*Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.*
*http://www.simfit.org.uk*

## Basic plotting styles

SIMFIT offers a large choice of options to present data and results from model fitting and, to illustrate the basic plotting styles, an example from fitting three epidemic differential equations to data using program **deqsol** will be presented in the next collage.

**Figure 1: Standard Axes**

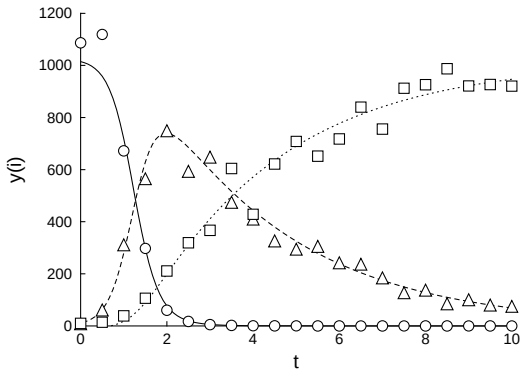



**Figure 2: Offset Axes**

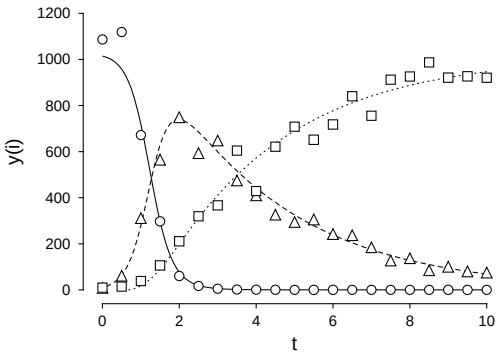



**Figure 3: Offset and Box**

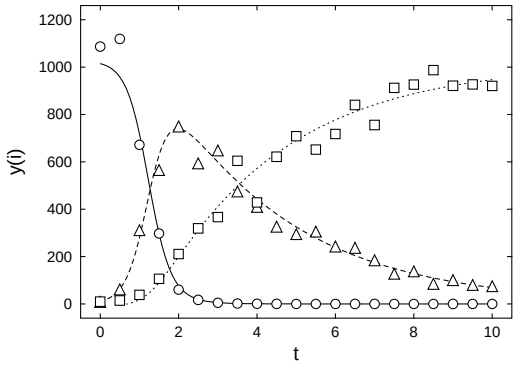



**Figure 4: Grid Lines**

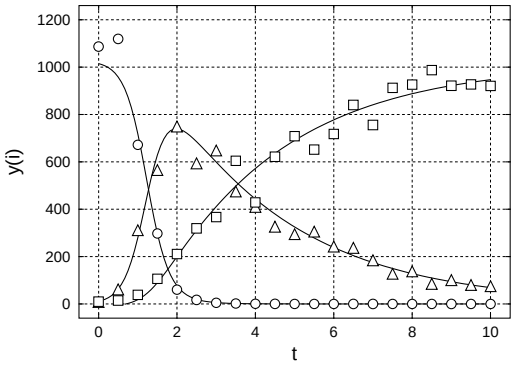



**Figure 1** Perhaps the usual style for scientific graphics, i.e., to simply display the data and axes with tick marks pointing inwards and no additional features as in the Standard Axes plot in Figure 1 above.

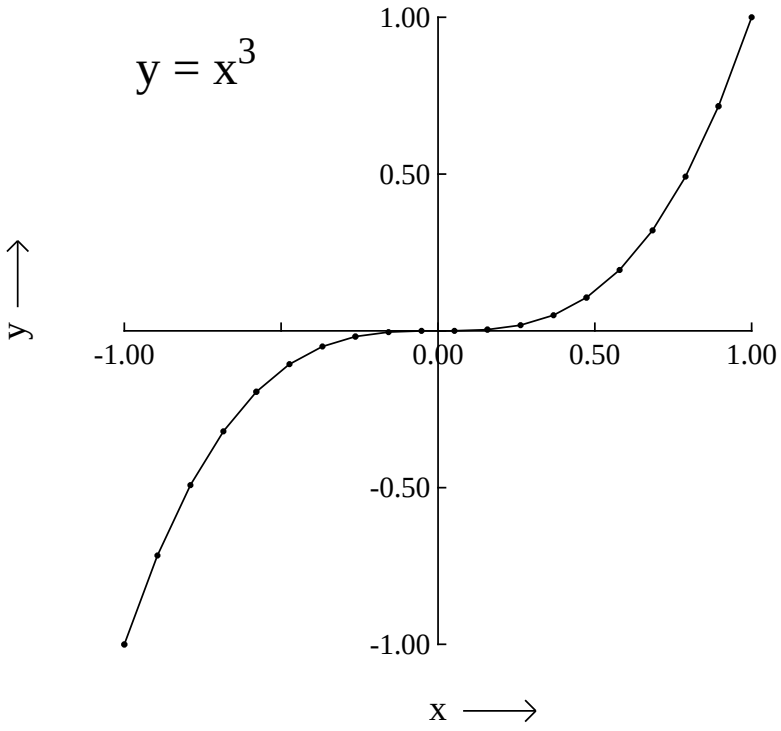
**Figure 2** However, the tick marks pointing inwards coupled with overlaying the data and fitted curve on top of the $X$ axis suggests that offset axes with tick marks pointing outwards, as in the Offset Axes plot of Figure 2 above, could be an improvement.

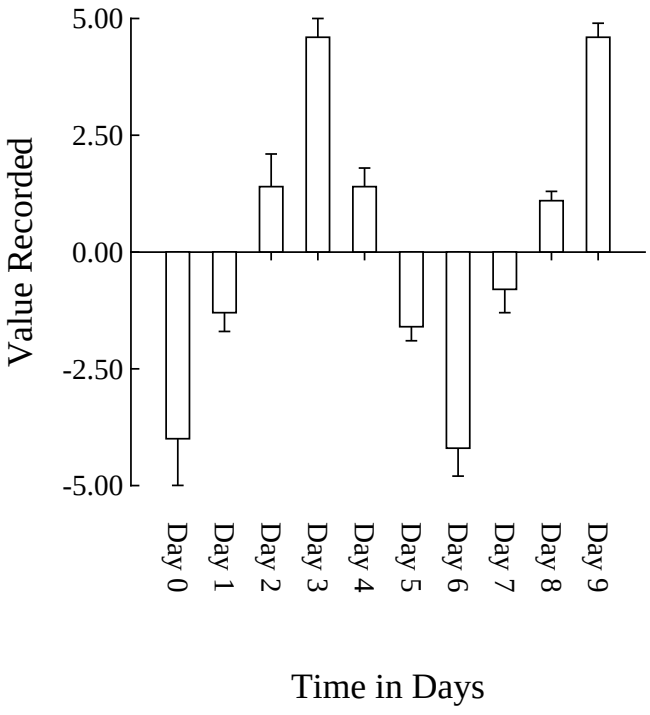
**Figure 3** Again, some regard a box round the data plotted, as in the Offset and Box plot of Figure 3 above, to be visually pleasing.

**Figure 4** Often grid lines, as in the Grid Lines example of Figure 4 above, help to establish coordinates, especially in calibration curves.

## Alternative axes and labels

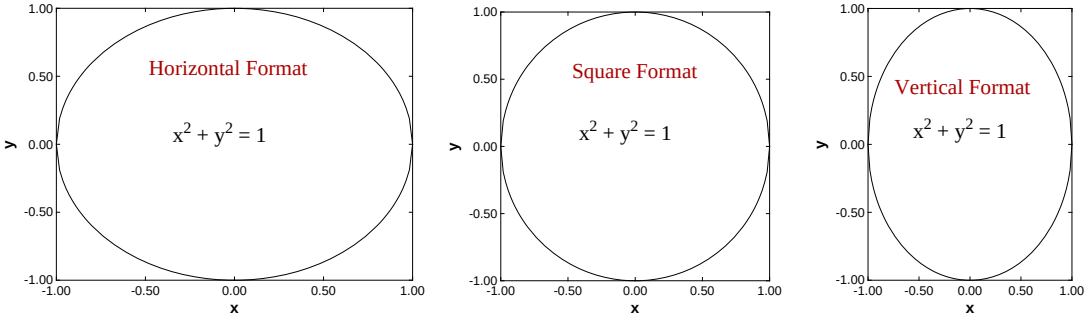
It is useful to move axes to make plots more meaningful, and it is sometimes necessary to hide labels, as with the plot of $y = x^3$ in the next figure, where the second $x$ and third $y$ label are suppressed. The figure also illustrates moving an axis in barcharts with bars above and below a baseline.



## Alternative sizes, shapes and clipping

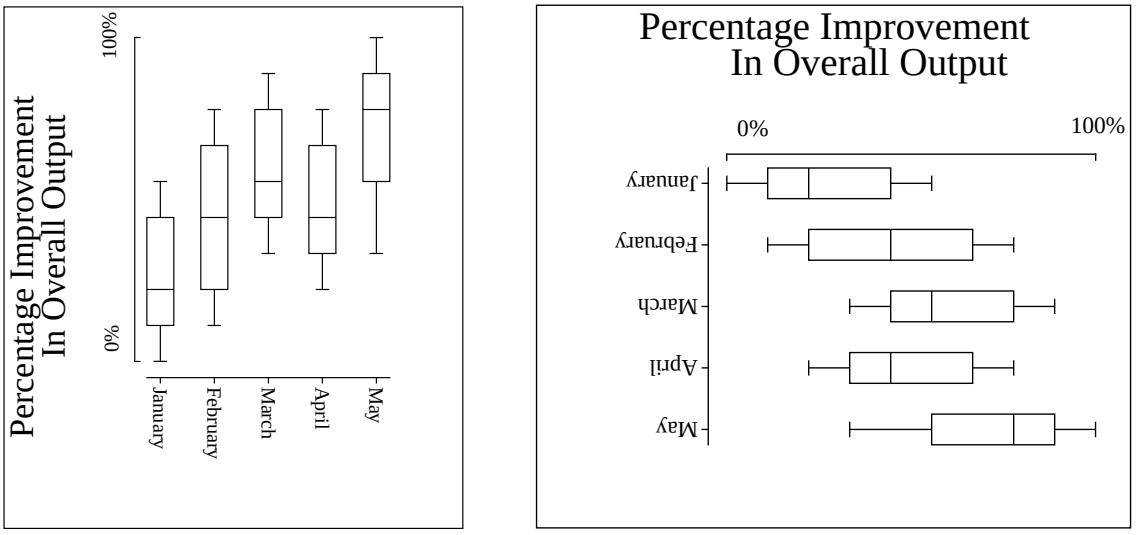
Plots can have horizontal, square or vertical format as in the next figure, and user-defined clipping schemes can be used. After clipping, SIMFIT adds a standard BoundingBox so all plots with the same clipping scheme will have the same absolute size but, when GSview/Ghostscript transforms ps into eps, it clips individual files to the boundary of white space and the desirable property of equal dimensions will be lost.




There is also a stretched format for long horizontal ribbon type graphs and a PostScript option to stretch white space without altering symbols and fonts, which is very useful with dense data sets such as dendrograms.

## Rotated and re-scaled graphs

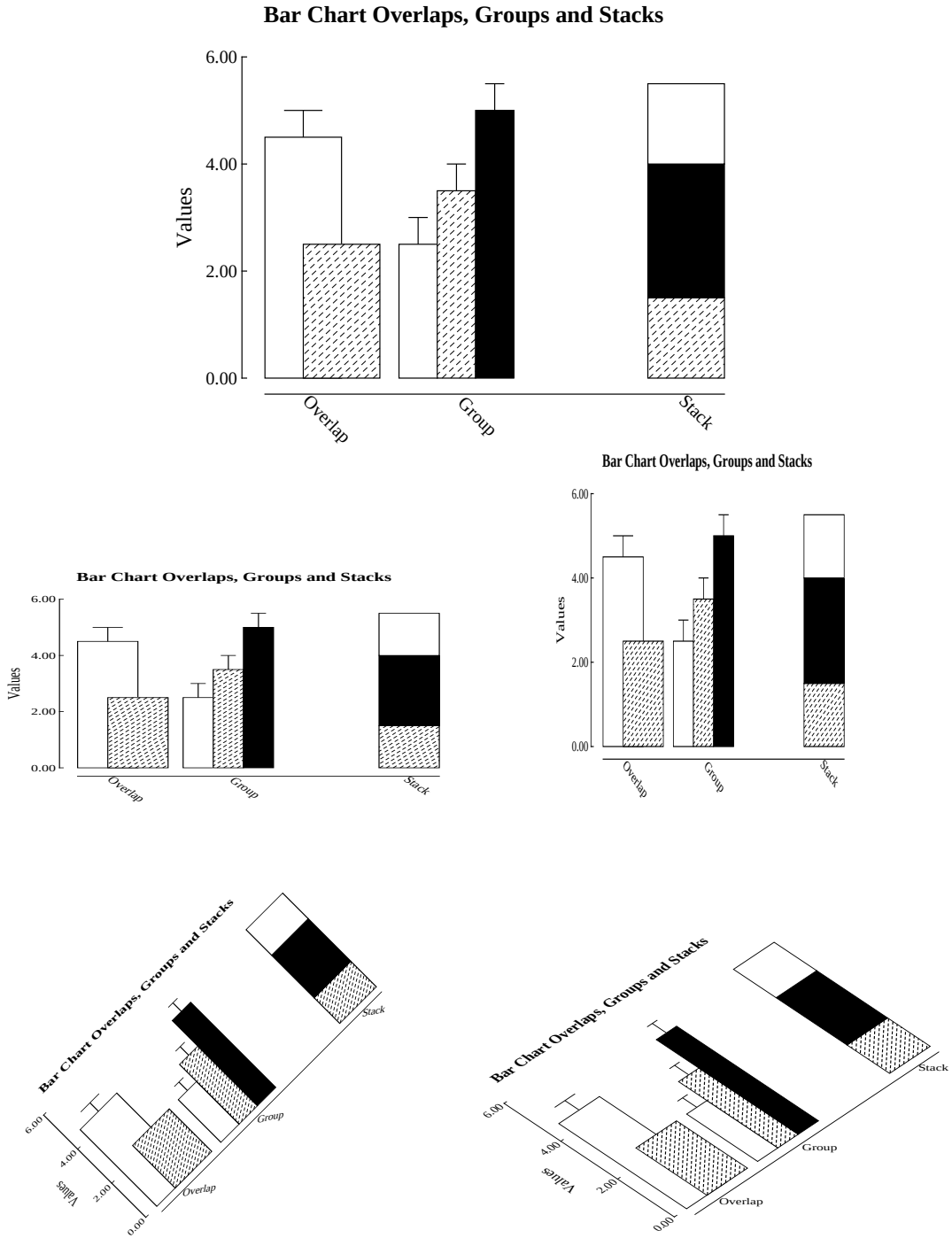
PostScript files can be read into **editps** which has options for re-sizing, re-scaling, editing, rotating, making collages, etc. In the next figure the box and whisker plot was turned on its side to generate a side-on barchart. To do this sort of thing you should learn how to browse a SIMFIT PostScript file in the SIMFIT viewer to read BoundingBox coordinates, in PostScript units of 72 to one inch, and calculate how much to translate, scale, rotate, etc.



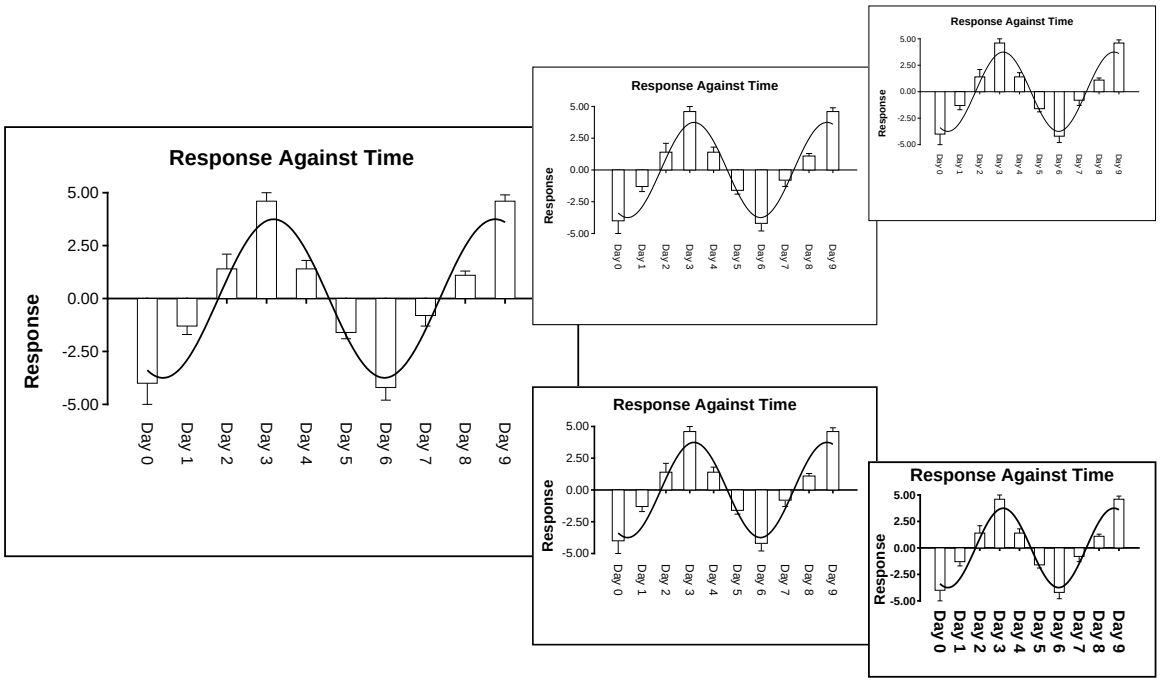

PostScript users should be warned that the special structure of SIMFIT PostScript files that allows extensive retrospective editing using **editps**, or more easily if you know how using a simple text editor like **notepad**, is lost if you read such graphs into a graphics editor program like Adobe Illustrator. Such programs start off by redrawing vector graphics files into their own conventions which are only machine readable.

## Changed aspect ratios and shear transformations

The barchart in the figure below was scaled to make the X-axis longer than the Y-axis and vice-versa, but note how this type of differential scaling changes the aspect ratio as illustrated. Since rotation and scaling do not commute, the effect created depends on the order of concatenation of the transformation matrices. For instance, scaling then rotation cause shearing which can be used to generate 3-dimensional perspective effects as in the last sub-figure.



## Reduced or enlarged graphs




It is always valuable to be able to edit a graph retrospectively, to change line or symbol types, eliminate unwanted data, suppress error bars, change the title, and so on. SimFIT PostScript files are designed for just this sort of thing, and a typical example would be altering line widths and font sizes as a figure is re-sized.

In the figure above the upper sub-figures are derived from the large figure by reduction, so the text becomes progressively more difficult to read as the figures scale down.

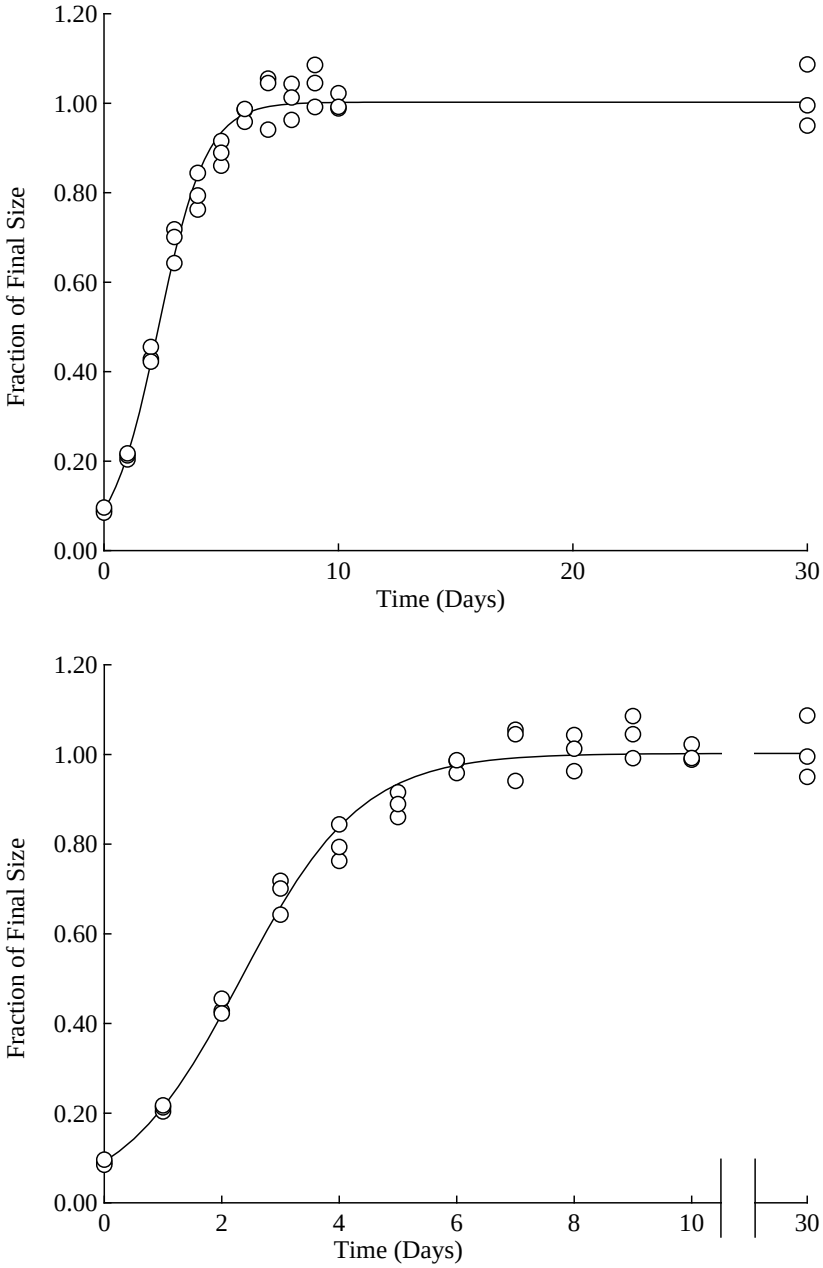
In the lower sub-figures, however, line thicknesses and font sizes have been increased as the figure is reduced, maintaining legibility. Such editing can be done interactively, but SimFIT PostScript files are designed to make such retrospective editing easy as described in the `w_readme.*` files and now summarized.

- Line thickness: Changing 11.00 setlinewidth to 22 setlinewidth doubles, while, e.g. 5.5 setlinewidth halves all line thicknesses, etc. Relative thicknesses are set by **simplot**.
- Fonts: Times-Roman, Times-Bold, Helvetica, Helvetica-Bold (set by **simplot**), or, in fact, any of the fonts installed on your printer.
- Texts: ti(title), xl(*x* legend), yl(*y* legend), tc(centered for *x* axis numbers), tl(left to right), tr(right to left), td(rotated down), ty(centered for *y* axis numbers).
- Lines: pl(polyline), li(line), da(dashed line), do(dotted line), dd(dashed dotted).
- Symbols: ce(i.e. circle-empty), ch(circle-half- filled), cf(circle-filled), and similarly for triangles(te, th, tf), squares(se, sh, sf) and diamonds(de, dh, df). Coordinates and sizes are next to the abbreviations to move, enlarge, etc.

If files do not print after editing you have probably added text to a string without padding out the key. Find the fault using the **GSview/Ghostscript**package then try again.

## Split axes

Sometimes split axes can show data in a more illuminating manner as in the figures below. The options are to delete the zero time point and use a log scale to compress the sparse asymptotic section, or to cut out the uninformative part of the best fit curve between 10 and 30 days.

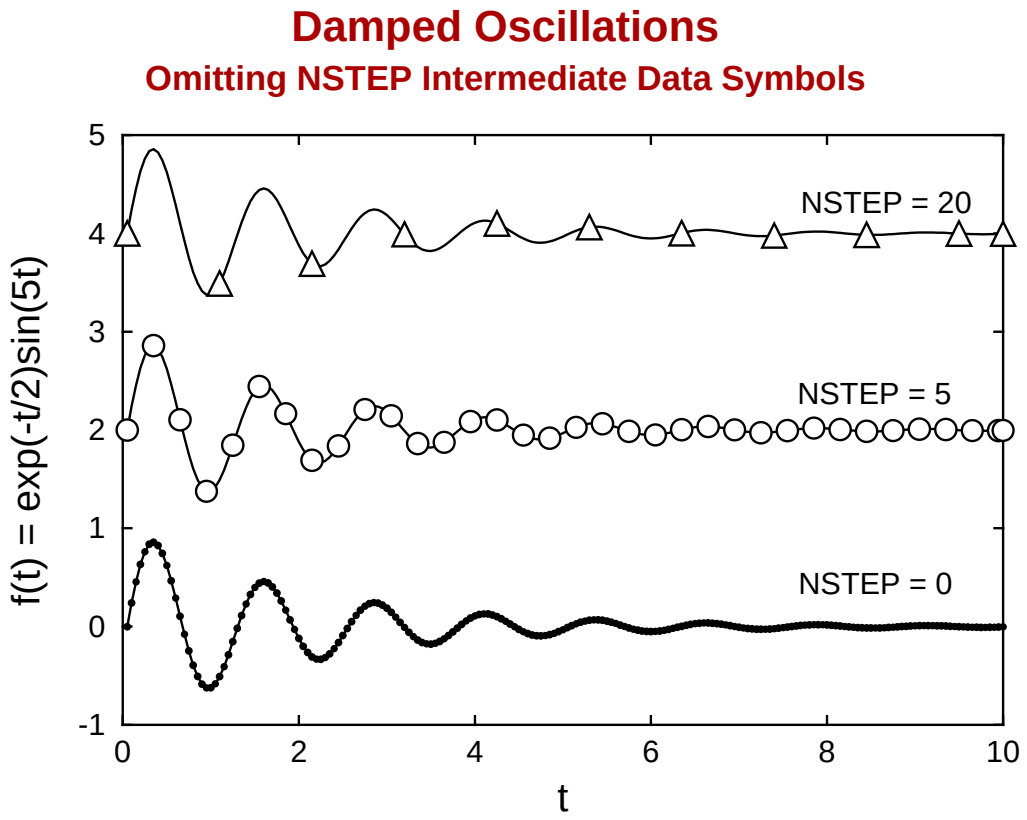



Windows users can do such things with enhanced metafiles (`*.emf`), but there is a particularly powerful way for PostScript users to split SIMFIT graphs in this way. When the SIMFIT PostScript file is being created there is a menu selectable shape option that allows users to chop out, re-scale, and clip arbitrary pieces of graphs, but in such a way that the absolute position, aspect ratio, and size of text strings does not change. In this way a master graph can be decomposed into any number of appropriately scaled slave sub-graphs. Then **editps** can be used to compose a graph consisting of the sub-graphs rearranged, repositioned, and resized in any configuration. The lower figure was created in this way after first adding the extra lines shown at the splitting point.

## Stepping over intermediate data points

Sometimes it is advantageous to step over intermediate data points and, for clarity, only plot points at intervals through the data set. For instance, there may be a large number of machine generated data points, far more than is required to define a curve by the usual technique of joining joining up successive points by straight lines. Plotting all the points in such cases would slow down the graphics display and, more importantly, could lead to very large PostScript output files. Again, there would be times when a continuous curve is required to illustrate a function defined by a reasonable number of closely spaced data points, but symbols at all points would obscure the curve, or might only be required for labeling purposes.

The next figure illustrates a case in point.




Here even small symbols, like dots in the bottom figure where no points are omitted, would obscure the plot and the middle plot with intermediate groups of five suppressed, could also be regarded as too crowded, while the upper plot has sufficient symbols even with twenty points stepped over to identify the plot, say in an information panel.

Of course such graphs can easily be created by pre-processing the data files before submitting for plotting. However, SimFIT has a special facility to do this interactively. The technique required to create plots like the above figures is to submit two files with identical data, one to be plotted as a line joining up data to create the impression of a continuous curve, the other to be plotted as symbols. Then, from the [Data] menu when the plot is displayed, a parameter $NSTEP$ can be defined, where $NSTEP$ is the number of intermediate points in each group to be stepped over.

Observe that, when using this technique, the first and last data points are always plotted to avoid misunderstanding.