

Csvgraph – a graph plotter for csv format files

Manual for Version 1.0 2/1/2021

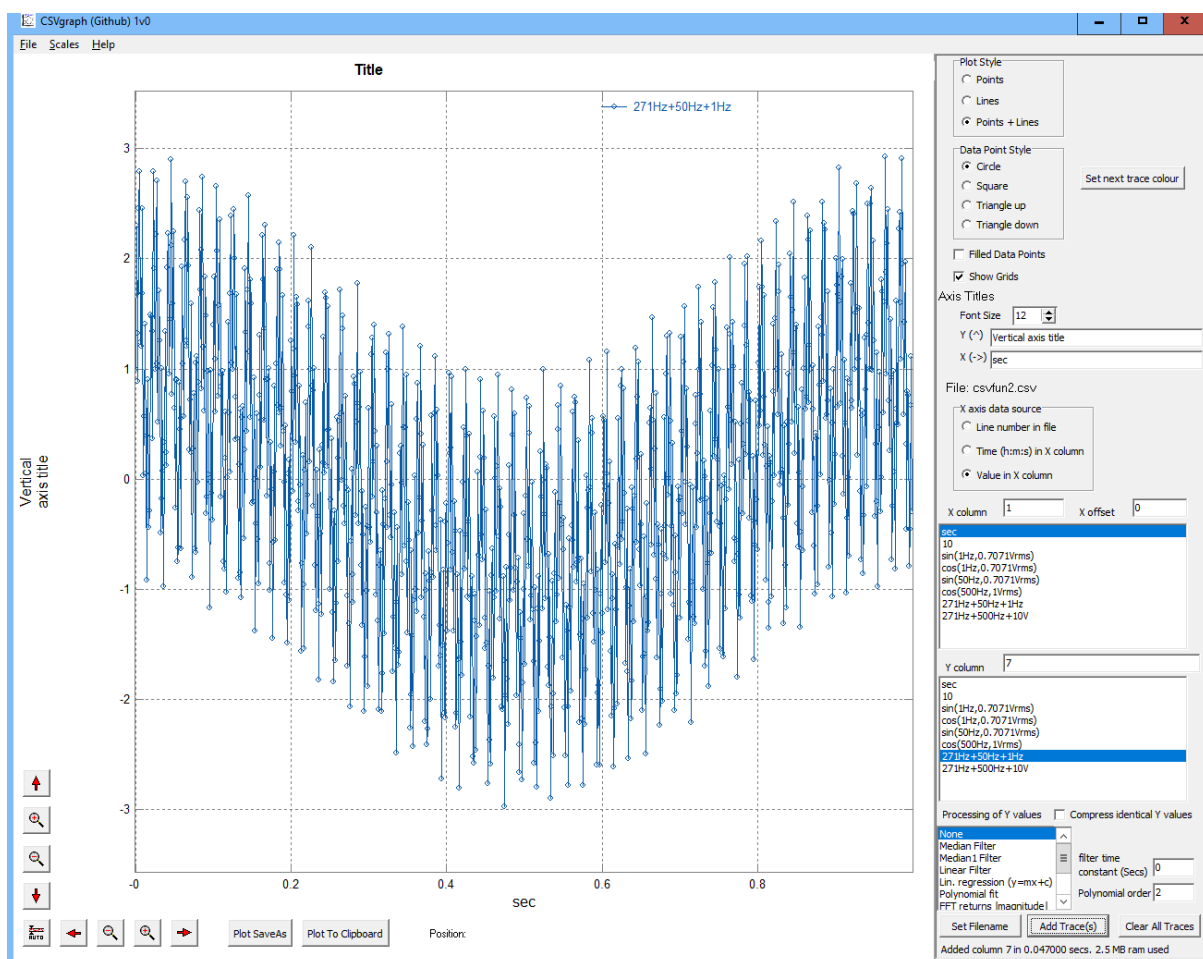
Introduction

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:

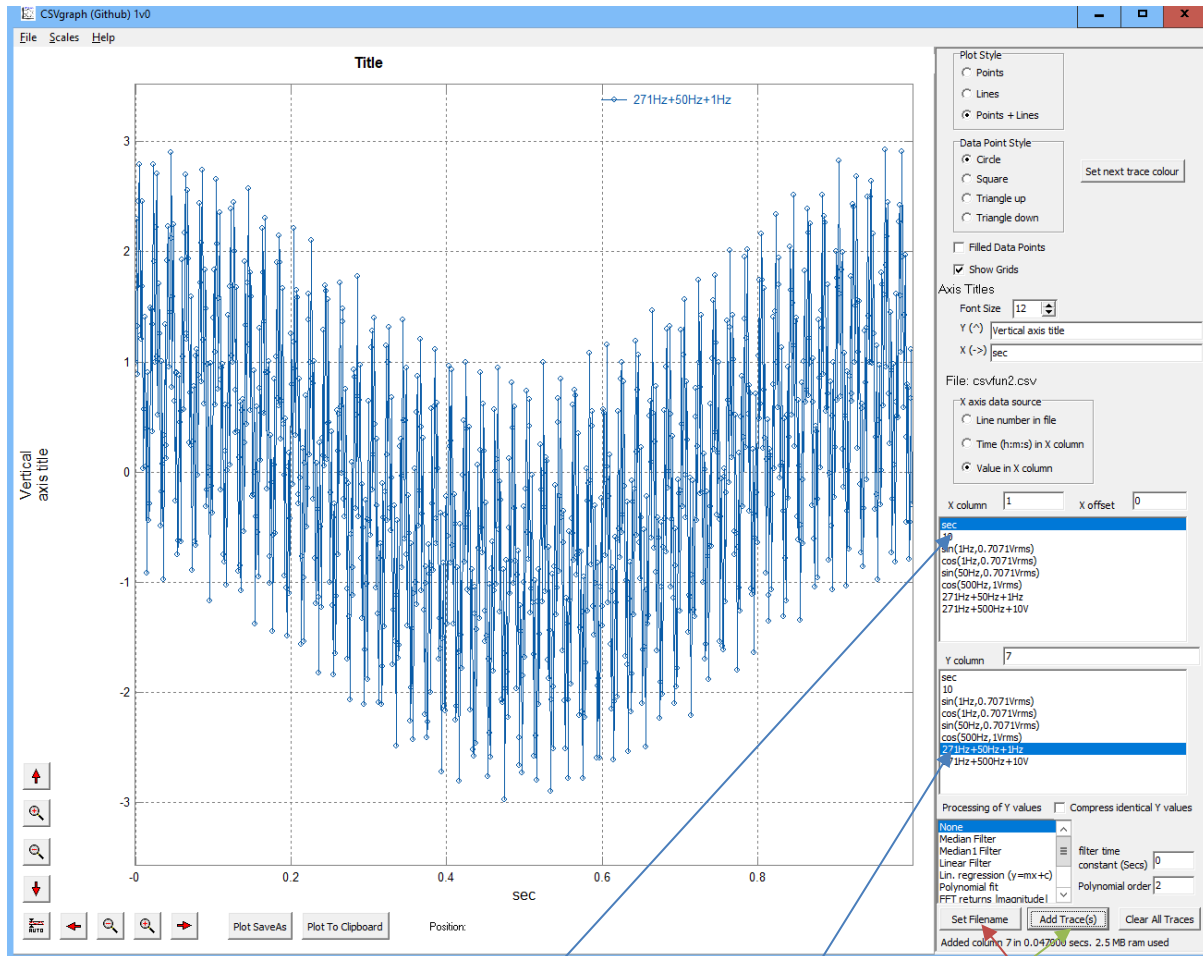
```
"Time(sec)", "Col-2", "Col-3", "Col-4", "Col-5"
99950,20,0,20,20
99950.1,10,1,11,12
```

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



Use

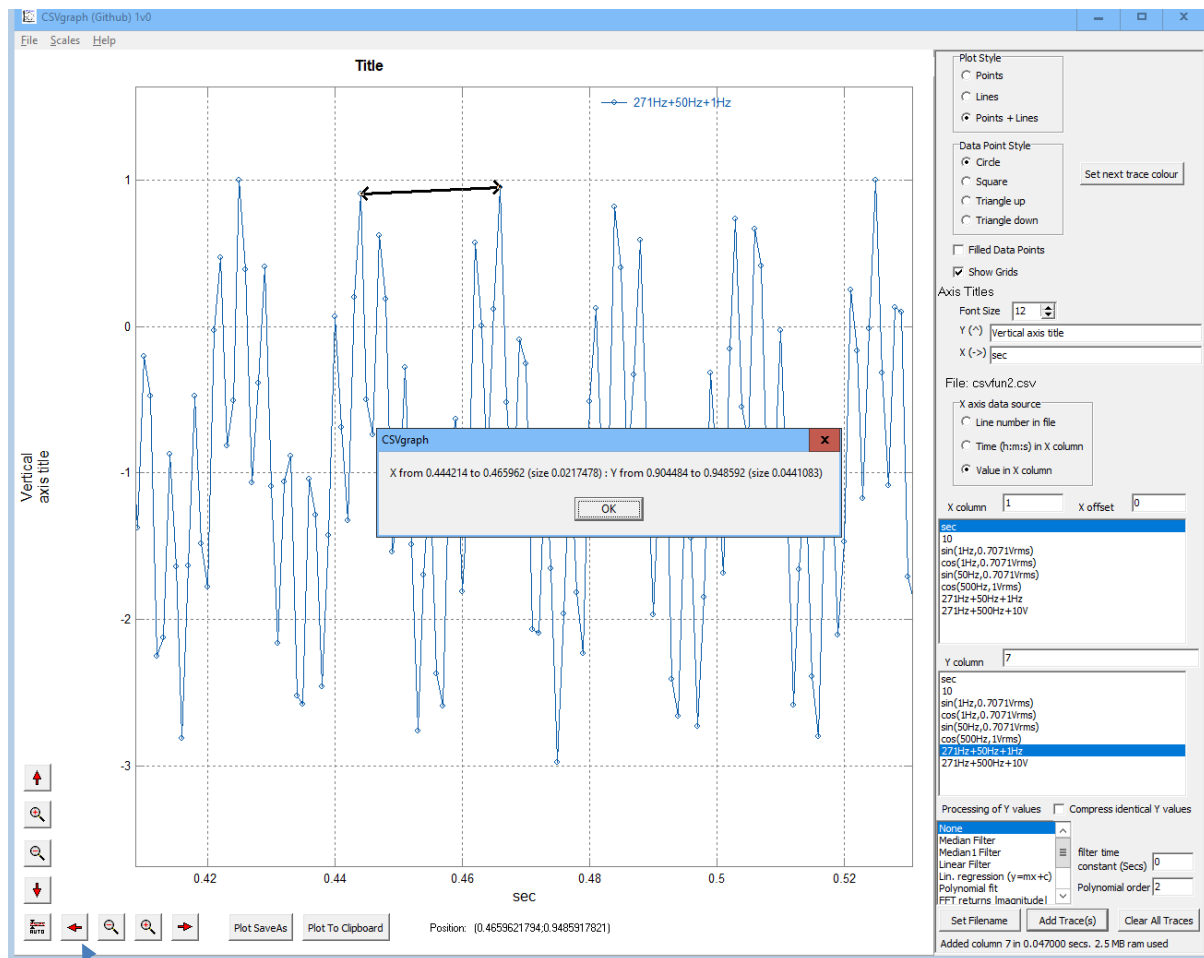
Run csvgraph.exe by double clicking on it.



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

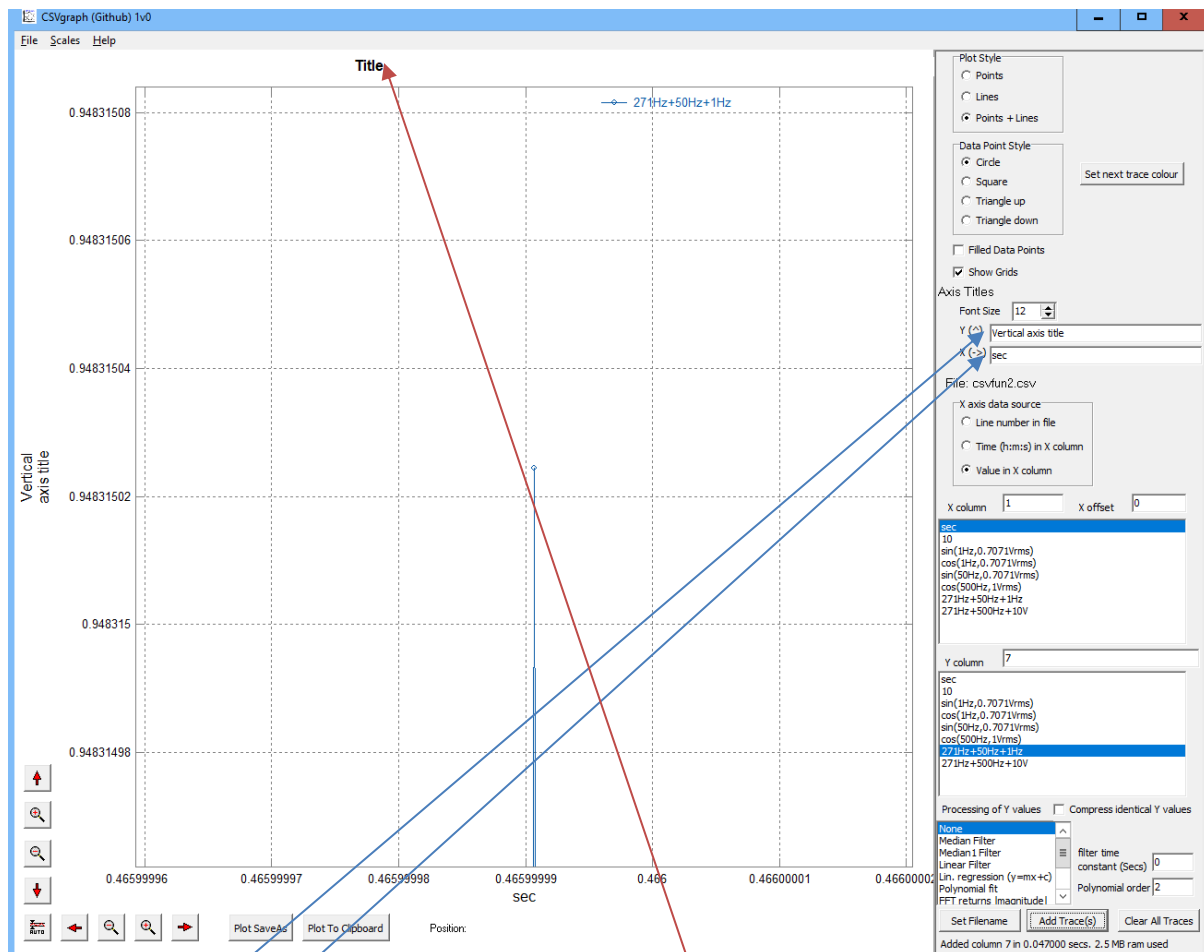
When a graph is displayed move the mouse over it and press the left mouse button, while keeping this button pressed move the mouse to select an area of the graph – when the mouse button is released the graph will zoom into the selected area. To restore the original view, press the middle mouse button. 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:



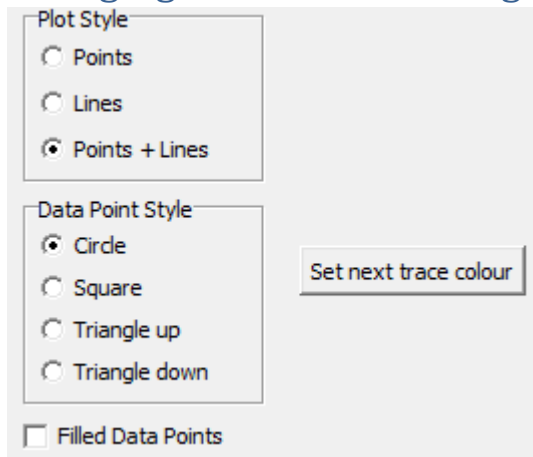
Adding Legends to a graph

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.

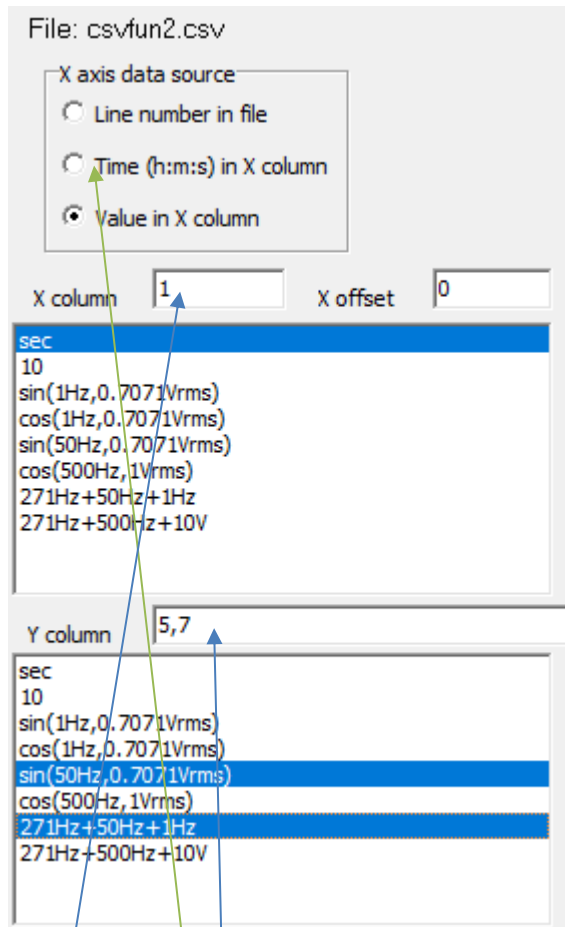
Saving graphs

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.

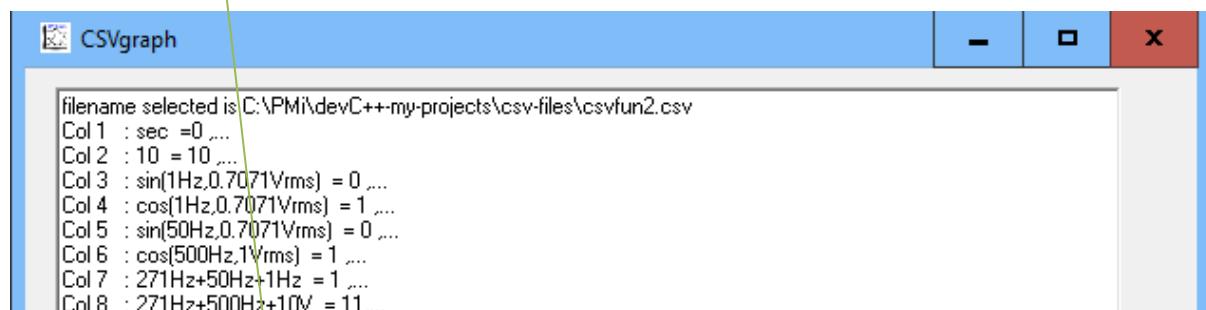
The data within a graph can be saved (which is useful if this has been calculated or filtered – see later for how to do this) by using the menu File/Save/Save Data as csv.

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 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).

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

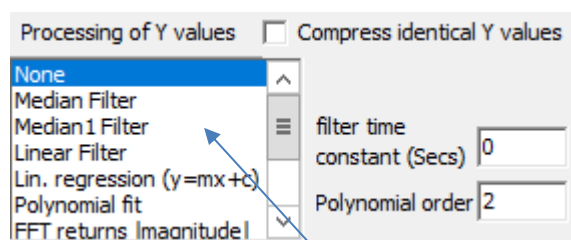
The y column value may also be described as an expression potentially combining the values from multiple columns e.g. \$2-\$3 would plot the difference between the 2nd and 3rd columns.

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.



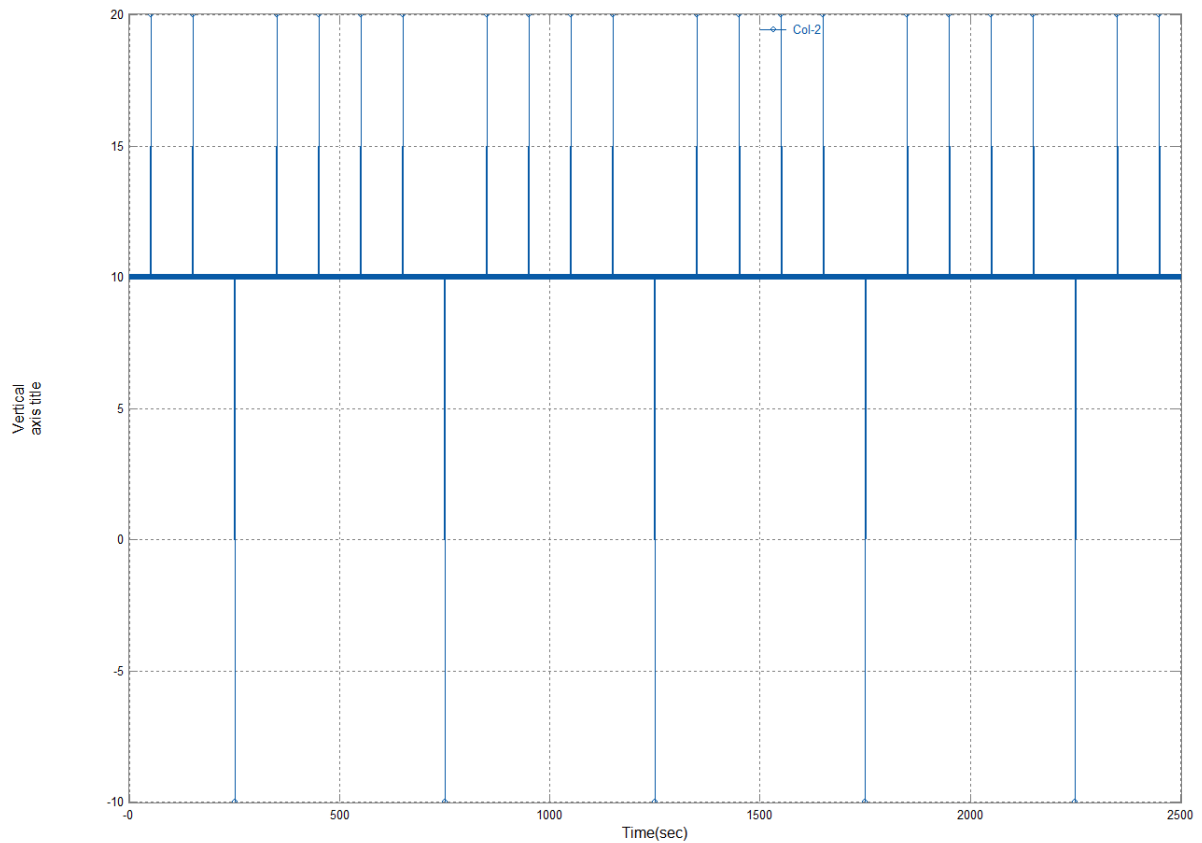
The simplest option which is unlikely to be needed unless you need to view extremely large files or have limited RAM is “Compress identical Y values”. If this is ticked then sequences with identical Y values will be compressed (just the first and last point in a sequence kept) – the line graph will be 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.

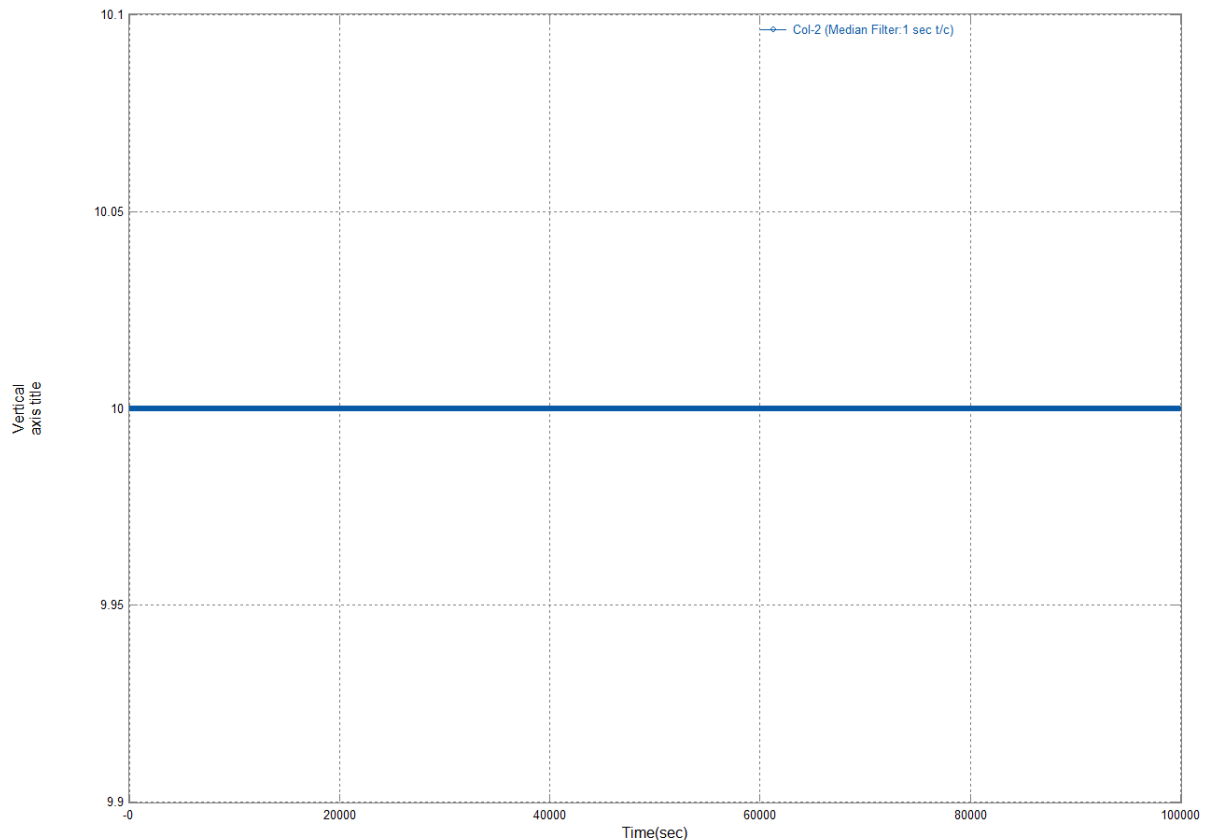
Looking at the example file demo1M.csv scaled with x from 0 to 2500 and Y from -10 to +20 (using menu/Scales to set these limits) gives:



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.

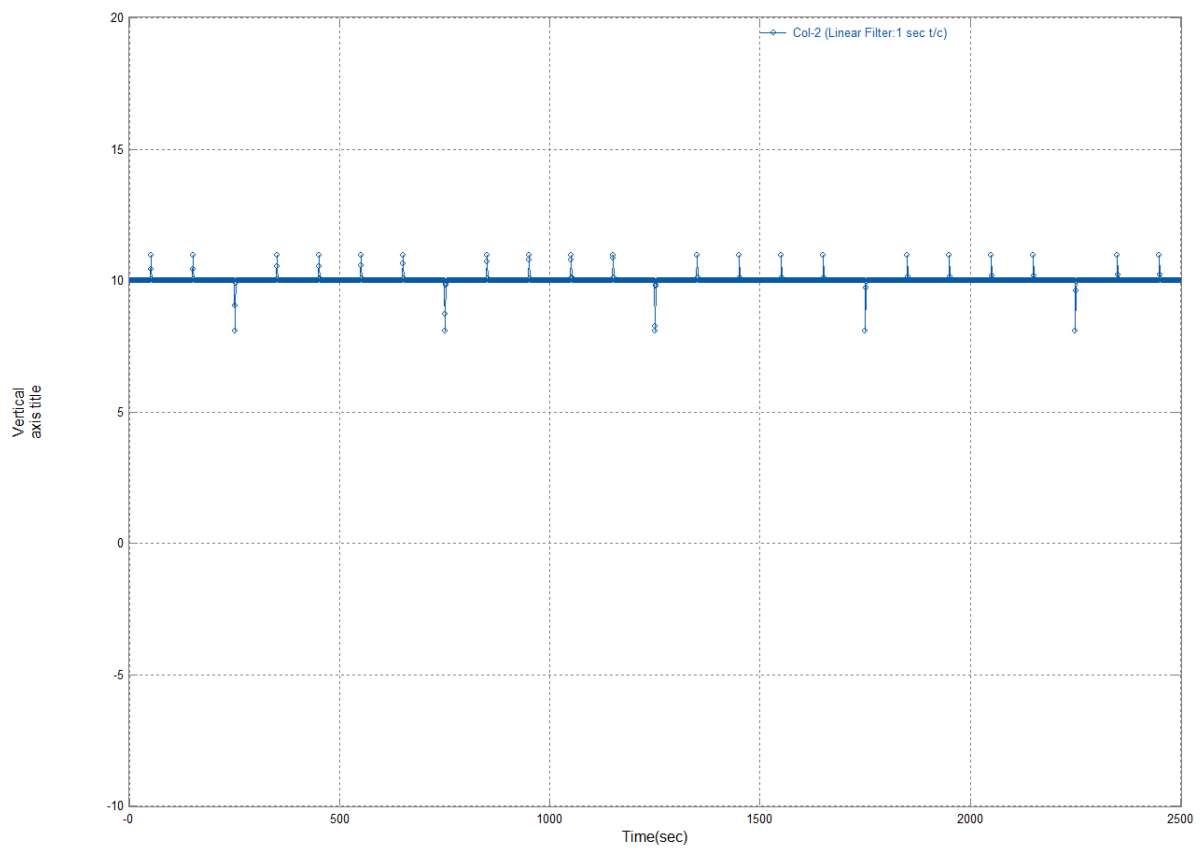
The difference between the Median and Median 1 filters is more subtle (they both give the same result in this example), the Median 1 filter gives a smoother change when the underlying data changes. Note there is no “lag” with a median filter (compare this to the linear filter below).

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.

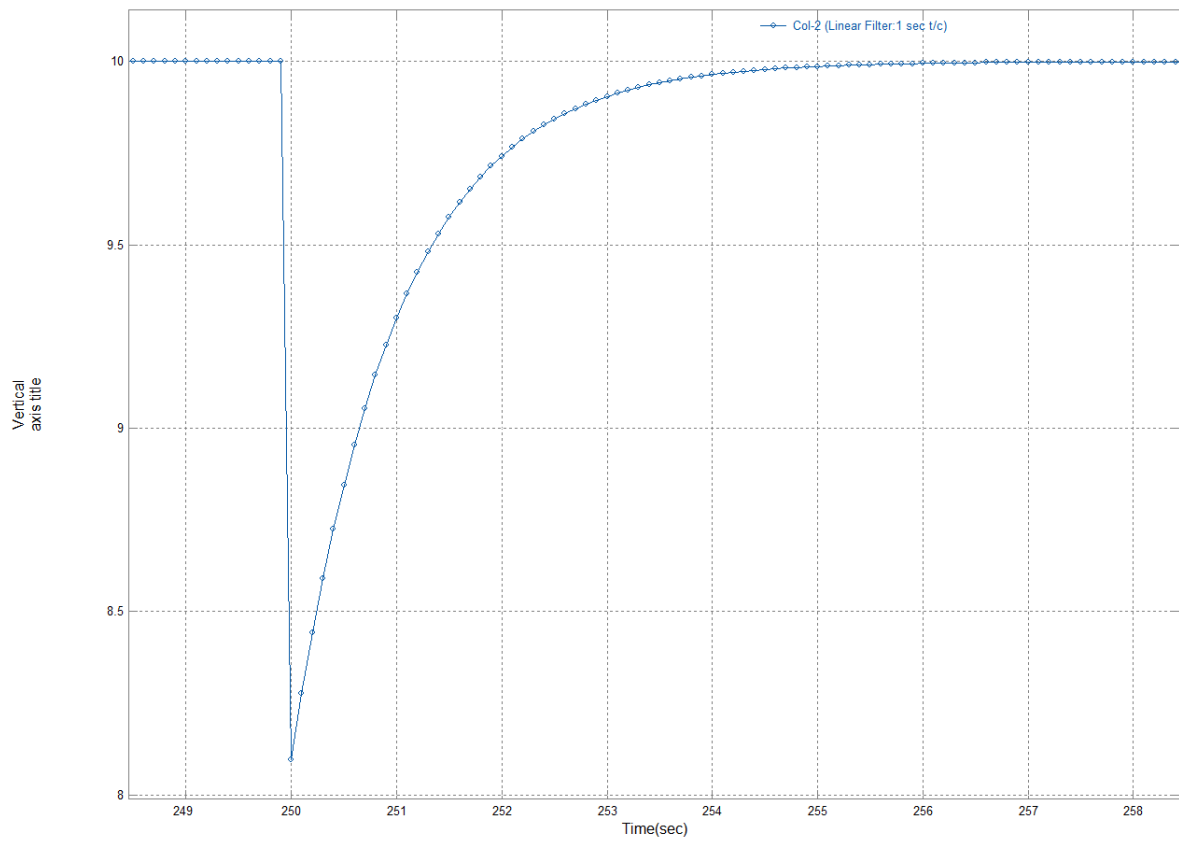
Linear Filter

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:



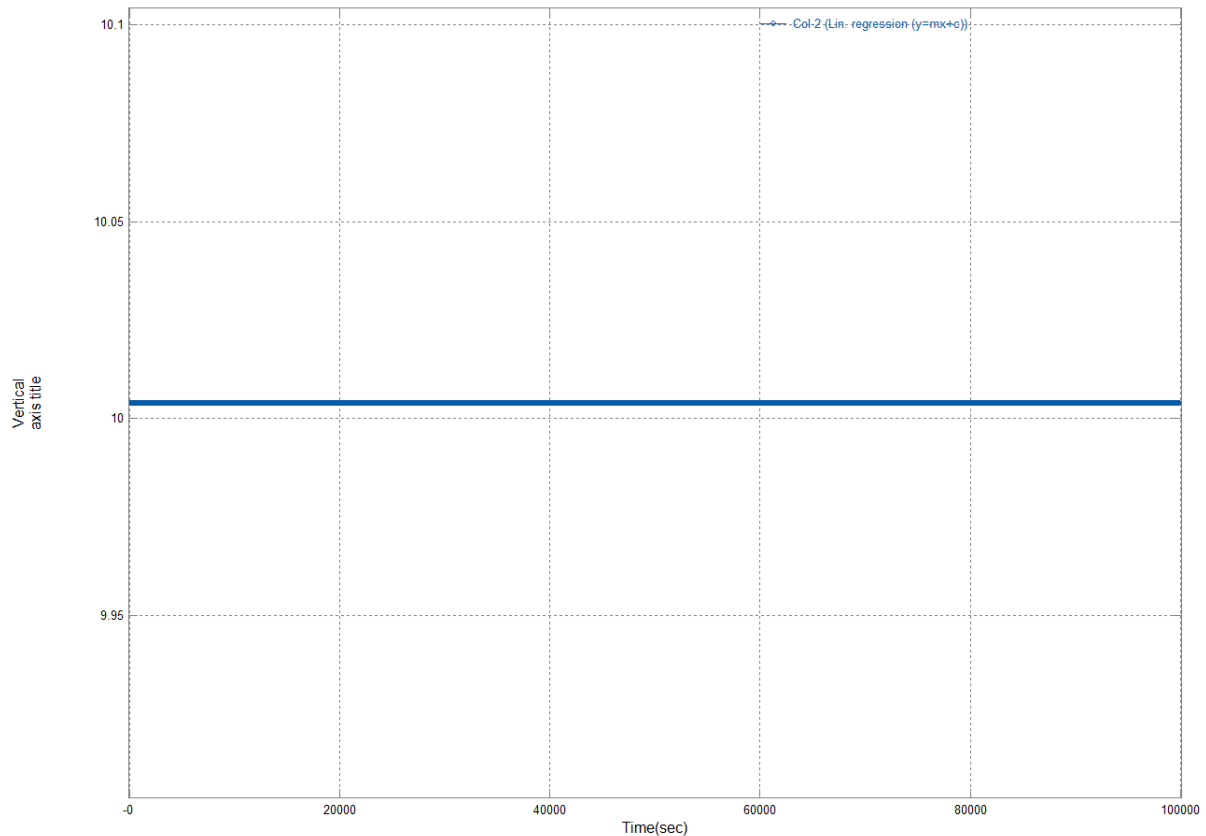
Zooming into one of the spikes shows that while this filter has reduced the height of the spikes it has also extended their duration:



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 ($Y=2.38342e-13X+10.004$) 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).

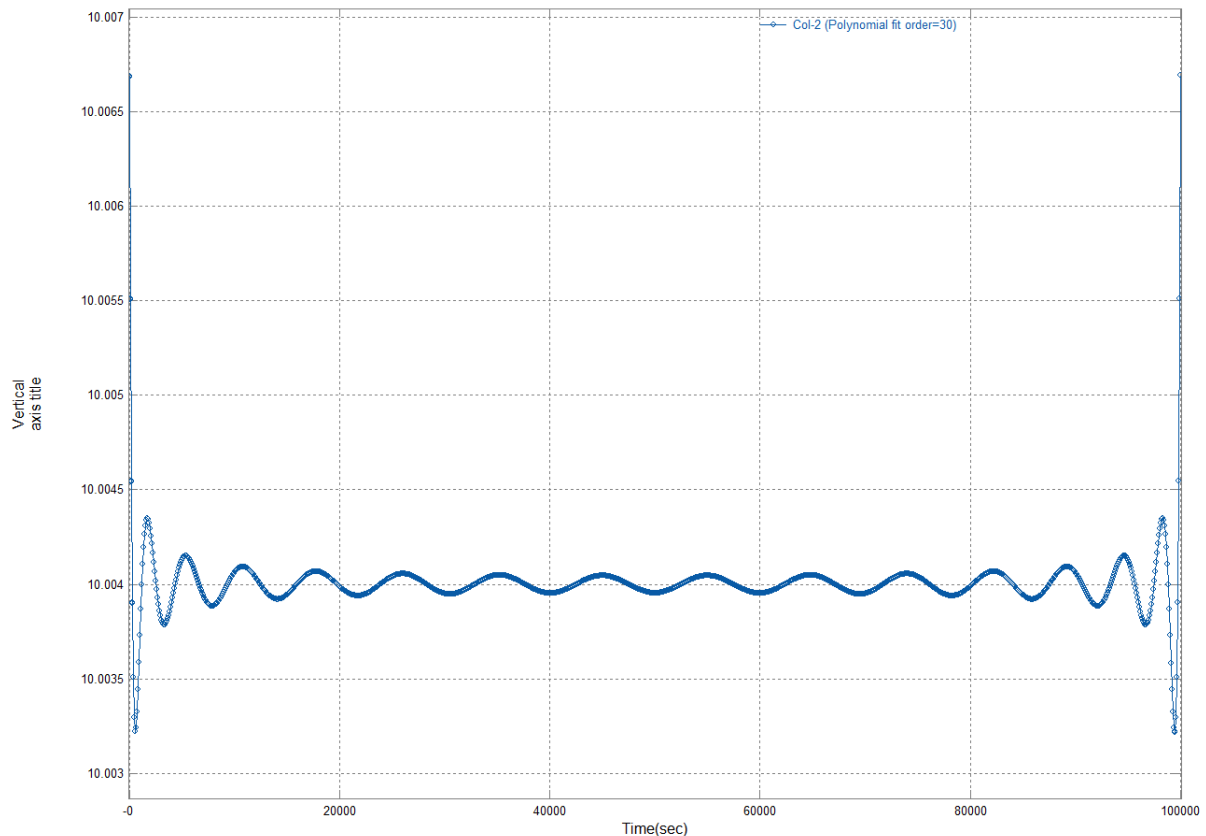
In this case it has removed the spikes in a similar way as the median filter, but the resultant y value is a little different (here it is the average Y value (10.004) rather than the median (10.000)).

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 linear regression option, but higher orders can give a better fit.

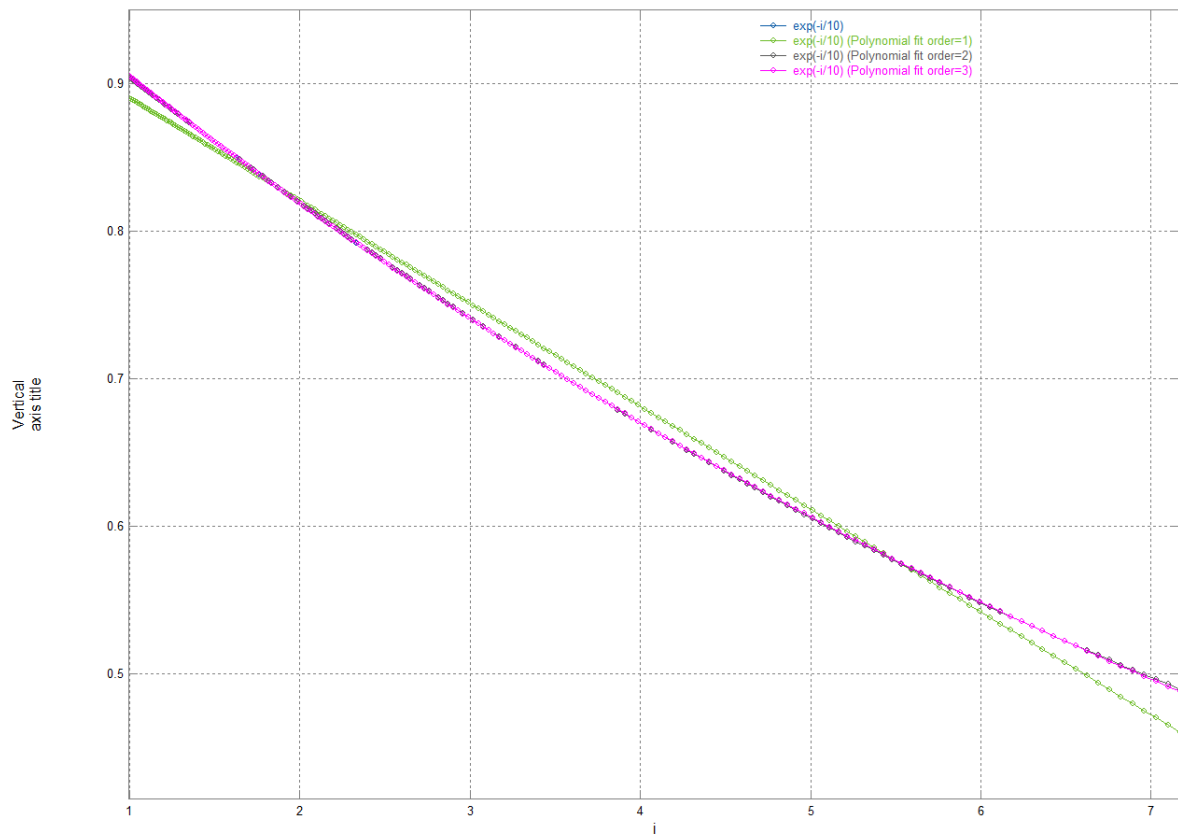
Using the same example as above and a 30th order polynomial gives:



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+10.00685239
```

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:



FFT

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

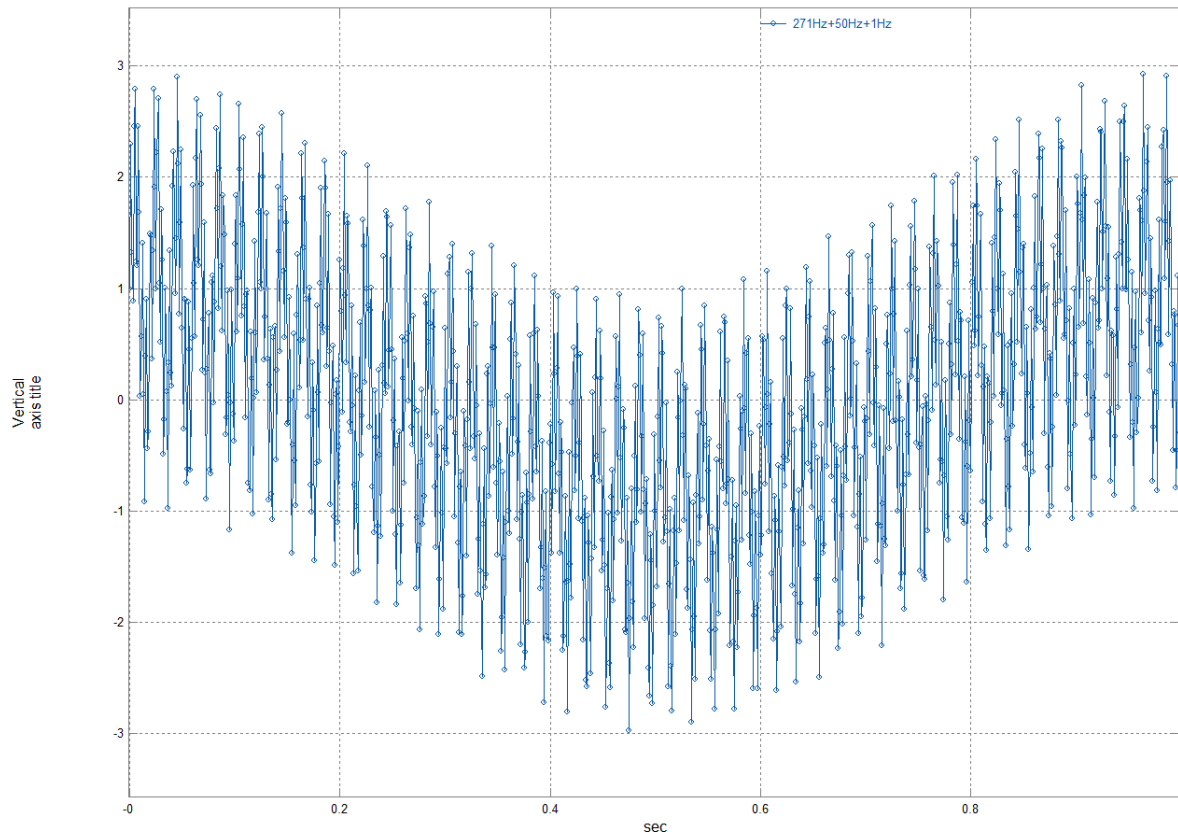
If the original data had an x axis in seconds the new x axis will be scaled in frequency (Hz).

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

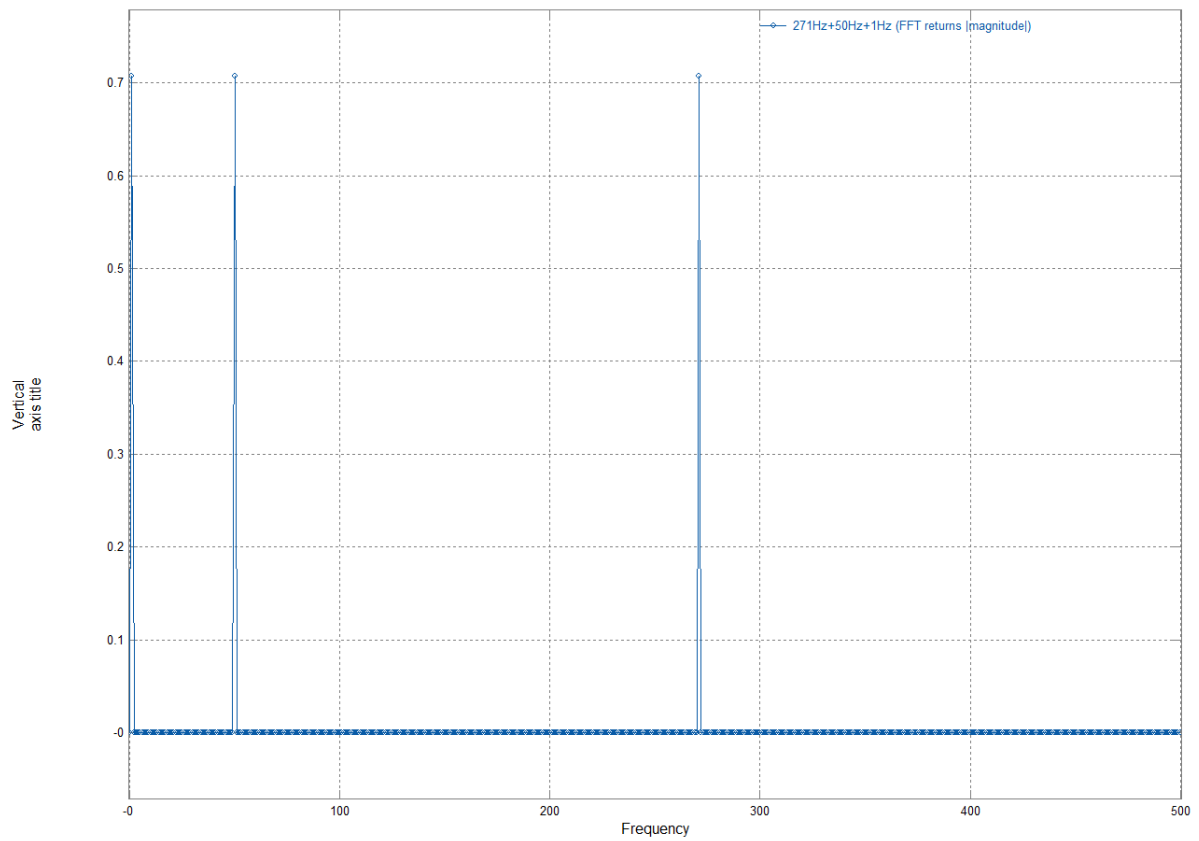
The result can either be viewed as a magnitude or log(magnitude) [in dB], using the log form compresses the dynamic range so that small values are easier to see.

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

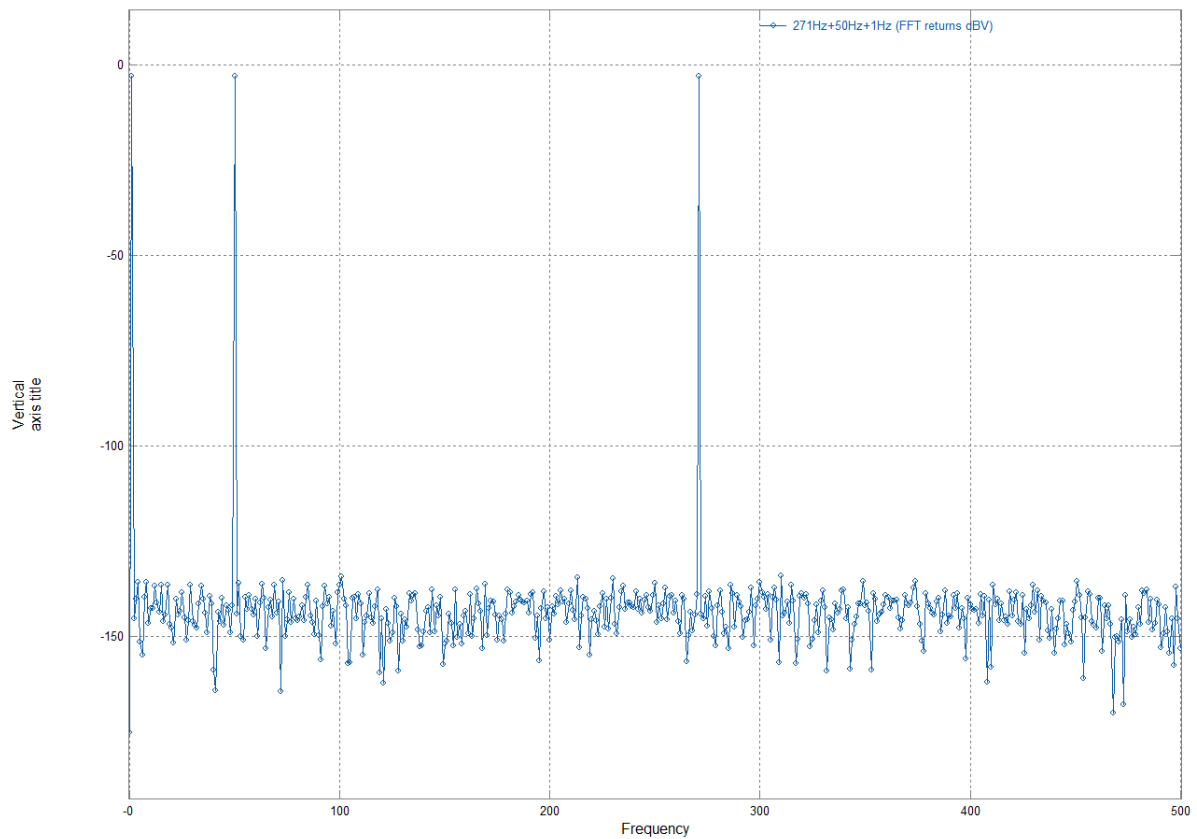
The first graph below is column 7 of this file which shows 271Hz, 50Hz and 1Hz sinewaves – in the time domain (with “None” selected as a filter) this gives:



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

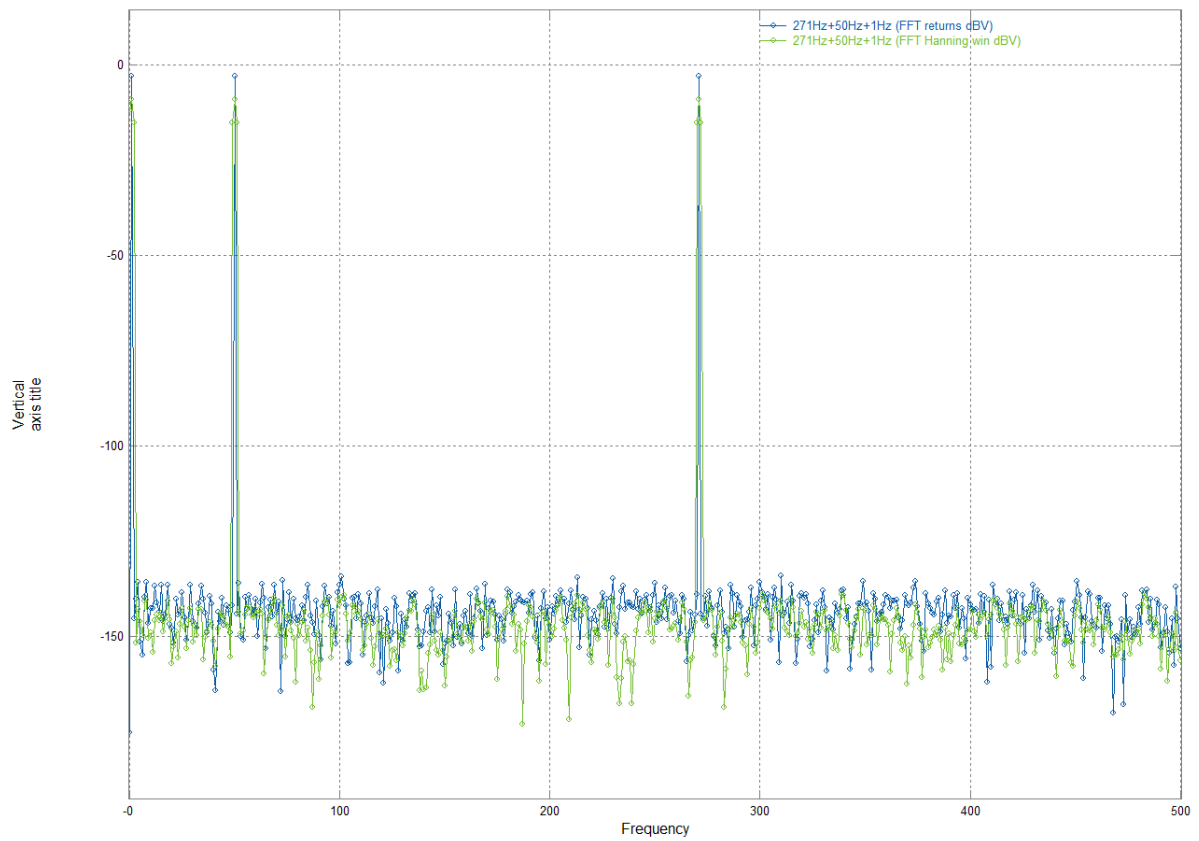


If we view the $\log(\text{magnitude})$ plot:



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.

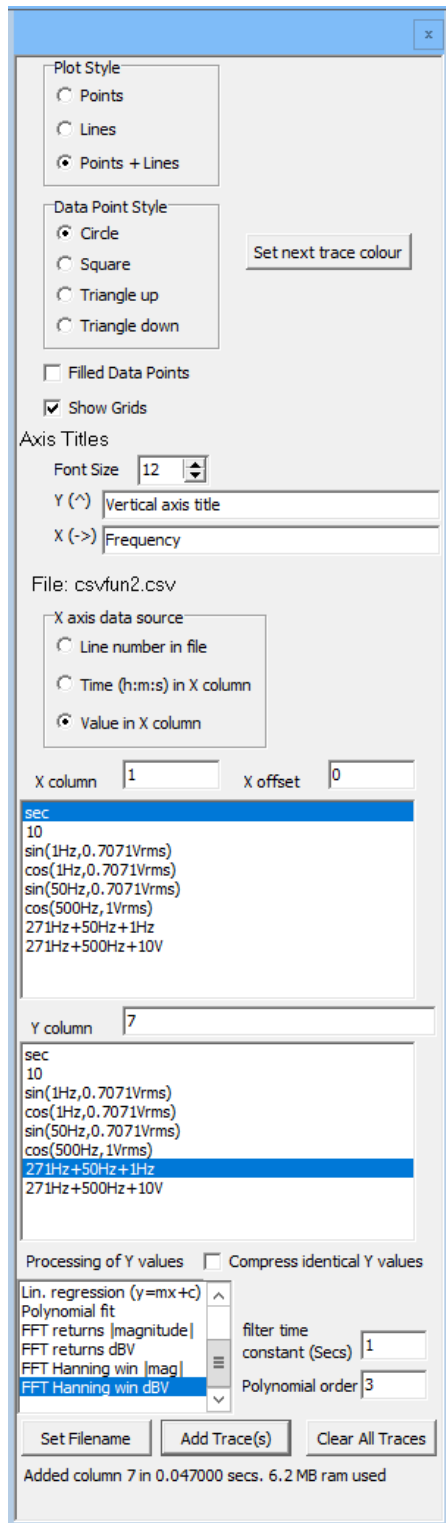
This effect (small non-zero values) can be reduced a little by using a “window” function and csvgraph offers the Hanning (sometimes called the Hann) window function and its effect can be seen below:



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.

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 is 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 (eg 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.

Installation

Csvgraph 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).

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

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