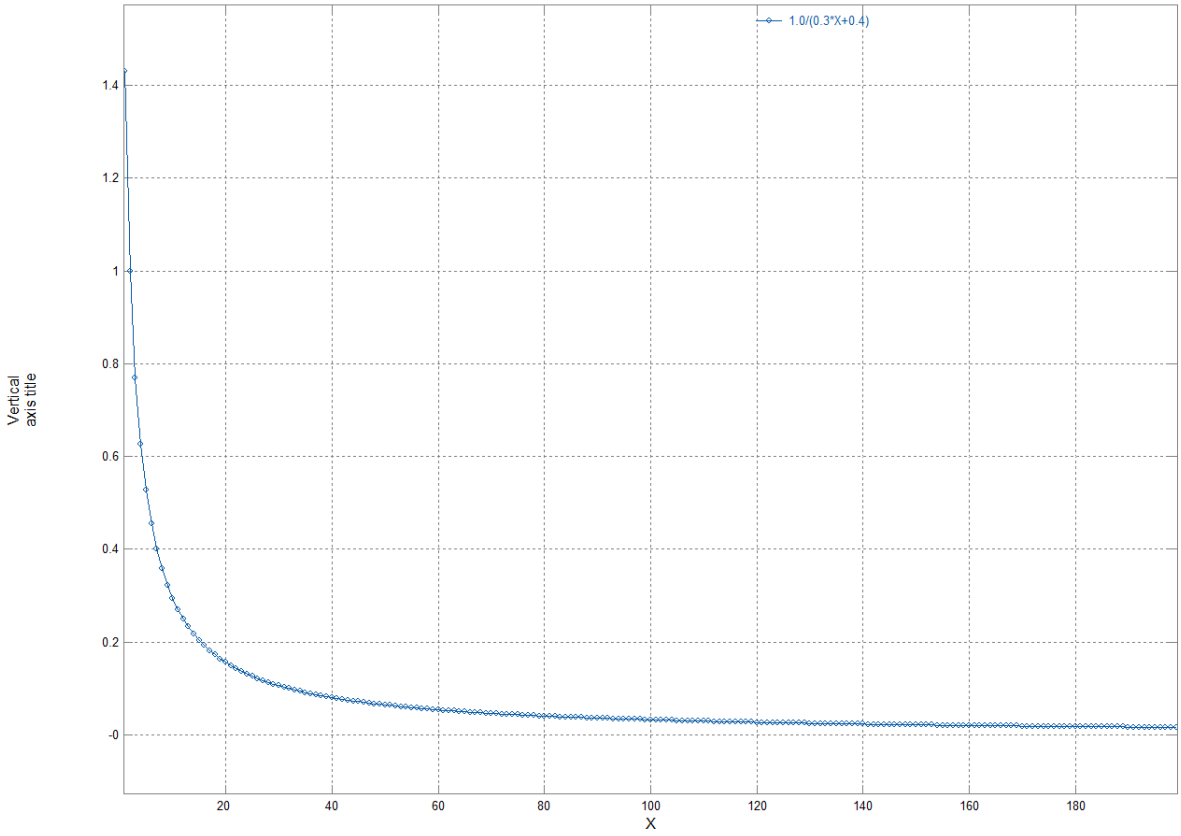
The supplied data file csvfun3.csv includes an example, see graph below.



Inverse Linear Regression

Fits the equation y=1/(m*x+c) using a best (least squares) fit to the input data.

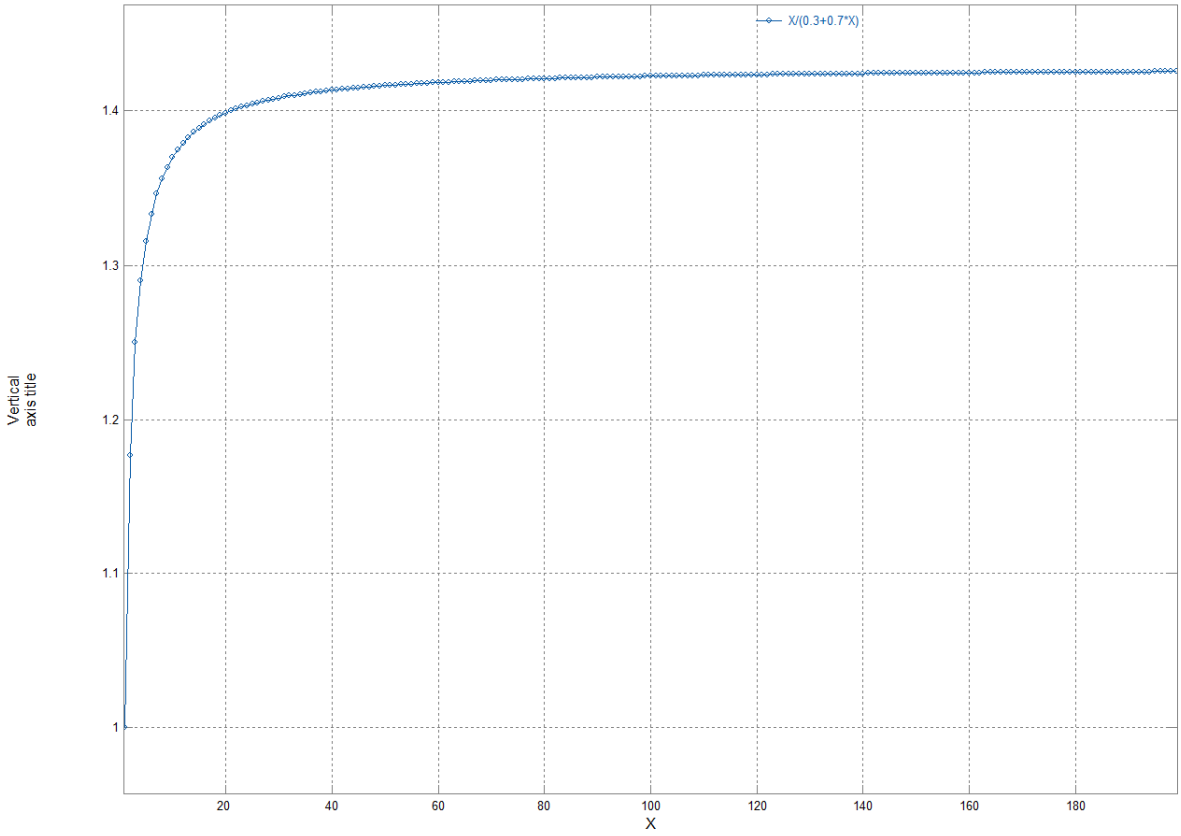
The supplied data file csvfun3.csv includes an example, see graph below.



Hyperbolic Regression

Fits the equation y=x/(m*x+c) using a best (least squares) fit to the input data.

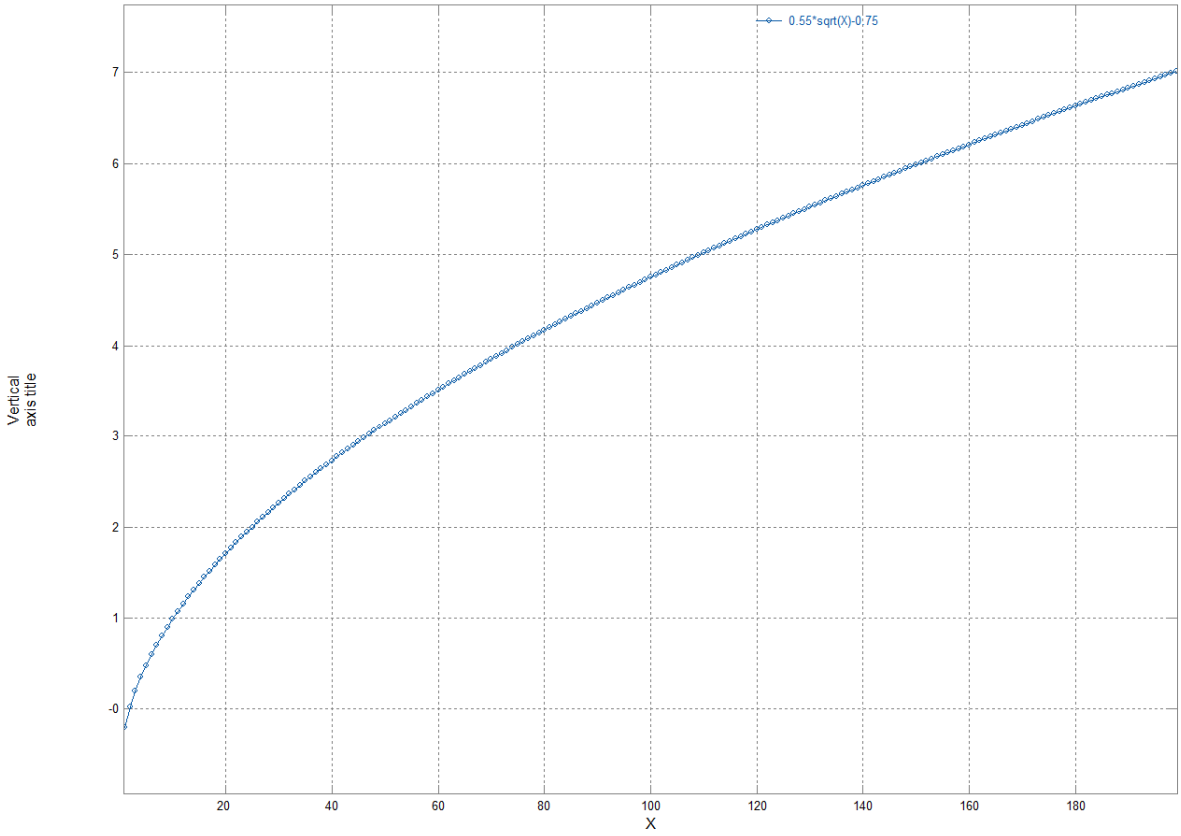
The supplied data file csvfun3.csv includes an example, see graph below.



Square Root Regression

Fits the equation y=m*sqrt(x)+c using a best (least squares) fit to the input data.

The supplied data file csvfun3.csv includes an example, see graph below.



Y=a*x+b*sqrt(X)+c

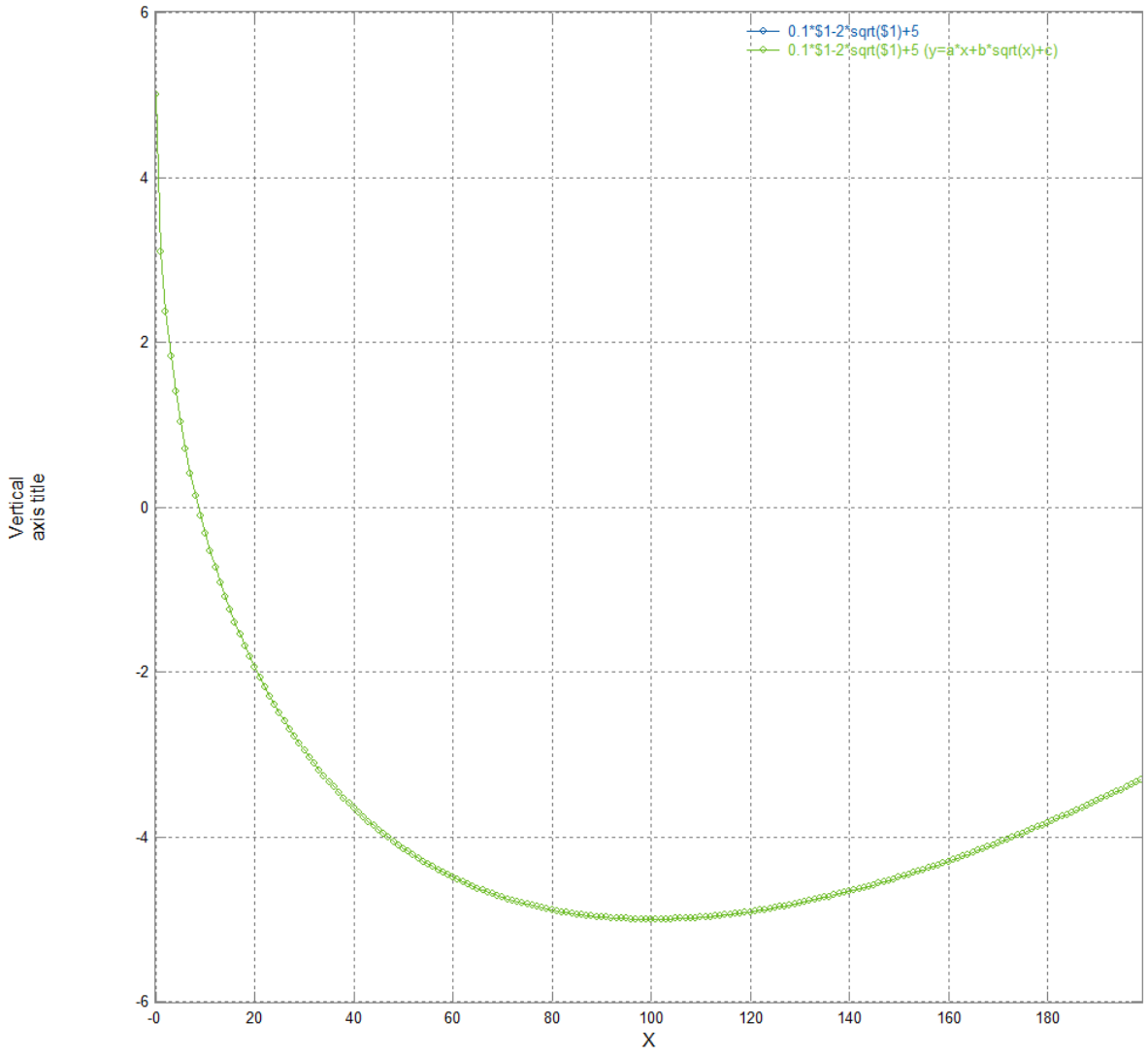
This fits the equation y=a*x+b*sqrt(x)+c using a best (least squares) fit to the input data.

The built-in maths capabilities of csvgraph allows us to create a suitable example as in csvfun3.csv X is the first column ($1).

Looking in the 2nd csv graph window the fitted equation is found:

```
Best fit found is Y=0.1*X-2*sqrt(X)+5
  Max abs error of above curve is 4.76837e-07
```

The resultant graph is shown below:

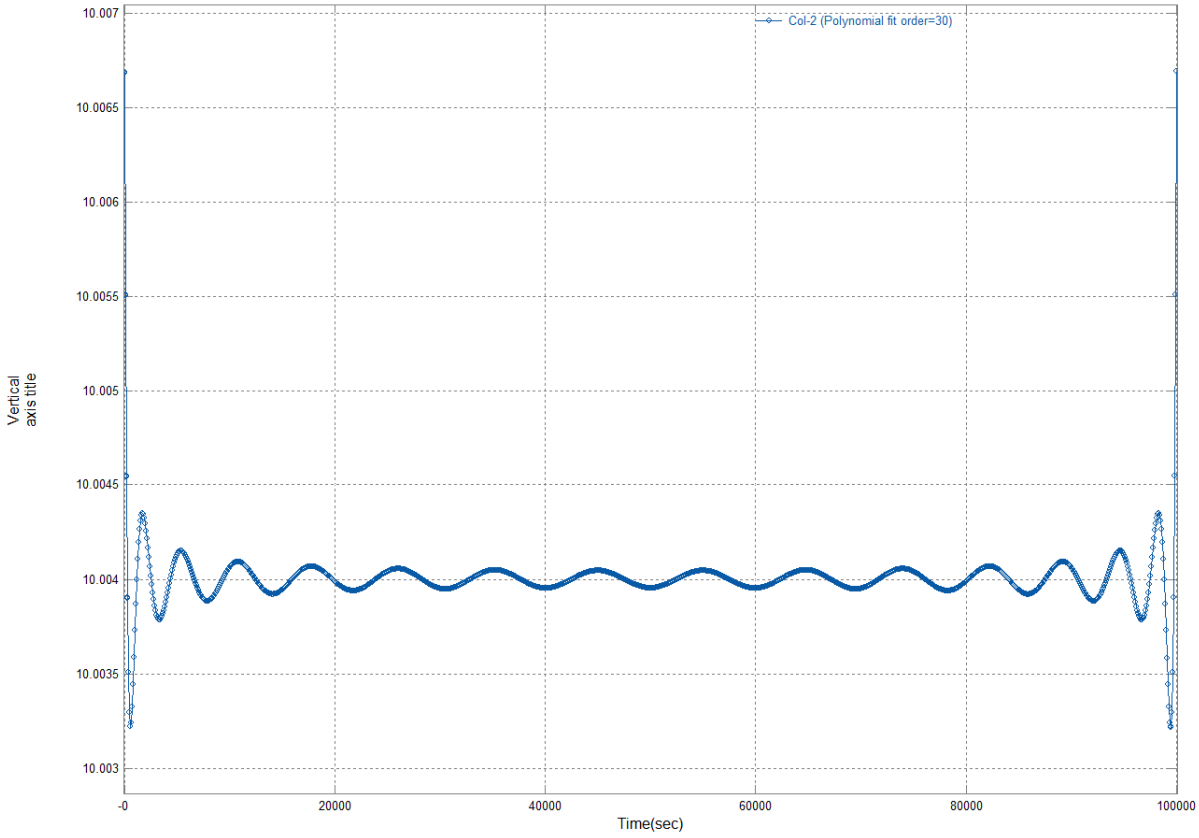


Polynomial fit

This is a more general form than linear regression as the order of the polynomial can be selected.

If the order is 1 this gives identical results to the (least squares) linear regression y=mx+c option, but higher orders can give a better fit.

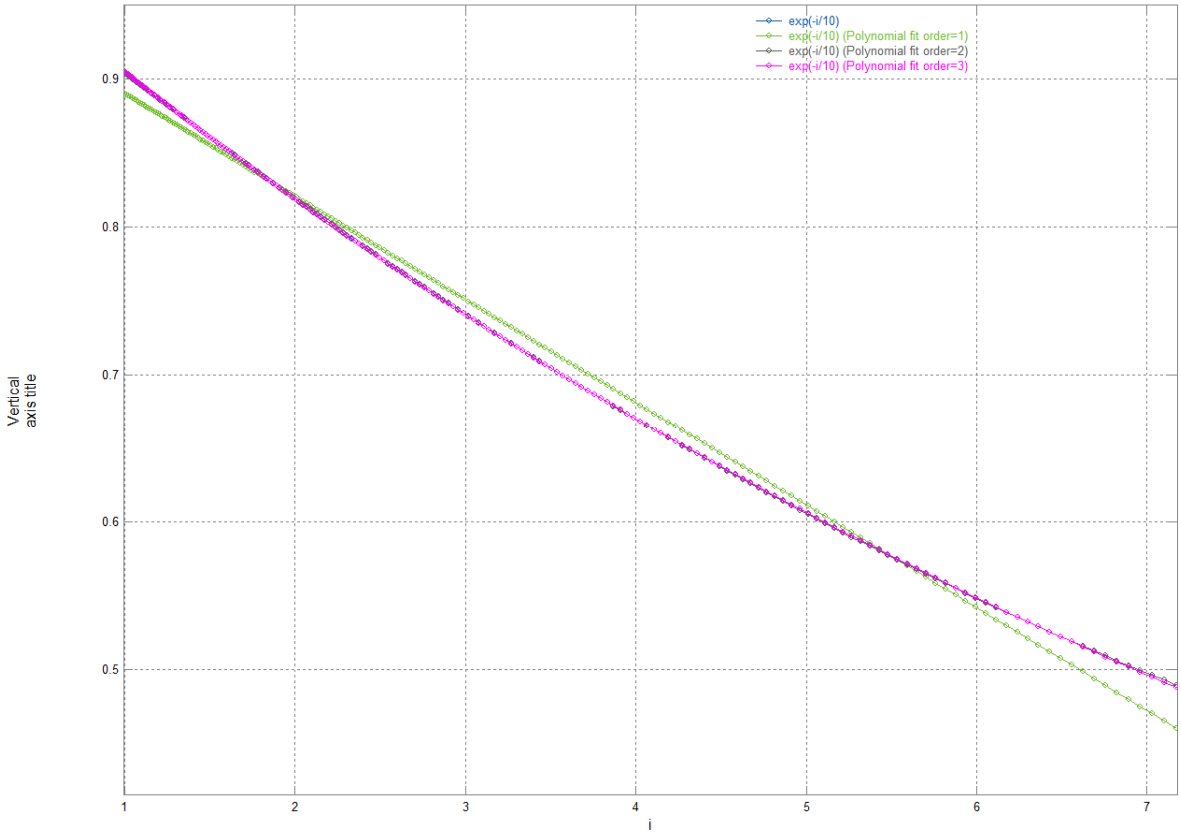
Using demo1M.csv and a 30th order polynomial gives:



Which is the polynomial with the equation (as before this is given in the 2nd csvgraph window):

Y=((((((((((((((((((((((((((((((6.97951274809e-137*X-1.04691504709e-130)*X+7.46518751585e-125)*X-3.36717955513e-119)*X+1.07841837595e-113)*X-2.60977659582e-108)*X+4.95825937558e-103)*X-7.585603359e-98)*X+9.51094773116e-93)*X-9.8953321991e-88)*X+8.61839059489e-83)*X-6.32174846775e-78)*X+3.92052333073e-73)*X-2.05984542296e-68)*X+9.172676001e-64)*X-3.45829252723e-59)*X+1.10109000948e-54)*X-2.94854105574e-50)*X+6.60274774225e-46)*X-1.22708309395e-41)*X+1.87412804271e-37)*X-2.32316065069e-33)*X+2.30046310128e-29)*X-1.78304928219e-25)*X+1.0534361431e-21)*X-4.5791256059e-18)*X+1.39448963045e-14)*X-2.76963858661e-11)*X+3.20180968581e-08)*X-1.7483403854e-05)*X+10.006685239

The file csvfun1.csv gives a more realistic example of polynomial fitting – the screen shot below shows a 1st, 2nd and 3rd order fit to exp(-i/10) – it can be seen that the 3rd order polynomial is a very good fit:



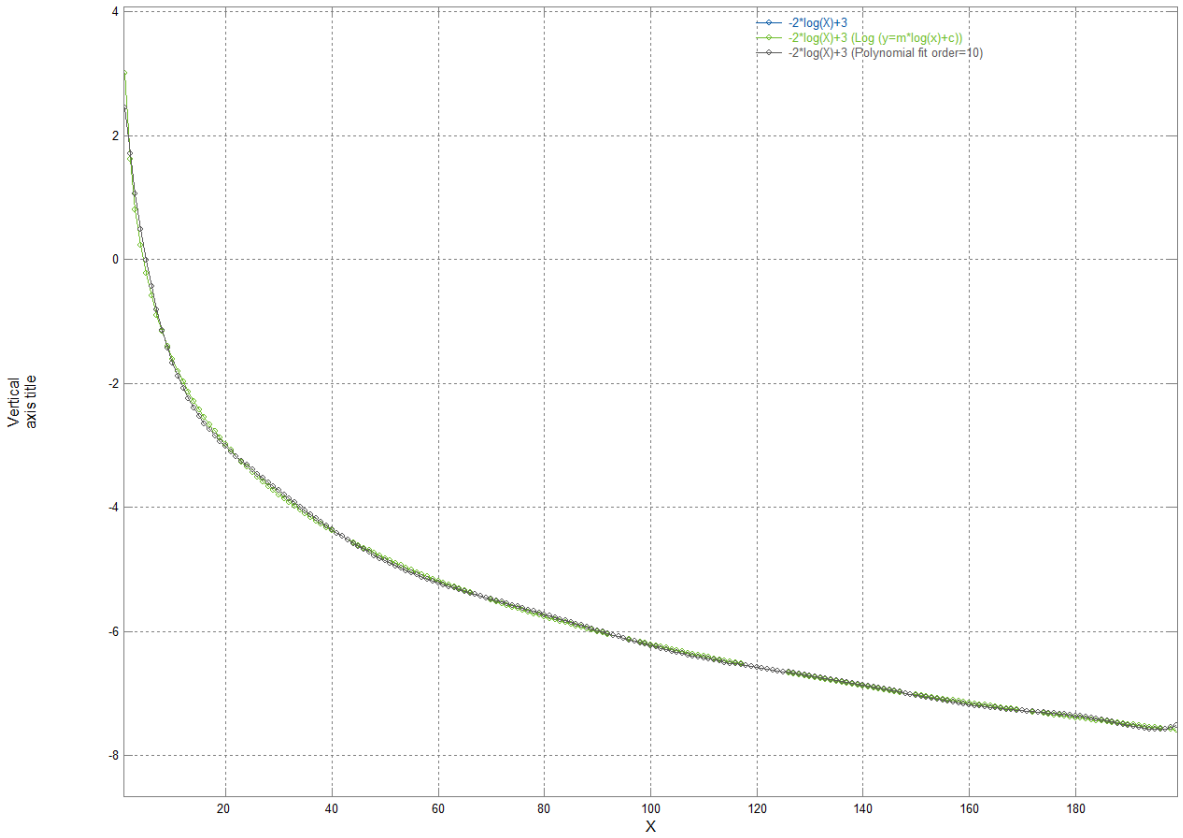
Note that a polynomial fit may require a quite high order (and thus a complex equation) to fit shapes that could be better described by one of the other equations listed earlier.

As an example, see the graph below which shows a Logarithmic regression fits it exactly with a simple equation (with two coefficients) y=m*log(x)+c with m=-2 and c=3, whereas even an 10th order polynomial (with eleven coefficients) is not as accurate (while it captures the overall shape quite well it has an error of 0.56 which is 19% at x=1).

The accuracy of the polynomial fit is given in the 2nd csv graph window together with the equation – in this case this gives:

Polynomial approximating function is:
Y=(((((((((3.95024207162e-19*X-4.19902777376e-16)*X+1.92483796192e-13)*X-4.9806768972e-11)*X+7.99852880753e-09)*X-8.25907300266e-07)*X+5.5007335504e-05)*X-0.00231419319452)*X+0.0590981377101)*X-0.902570371669)*X+3.28864444205
with orthogonal poly : max abs error is 0.557087795729, rms error is 0.0613672600078
with conventional poly: max abs error is 0.557087795729, rms error is 0.0613672600078

This is why it worth trying all the simpler regressions offered by csvgraph before resorting to a high order (greater than 3) polynomial.



FFT

The FFT filters apply a Fast Fourier Transform to the data to create the frequency spectrum.

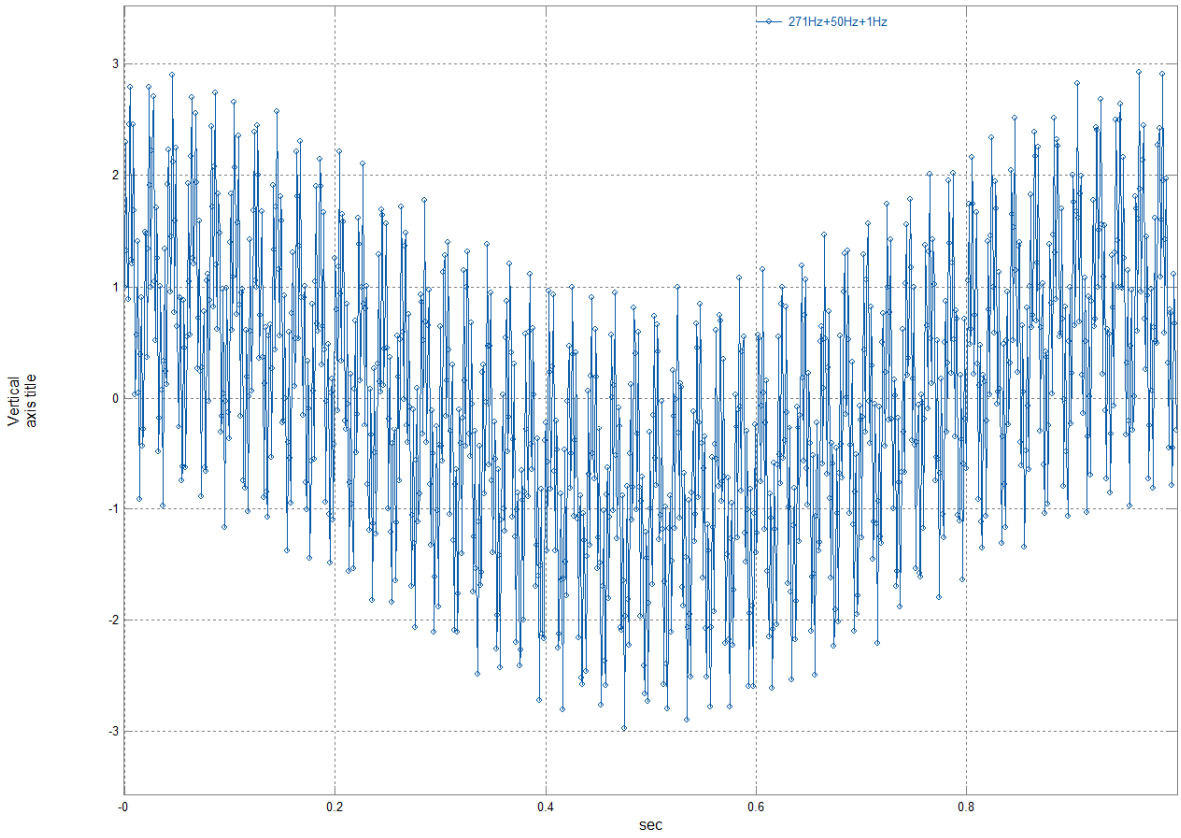
If the original data had an x axis in seconds the new x axis will be scaled in frequency (Hz).

The FFT assumes a constant time step is present in the supplied csv file – csvgraph will warn you if this is not true.

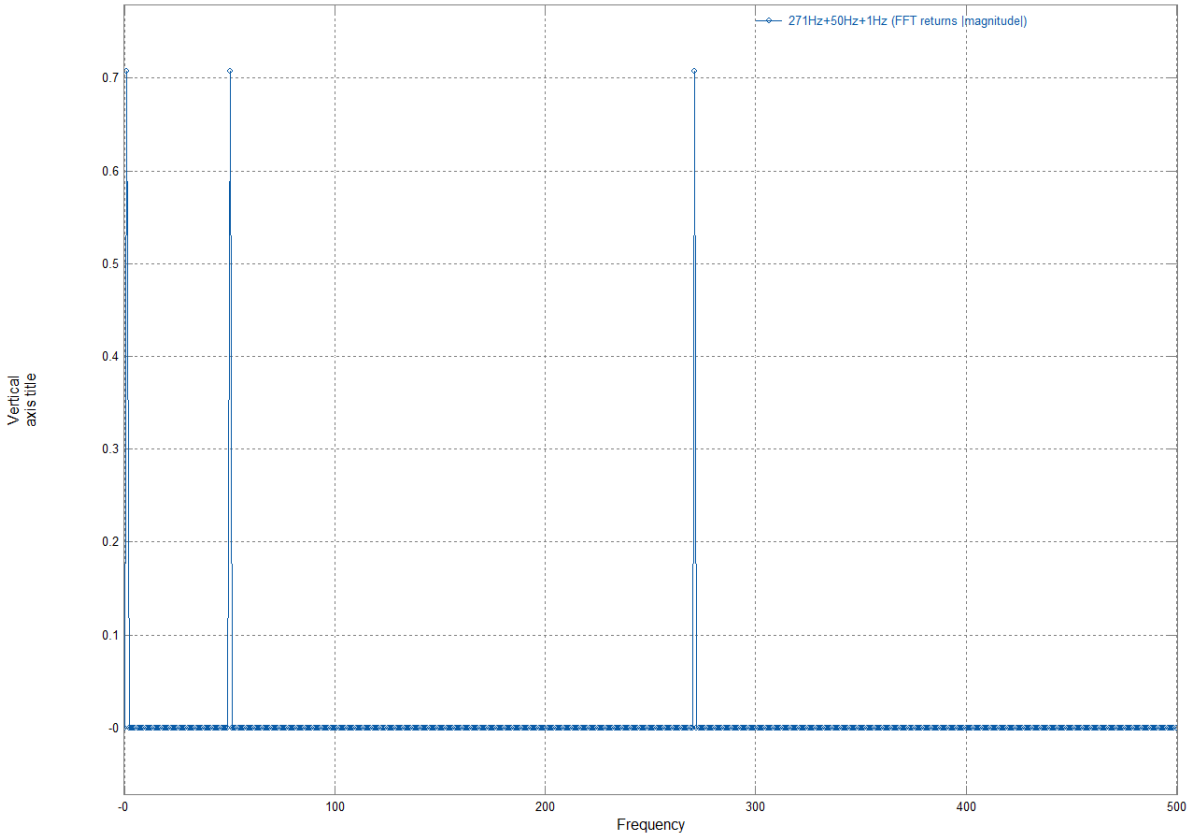
The result can either be viewed as a magnitude or log(magnitude) [in dB], using the log form compresses the dynamic range so that small values are easier to see.

The supplied file csvfun2.csv has a number of waveforms that illustrate the Fourier transform.

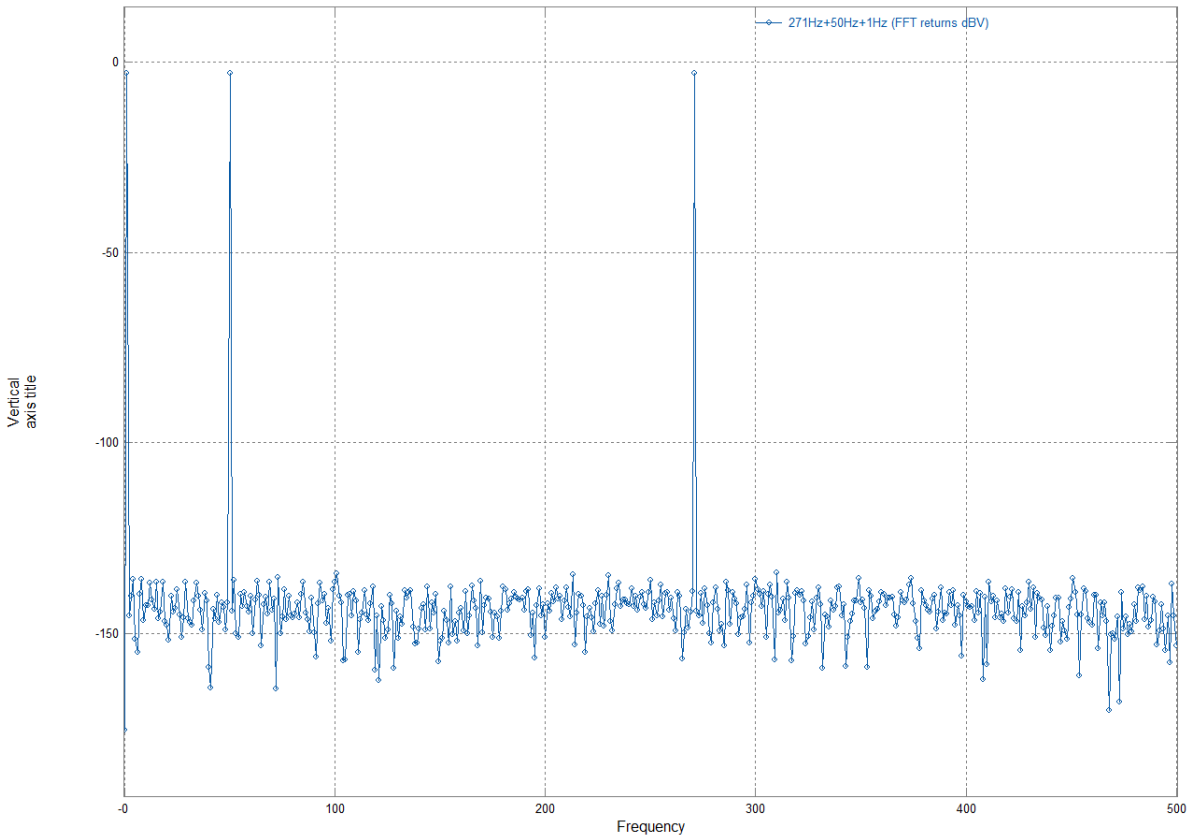
The first graph below is column 7 of this file which shows 271Hz, 50Hz and 1Hz sinewaves – in the time domain (with “None” selected as a filter) this gives:



While in the frequency domain we can clearly see the 3 individual frequencies:

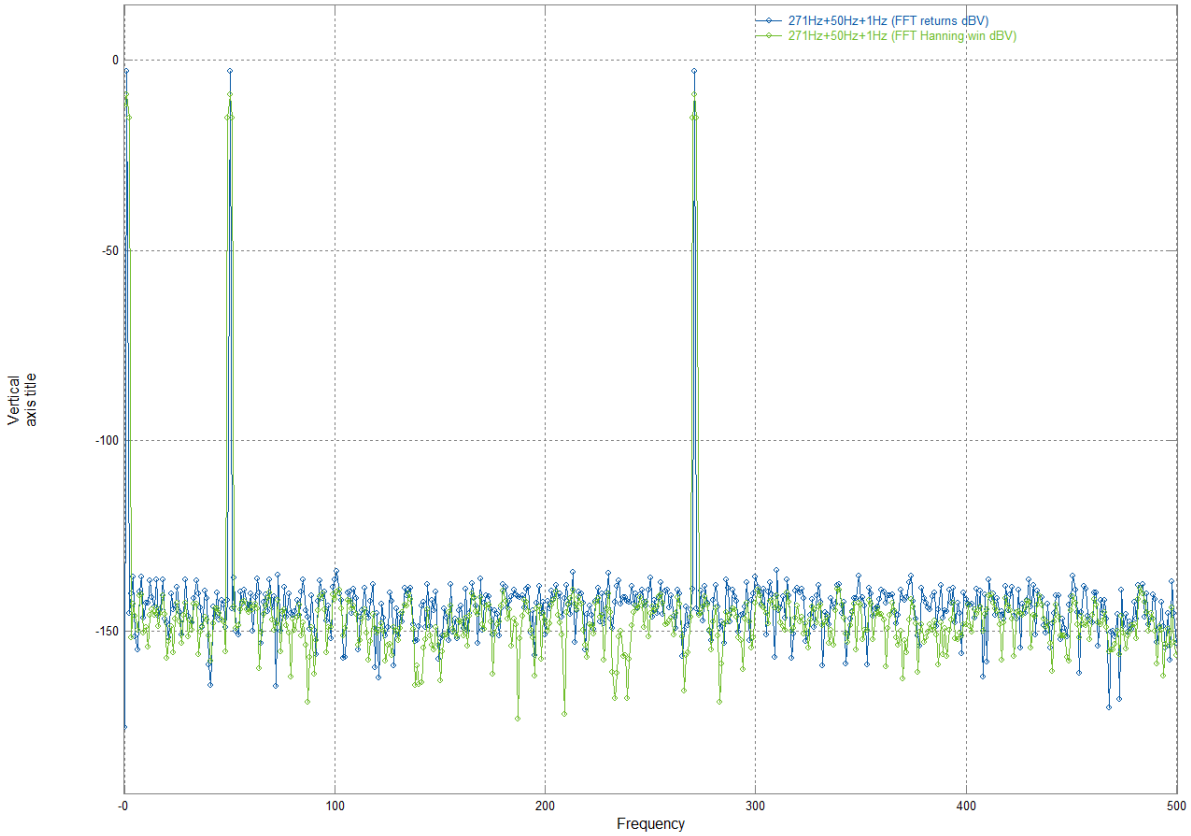

271Hz+50Hz+1Hz (FFT returns |magnitude|)
Vertical axis title
0.7
0.6
0.5
0.4
0.3
0.2
0.1
-0
-0
100
200
300
400
500
Frequency

If we view the log(magnitude) plot:



It can be seen that what appeared to be zero when viewed on a linear scale are actually very small numbers – but again the peaks are clearly visible.

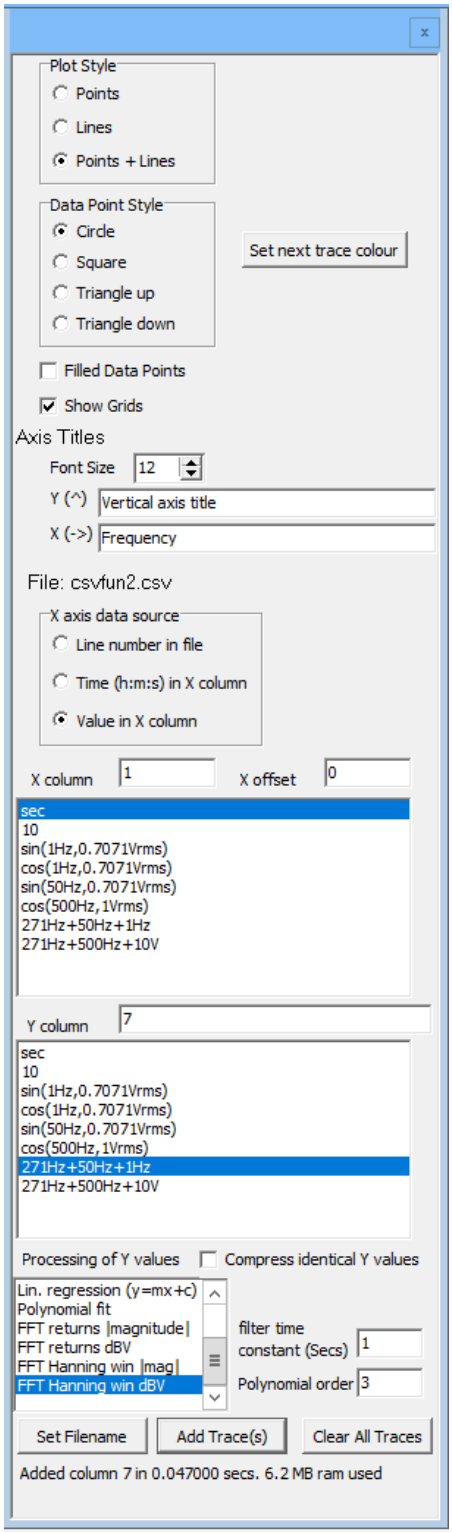
This effect (small non-zero values) can be reduced a little by using a “window” function and csvgraph offers the Hanning (sometimes called the Hann) window function and its effect can be seen below:



For more information on FFT’s and windowing functions please search the web or read one of the many books on this subject.

Undocking the control panel

If the right-hand control panel is clicked in an area where there is no control then it will “undock” from the graph window to give a 3rd csvgraph window:



The graph will expand to take up the whole of the original window so this can be useful for use with smaller displays. Clicking the “x” at the top right of this control window will make it disappear, it can be recovered by using the main menu, Panel, visible.

Help

Csvgraph is designed to be simple to use and most active areas of the screen have a tooltip that appears when you hover the mouse over them. From the main menu Help/About gives a brief summary of the available functions, while Help/Manual shows (this) manual assuming its available (see installation).

Appendix A – expressions

Csvgraph allows expressions that are basically a subset of those available in the language AWK (which for expressions is very similar to C), and AWK also provides the syntax to select a column ($n to select column n, e.g., $5 to select column 5).

The operators available in decreasing priority are:

+/-constants, +/-(), +/-$n

~,!

*,/,%

+,-

>>,<<

<,>,>=,<=

==,!=

&

^

|

&&

||

Conditional expressions can be written using ? and : for example ($2==5)?3:4 which gives 3 if the value in column 2 is 5 and 4 if it's not [multiple ?: pairs can be used in an expression if necessary.

One constant (pi) is available and the following functions:

abs()

acos()

asin()

atan(),

cos(),

cosh()

exp()

log()

max(,)

min(,)

pow(,)

sin()

sinh()

sqrt()

tan()

tanh()

Functions take one argument except when shown "(,)" when they take two.

All the trig functions work in radians.

log () is log base e.

Numbers can be integers, normal floating-point numbers (e.g., 0.1 or 1e-20) or hex numbers (0xnnnn) – the operators &,^,| (and, xor, or) work with 32 unsigned integers.

For efficiency expressions are "compiled" (and optimised) before a csv file is loaded.

For more complex processing on csv files the author recommends the use of AWK, see for example https://github.com/p-j-miller/wmawk2 .

Installation

Cvsgraph is a portable program which does not need installation.

Copy the file csvgraph.exe to any location on your computer (or run it from a USB-stick or similar).

If you wish the Help/Manual function to work then copy csvgraph.pdf to the same directory (location) as csvgraph.exe.

A shortcut on your desktop makes it simple to execute csvgraph.

The first time you run csvgraph you may see a Windows warning "The Publisher could not be verified. Are you sure you want to run this software" (or similar), you can either run anyway (the executable from github should be safe) or compile your own executable from the source files (a free version of the required Builder C++ compiler is available at www.embarcadero.com/products/cbuilder/starter).

You may also get a similar message from your pdf reader the first time you use Menu/Help/Manual function, again you can accept this as you know why the pdf reader was invoked.

See the file LICENSE for details but csvgraph is free for both commercial and non-commercial use.

Changes

1v0 - 3/1/2021 - 1st release on Github

1v1 – 6/1/2021 – Improvements to (this) manual.

- Bug fix to potentially incorrect DC component of FFT

- ability to access (this) manual with csvgraph using menu/Help/Manual

1v2 – 24/1/2021 – bug fix “inf” in csv file would be read as an extremely large number (infinity) which then caused issues when csvgraph tried to scale numbers and draw the graph

- Added many more options for “filtering” including exponential, power, hyperbolic and sqrt.

1v3 – 3/2/2021 - more curve fitting options added, y=mx, y=mx+c with GMR , minimum absolute error and minimum relative error, and y=a*x+b*sqrt(x)+c.

2v0 – 17/2/2021 – Major internal changes to reduce RAM usage and improve speed.

No changes to function.