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

There are essentially three types of image files as follows.

1. **Bitmaps and compressed bitmaps**

   Raw bitmaps (e.g., `.bmp`) are used to record the characteristics of every pixel in a display or hardcopy, typically a digital photograph. They are limited by the resolution of the captured image and are usually large, also the images break up by pixelation if they are enlarged. They are generally compressed into alternative formats (e.g., `.jpg`, or `.png`) where a certain loss of quality is offset by a great decrease in size. In scientific work they are mainly used for complicated diagrams, e.g., photographs of microscopic sections, where there are no distinct objects such as titles, legends, lines, curves, plotting symbols, etc.

2. **Vector graphics**

   Often scientific graphs consist only of lines for axes, curves, plotting symbols, and text for titles and legends with featureless backgrounds, so that storing a bitmap would be wasteful of space. However the main advantage of vector formats (e.g., `.eps`, `.svg`, or `.emf`) is that they are device-independent so they can be displayed or printed at any resolution with no loss of information. They can also be used to generate compressed bitmap files, but it should be noted that scientific graphs can sometimes generate vector hardcopy that is even more bulky than the corresponding bitmap if there are very large numbers of objects being plotted.

3. **Embedded bitmaps**

   Unfortunately vector graphics files can often consist of wrappers containing bitmaps which leads to files with a vector file extension that are actually no better than bitmaps. Some graphics programs export supposed `.eps` files as embedded bitmaps, so losing many of the advantages of true vector files. Also note that `.pdf` files created from vector `.eps` files using GhostScript retain some of the characteristics of actual vector files, but many programs simply distill graphs into `.pdf` files as embedded bitmaps.

Users of the SimFIT package are strongly urged to save all graphs as SimFIT `.eps` files because they have the following advantages.

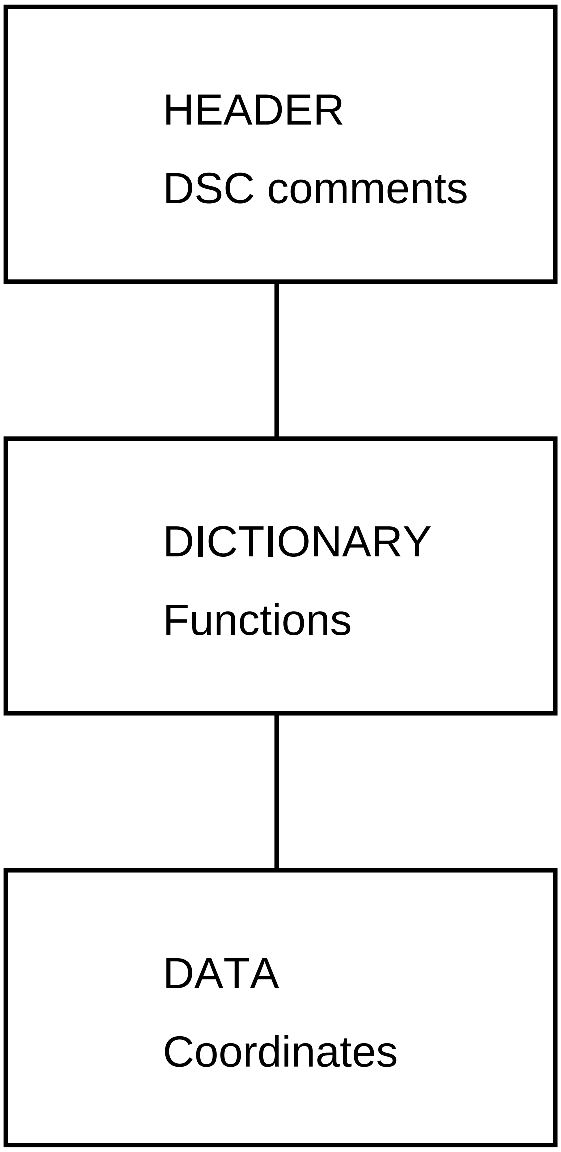
- They are true encapsulated PostScript vector files consisting of a single page with a BoundingBox.
- They are structured in such a way that they can be edited using a text editor to change dimensions, line types, symbols, colours, titles and legends, etc.
- They can be used retrospectively to create alternative types of image files.
- SimFIT provides facilities to edit such files, or make various types of collages.

Although SimFIT can make and edit `.eps` files with no additional software it will be found that, in order to make full use of the PostScript opportunities, it is necessary to download and install the GhostScript package, and also advisable to download and install the GSview package. GhostScript can be used to transform `.eps` files into other graphics formats, while GSview can be used to view and print them.

The sections of Simfit eps files

In order to be able to edit SimFIT `.eps` files it is necessary to appreciate that there are three distinct sections. The first section contains the BoundingBox coordinates that must be present in `.eps` files to specify the size of such one page graphs along with the document structuring comments. The second section is a dictionary containing the functions that can be used in SimFIT `.eps` files to draw lines and plotting symbols. The third section is a list of coordinates and colors where lines, symbols and text strings are to be drawn. This structure is summarized in the next diagram.

SimFIT .eps file structure



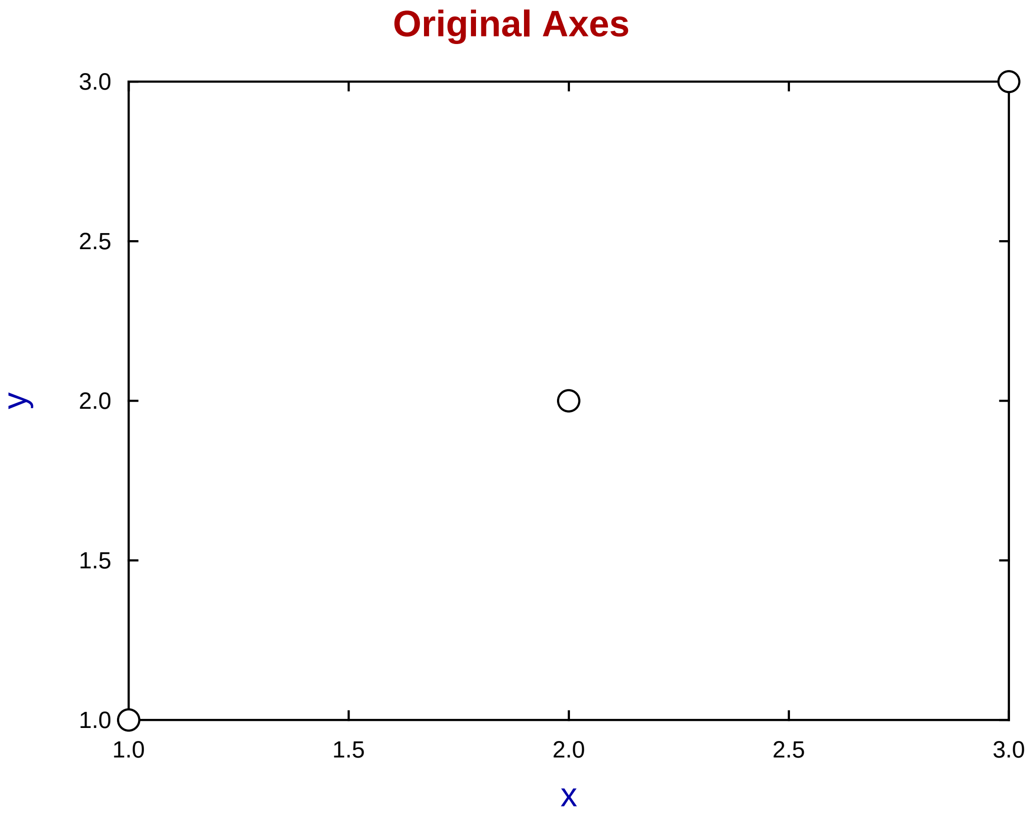
When editing SimFIT PostScript files retrospectively the following rules must be obeyed until you become familiar with the structure.

1. Never change anything in the HEADER or DICTIONARY sections.
2. You can edit the title, legends, colors, and other display features in the DATA section.
3. Perform editing a step at a time and check the effects using GSview.
4. Make sure you save the file at each stage of editing.

Extensive descriptions about editing the DATA section will be found in the SimFIT reference manual `w_manual.pdf` available from the SimFIT website, but a number of simple examples will be given in the succeeding tutorials by way of a more gentle introduction.

A simple example

As an example consider the following simple default graph which is easily constructed by requesting to plot the coordinates (1,1), (2,2), and (3,3) using program **simplot**



Some of things a user might wish to change retrospectively could be as follows.

- Change the title
- Change the legends
- Change the plotting symbols
- Change the colors

Now SIMFIT provides a PostScript editor program **editps** to perform such tasks but, in actual practise, it is easier to edit the PostScript file in a text editor such as **notepad** because of these three special features found in SIMFIT Postscript files.

1. They are ASCII text files and can be edited in any text editor.
2. They are encapsulated PostScript files (`.eps`) and describe just a single page.
3. They have been uniquely designed to make such editing very easy.

As you will need to view the results of editing you will need a PS viewer such as **GSview** which requires an installed copy of **GhostScript**.

The header section

The following code is contained in the header.

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox:    72   252   520   588
%%Creator: Simfit Version 7.2.8 (simfit.org.uk)
%%Title: colours=72/ISOLatin1Encoding/Accents/special/PSfrag/dict=300
%%CreationDate: Saturday, 15 April 2017
%%EndComments
%
%Start of SIMFIT PostScript file
%
save %save current state before clipping, etc.
    70   250   522   250   522   590    70   590%#clipping
newpath moveto lineto lineto lineto closepath clip newpath
   72.00  252.00 translate    0.07    0.07 scale    0.00 rotate%#portrait
   12.00 setlinewidth 0 setlinecap 1 setlinejoin [] 0 setdash
    2.50 setmiterlimit
%
%prolog(1) to (6) can be used by DVIPS as a header
%****cut the invariant prolog/header out from here
```

It includes the document structuring comments together with some technical instructions, and this section would only be edited by experienced users.

The dictionary section

This contains definitions for all the plotting functions, colors and fonts required to display the data contained in the data section, and this section would only be edited by very experienced users.

```
/SIMFIT 300 dict def SIMFIT begin
%
% prolog(1): definitions
%
/C{copy}def /D{def}def /E{exch}D /F{findfont}D /GR{grestore}D
/GS{gsave}D /M{moveto}D /N{newpath}D /P{pop}D /R{rmoveto}D
/S{scalefont setfont}D /d{dup}D /i{putinterval}D /p{put}D
%
% prolog(2): construct Greek/maths font
%
...
...
...
/ty-font /Helvetica      D%text right y-mid
/tz-font /Helvetica      D%text left  y-mid
/ti-size 204 D /xl-size 187 D /yl-size 187 D /zl-size 187 D
/tc-size 144 D /td-size 144 D /tl-size 144 D /tr-size 144 D
/ty-size 144 D /tz-size 144 D
/sb-size 0.75 D /sp-size 0.75 D%sub/superscript expansion
/y-down -0.33 sb-size mul D /y-up 0.33 sp-size div D%sub/sup shift
%
foreground thickness setlinewidth
```

The data section

This is first given in full and then the sections that are most likely to be edited are discussed in detail.

```
/background{c15}D
2 setlinecap
1070 671 5959 671 5959 4215 1070 4215 4 pc%#8
/ty-size ty-size   0.900 mul def
/tl-size tl-size   0.900 mul def
1070 671 1118 671 li%#4
5959 671 5911 671 li%#4
(1.0) 974 671 ty%#()2
(000) fx
1070 1557 1118 1557 li%#4
5959 1557 5911 1557 li%#4
(1.5) 974 1557 ty%#()2
(000) fx
1070 2443 1118 2443 li%#4
5959 2443 5911 2443 li%#4
(2.0) 974 2443 ty%#()2
(000) fx
1070 3329 1118 3329 li%#4
5959 3329 5911 3329 li%#4
(2.5) 974 3329 ty%#()2
(000) fx
1070 4215 1118 4215 li%#4
5959 4215 5911 4215 li%#4
(3.0) 974 4215 ty%#()2
(000) fx
/ty-size ty-size   1.111 mul def
/tl-size tl-size   1.111 mul def
/tc-size tc-size   0.900 mul def
/tl-size tl-size   0.900 mul def
1070 671 1070 719 li%#4
1070 4215 1070 4167 li%#4
(1.0) 1070 462 tc  %#()2
(000) fx
2292 671 2292 719 li%#4
2292 4215 2292 4167 li%#4
(1.5) 2292 462 tc  %#()2
(000) fx
3515 671 3515 719 li%#4
3515 4215 3515 4167 li%#4
(2.0) 3515 462 tc  %#()2
(000) fx
4737 671 4737 719 li%#4
4737 4215 4737 4167 li%#4
(2.5) 4737 462 tc  %#()2
(000) fx
5959 671 5959 719 li%#4
5959 4215 5959 4167 li%#4
(3.0) 5959 462 tc  %#()2
(000) fx
/tc-size tc-size   1.111 mul def
/tl-size tl-size   1.111 mul def
/ti-size ti-size   1.000 mul def
c4
(Original Axes) 3195 4467 ti%#title
(0000000000000) fx
/ti-size ti-size   1.000 mul def
/xl-size xl-size   1.000 mul def
c1
(x) 3515 192 xl%#x legend
```

```
(0) fx
/xl-size xl-size   1.000 mul def
/yl-size yl-size   1.000 mul def
(y) 501 2443 yl%#y legend
(0) fx
/yl-size yl-size   1.000 mul def
0 setlinecap
c0
1070 671 59 ce%#2
3514 2443 59 ce%#2
5959 4215 58 ce%#2
```

Editing the title and legends

Searching for the key word title locates the next section.

```
c4
(Original Axes) 3195 4467 ti%#title
(0000000000000) fx
/ti-size ti-size   1.000 mul def
/xl-size xl-size   1.000 mul def
c1
(x) 3515 192 xl%#x legend
(0) fx
/xl-size xl-size   1.000 mul def
/yl-size yl-size   1.000 mul def
(y) 501 2443 yl%#y legend
(0) fx
```

Interchanging the colors c1(blue) and c4(red) and altering the title and can then be done leading to the replacement section shown next.

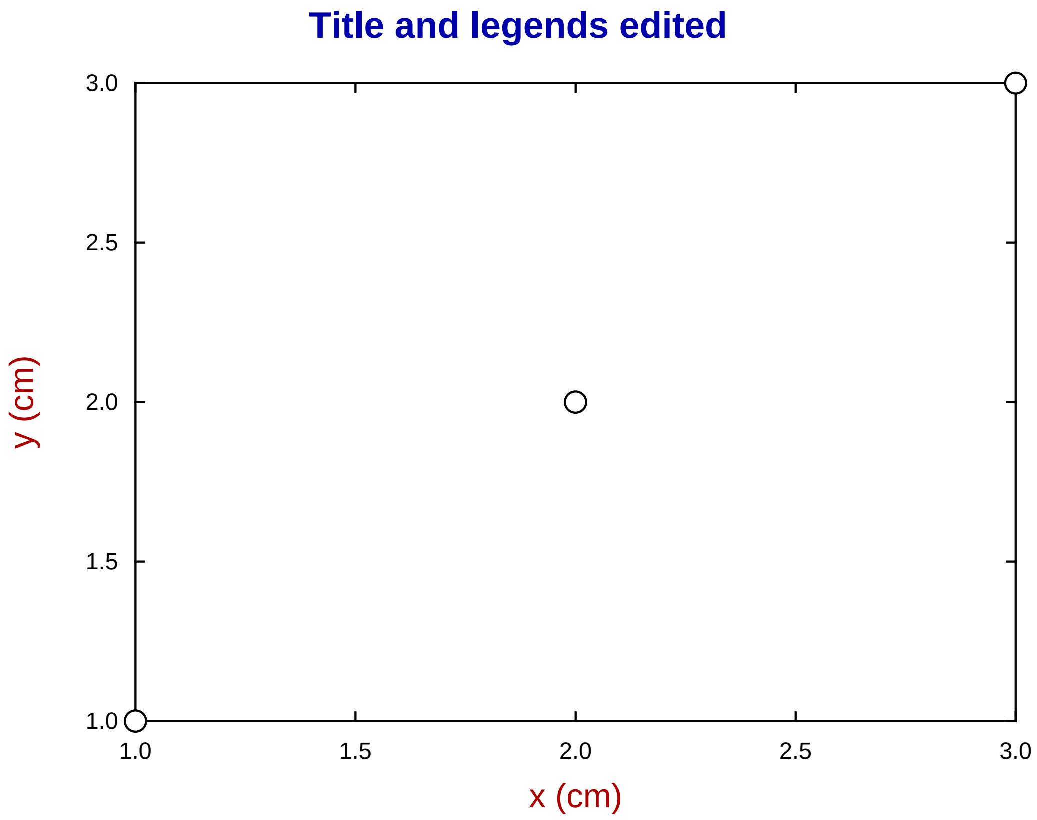
```
c1
(Title and legends edited) 3195 4467 ti%#title
(000000000000000000000000) fx
/ti-size ti-size   1.000 mul def
/xl-size xl-size   1.000 mul def
c4
(x \(mm\)) 3515 192 xl%#x legend
(00000000) fx
/xl-size xl-size   1.000 mul def
/yl-size yl-size   1.000 mul def
(y \(mm\)) 501 2443 yl%#y legend
(00000000) fx
```

Two comments are needed in order to understand the results of this editing.

1. When a text string such as a title or legend is edited it is necessary to make sure that the character key string underneath the text is padded or contracted if required to make sure the key has at least as many characters (in this case the 0 characters denoting a normal font) as the text string.

2. Since character strings in PostScript are enclosed in round brackets it is necessary to prefix any brackets introduced inside the string by the backslash escape character.

The edited file is shown next.



Editing line and symbol types

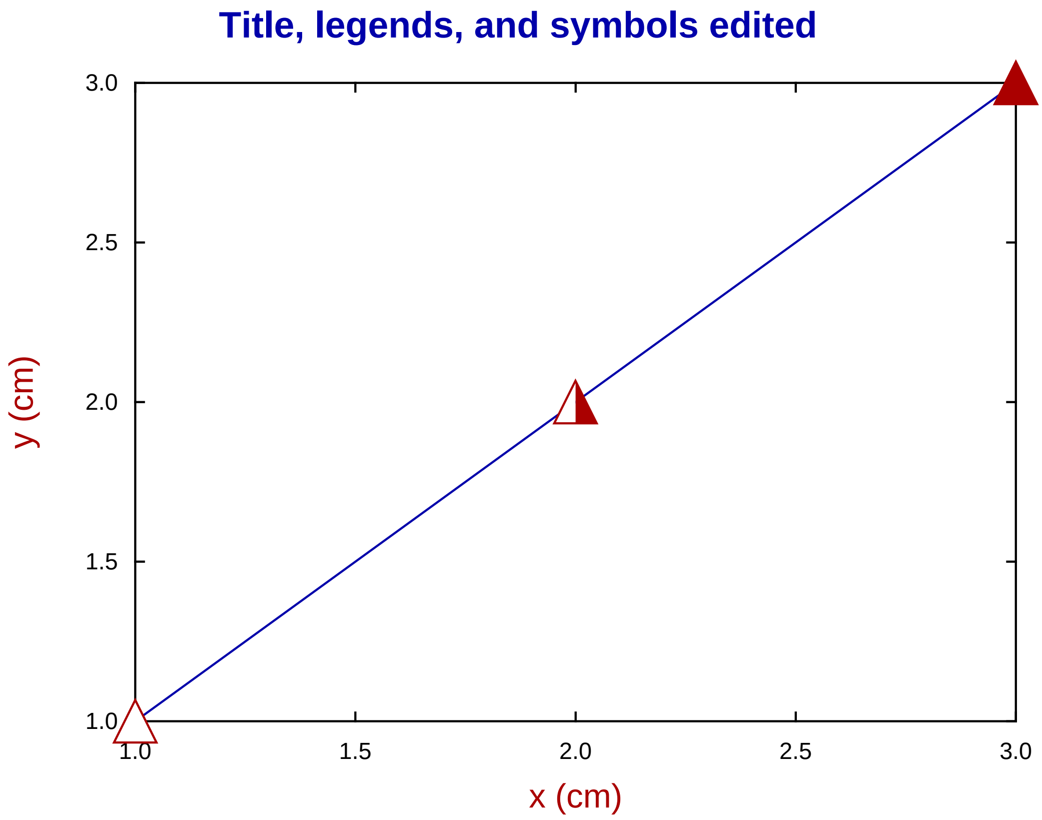
It is frequently required to change the size, colors, and types of lines and plotting symbols and the code in the data section defining the black (c0) empty circles (ce) is as follows.

```
c0
1070 671 59 ce%#2
3514 2443 59 ce%#2
5959 4215 58 ce%#2
```

For example, using the command c1 and li to add a blue line, altering c0 to c4 to change color to red, then changing empty circles (ce) into empty triangle (te), half-filled triangle (th), and filled triangle (tf), together with doubling the size of the symbols from 59 to 118, as in this code

```
c1
1070 671 5959 4215 li
c4
1070 671 118 te%#2
3514 2443 118 th%#2
5959 4215 118 tf%#2
```

creates the next graph.



Clearly users will require much more information to be able to edit all the features of SIMFIT PostScript files, and the necessary details along with numerous worked examples follows.