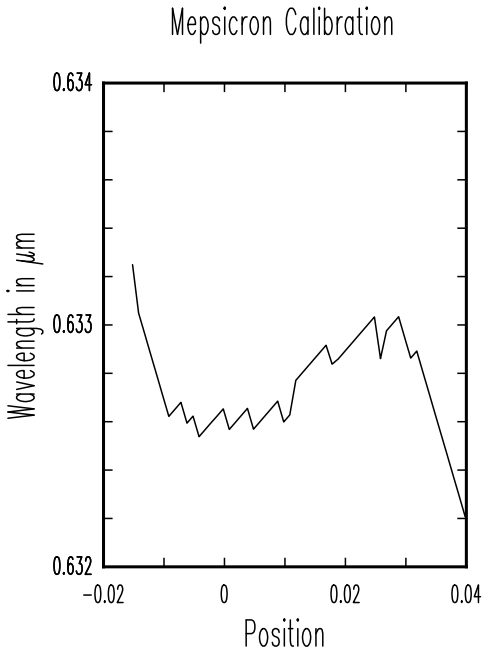
Chapter 5

Sample Programs

This chapter contains printouts of several example drawing or plotting files with the corresponding printed figure that the file generates. These for the most part are contributed files of actual figures used in publications.

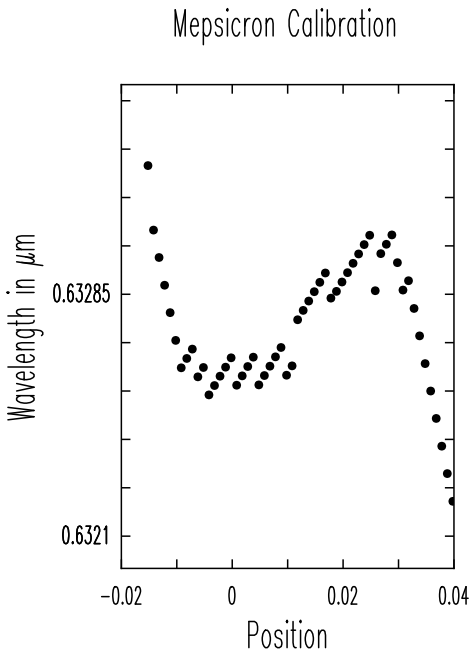
5.1 simple.spt

```
/* demo of simplest possible data */
/* plotting using all the defaults */

#include <splot.h>
double *data;
main()
   {
   readdata("demo\data1.dat",data);
   plotdata(data);
   label(LOWER,"Position");
   label(LEFT,"Wavelength in !m!m");
   text(6.20,22.98,"Mepsicron Calibration");
   }
```



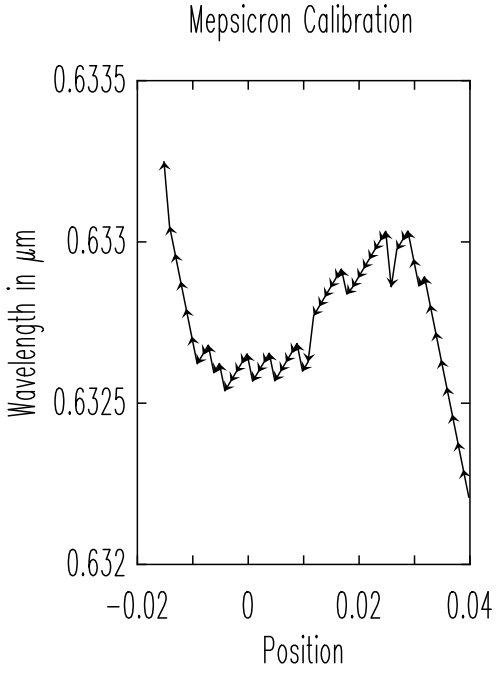
5.2 simple2.spt

```
/* demo of simple data plotting but*/
/* over riding some of the defaults */

#include <splot.h>
double *data;
main()
   {
   readdata("demo\data1.dat",data);
   set(PLOTTYPE,SYMBOLS);
   set(AXESCLIP,ON);
   axes_box(11,16,-0.02,0.632,0.04,0.6335);
   plotdata(data);
   label(LOWER,"Position");
   label(LEFT,"Wavelength in !m!m");
   text(6.20,22.98,"Mepsicron Calibration");
   }
```



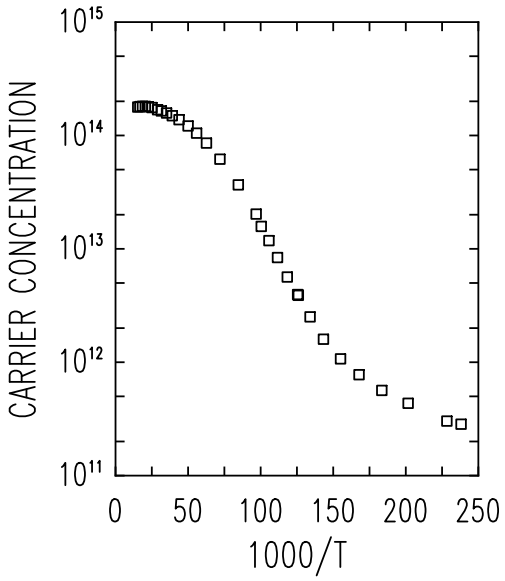
5.3 simple3.spt

```
/* demo of simple data plotting but*/
/* over riding some more of the defaults */

#include <splot.h>
double *data;
main()
   {
   readdata("demo\data1.dat",data);
   set(PLOTTYPE,SYM_LINES);
   set(CURSYMBOL,ARROW);
   set(AXESCLIP,ON);
   axes_box(11,16,-0.02,0.632,0.04,0.6335);
   tickmarks(YAXES,0.632,0.6325,0.633,0.6335);
   tickmarks(XAXES);
   ticklabel();
   label(LOWER,"Position");
   label(LEFT,"Wavelength in !m!m");
   text(6.20,22.98,"Mepsicron Calibration");
   set(FONTMULT,2);
   plotdata(data);
   }
```



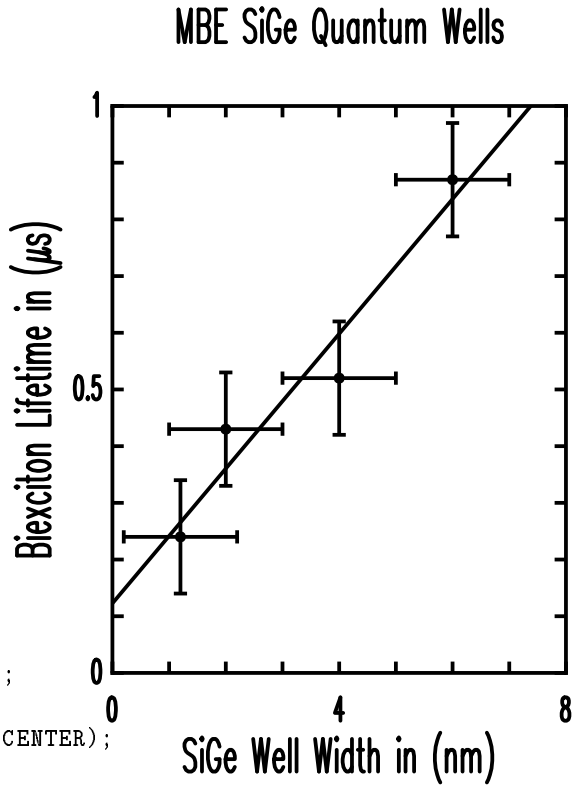
5.4 inverax.spt

```
/* This file illustrates the use of inverse and log axes */
\index{axes, log example}
\index{axes, inverse example}

#include <splot.h>
double *data;
main()
   {
   set(LINEWIDTH,0.06);
   set(CURSYMBOL,OSQUARE);
   set(AXESCLIP,ON);
   set(FONTASPECT,1.5);
   set(AXESTYPE,INVXLOGY);
   axes_box(12,15,1e20,0.001,4,10);
   set(FONTWIDTH,0.8);
   tickmarks();
   ticklabel(BOTTOM,1e+20,"0",20,"50",10,"100",6.6666,"150"
             ,5,"200",4,"250");
   ticklabel(LEFT,1e-3,"10^11^",1e-2,"10^12^",1e-1,"10^13^",1,"10^14^",10,"10^15^");
   set(XMULT,1);
   set(FONTWIDTH,0.9);
   label(BOTTOM,"1000/T");
   label(LEFT,"CARRIER CONCENTRATION");
   set(PLOTTYPE,SYMBOLS);
   readdata("demo\tb4_2_2.dat",data);
   plotdata(data);
   }
```

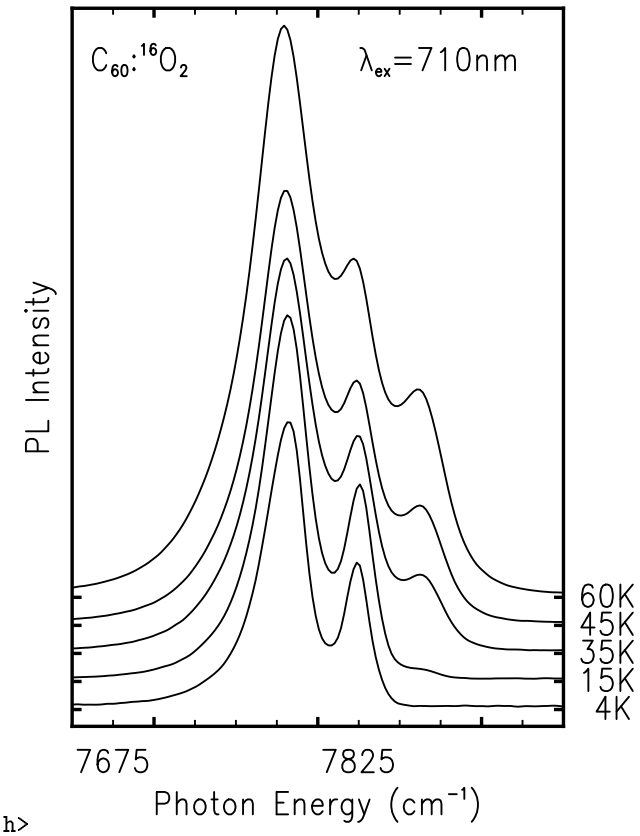


5.5 errorbar.spt

```
#include <splot.h>

/* This example file illustrates the use of errorbars and straight */
/* line fits to data. It also uses a small internal data array rather */
/* than reading data in from a file */

double data[4][4] = {
   1.2, 0.24, 0.1, 1,
   2.0, 0.43, 0.1, 1,
   4.0, 0.52, 0.1, 1,
   6.0, 0.87, 0.1, 1};

double yint,slope;


main()
   {
   set(LINEWIDTH,0.1);
   set(PLOTTYPE,SYMBOLS);
   axes_box(12,15,0,0,8,1.0);
   tickmarks();
   set(FONTWIDTH,0.5);
   ticklabel();
   set(FONTWIDTH,0.7);

   label(BOTTOM,"SiGe Well Width in (nm)");
   label(LEFT,"Biexciton Lifetime in (!m!s)");

   text(10.03,22.56,"MBE SiGe Quantum Wells",CENTER);

   /* draw the data points  */
   drawdata(data,0,1);

   /* put in the error bars for the x values */
   errorbars(XVALS,data,0,1,3);
   /* put in the error bars for the y values */
   errorbars(YVALS,data,0,1,2);

   /* fit the best line to the data and return the slope and intercept */
   fitline(data,0,1,&yint,&slope);
   print(slope);
   print(yint);
   }
```



5.6 manyspec.spt

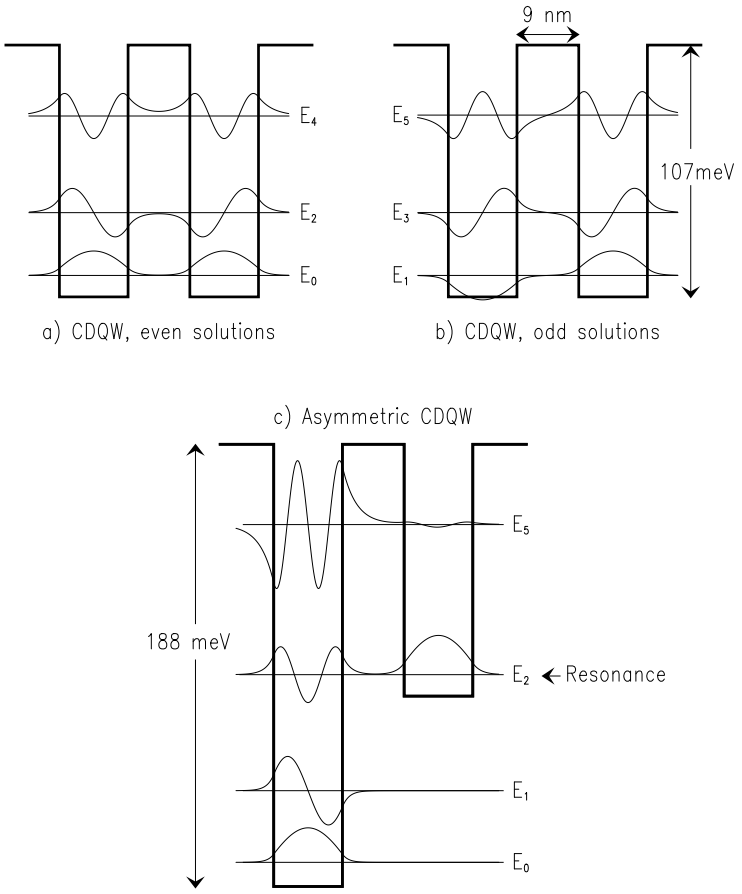



```
#include <splot.h>
/* Created by Scott Wilson 02/05/93 */
/* this splot file generates an xy
   plot of 5 data sets. Each curve
   has the same x scale and y scale
   but have been offset from each
   other in the y direction by an
   equal amount
*/
/* define the data arrays */
double *pldata1, *pldata2, *pldata3, *pldata4, *pldata5;
main()
  {

   /* define a tic length  */
   /* major and minor tics */

   double majortic,minortic;
   majortic=0.25;
   minortic=0.17;

   /* set up some default values */
```

```
set(FONTASPECT,1.2);
set(FONTWIDTH,0.8);
set(LINECOLOUR,BLACK);
set(AXESCLIP,ON);
set(LINEWIDTH,0.09);
set(PLOTTYPE,LINES);

/* load in the data */

readdata("demo\4kc.dat",pldata1);
readdata("demo\15kc.dat",pldata2);
readdata("demo\35kc.dat",pldata3);
readdata("demo\45kc.dat",pldata4);
readdata("demo\60kc.dat",pldata5);

/* define an axis box */

axes_box(13,19,7650,-3000,7950,125000);

/* set up some major tics*/

set(TICKLENGTH,majortic);

/* for the x-axis    */

tickmarks(XAXES,7700,7800,7900);

/* for the y-axis    */
/* note only denoting the zero point
for each curve */

tickmarks(YAXES,0,5000,10000,15000,20000);

/* set up some minor tics */

set(LINEWIDTH,0.05);
set(TICKLENGTH,minortic);

/* for the x axis */

tickmarks(XAXES,7675,7725,7750,7775,7825,7850,7875,7925);

/* place some ticklabels on the axes */

ticklabel(BOTTOM);
ticklabel(RIGHT,0," 4K",5000,"15K",10000,"35K",15000,"45K",20000,"60K");

 /* place some labels on the axes */

label(BOTTOM,"Photon Energy (cm^-1^)");
label(LEFT,"PL Intensity");
```

```
/* place some comments in the figure */

text(3.98,21.11,"C_60_:^16^O_2_");
text(11.10,21.11,"!l!_ex_=710nm");

/* put the data into the picture */

drawdata(pldata1);

set(YSHIFT,5000);
drawdata(pldata2);
set(YSHIFT,10000);
drawdata(pldata3);
set(YSHIFT,15000);
drawdata(pldata4);
set(YSHIFT,20000);
drawdata(pldata5);
}
```

5.7 invisax.spt




```
/* This example shows how invisible axes can be used to establish */
/* an alternate coordinate system and still use the drawdata function */
/* to draw the curves */

/*draws quantum wells and wave functions */
#include <splot.h>
double *data1;
double *data2;
double *data3;
double *data4;
double *data5;
```

```
main()
   {
   set(LINECOLOUR,INVIS);
   set(FONTWIDTH,0.6);
   set(FONTASPECT,1.4);
   set(LINEWIDTH,0.02);
   /* read in data generated by Mathematica */
   readdata("demo\cdqw_wfc.dat",data1);
   readdata("demo\cdqw_wfc.da2",data2);
   readdata("demo\cdqw_wfc.da3",data3);
   readdata("demo\cdqw_wfc.da4",data4);
   readdata("demo\cdqw_wfc.da5",data5);
   /* drawing turned out too big so reduce the overall size */
   scale(0.7,0.65);
   gsave();
   axes_box(8,15,-185,0.0,185,188.36);
   set(LINECOLOUR,BLACK);
   translate(-1.25,22.5);
   /* even solns to cdqw */
   /*   norm = 1.0/0.00009134 * 60.0; */
   set(YMULT,656886.);
   set(YSHIFT,9.13863);
   drawdata(data1);
   /*   norm = 1/.0002838 * 60.0;  */
   set(YMULT,211416.);
   set(YSHIFT,35.8858);
   drawdata(data1,0,3);
   /*   norm = 1/0.008109 * 60.0;*/
   set(YMULT,7400.);
   set(YSHIFT,77.0532);
   drawdata(data3);
   /* call subroutine to draw quantum well. Also sets coordinate scale to match*/
   s_qw_box();
   /* energy eigenvalues */
   moveto(-190,9.13863);
   lineto(190,9.13863);
   moveto(-190,35.8858);
   lineto(190,35.8858);
   moveto(-190,77.0532);
   lineto(190,77.0532);
   stroke();
   text(205,9.13863,"E_0_");
   text(205,35.8858,"E_2_");
   text(205,77.0532,"E_4_");
   text(0,-15,"a) CDQW, even solutions",CENTER);
   grestore();
   /* odd solns to cdqw */
   set(LINECOLOUR,INVIS);
   gsave();
   axes_box(8,15,-185,0.0,185,188.36);
   set(LINECOLOUR,BLACK);
   translate(11,22.5);
```

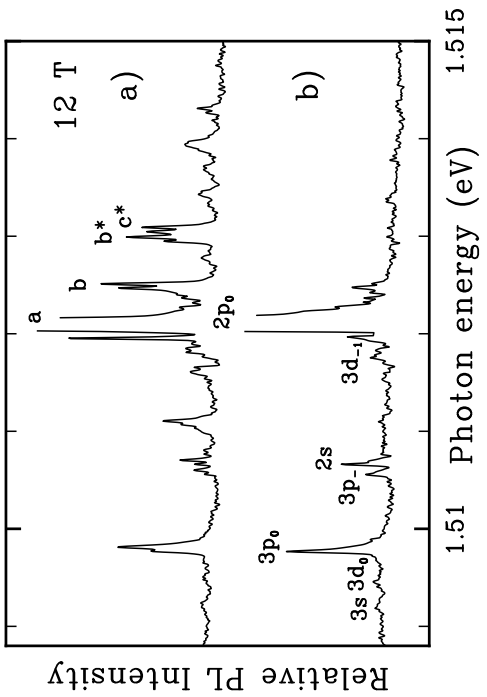
```
set(YMULT,656886.);
set(YSHIFT,9.14186);
drawdata(data1,0,2);
set(YMULT,211416.);
set(YSHIFT,35.9177);
drawdata(data1,0,4);
set(YMULT,7400.);
set(YSHIFT,77.4394);
drawdata(data3,0,2);
s_qw_box();
/* energy eigenvalues */
moveto(-190,9.14186);
lineto(190,9.14186);
moveto(-190,35.9177);
lineto(190,35.9177);
moveto(-190,77.4394);
lineto(190,77.4394);
stroke();
text(-227,9.14186,"E_1_");
text(-227,35.9177,"E_3_");
text(-227,77.4394,"E_5_");
text(217.,53.,"107meV",CENTER);
text(0,120,"9 nm",CENTER);
text(0,-15,"b) CDQW, odd solutions",CENTER);
grestore();
/*asymmetric coupled wells */
set(LINECOLOUR,INVIS);
gsave();
axes_box(8,15,-185,0.0,185,188.36);
set(LINECOLOUR,BLACK);
translate(5.5,2.5);
set(YMULT,1000.);
set(YSHIFT,90.23);
set(XSHIFT,-95.);
drawdata(data4,0,2);
a_qw_box();
moveto(-195,90.23);
lineto(190,90.23);
stroke();
text(215,90.23,"E_2_",CENTER);
text(280,90.23,"Resonance");
text(-270.,104.,"188 meV",CENTER);
grestore();
set(LINECOLOUR,INVIS);
gsave();
axes_box(8,15,-185,0.0,185,188.36);
set(LINECOLOUR,BLACK);
translate(5.5,2.5);
set(XSHIFT,-95.);
set(YMULT,-80.0);
set(YSHIFT,10.29);
drawdata(data5);
```

```
   set(YMULT,-80.0);
   set(YSHIFT,40.81);
   drawdata(data5,0,2);
   set(YMULT,-275.0);
   set(YSHIFT,154.16);
   drawdata(data5,0,3);
   set(YSHIFT,116.97);
   set(YMULT,500.0);
   a_qw_box();
   moveto(-190,10.29);
   lineto(190,10.29);
   moveto(-190,40.81);
   lineto(190,40.81);
   moveto(-190,154.16);
   lineto(190,154.16);
   stroke();
   text(215,10.29,"E_0_",CENTER);
   text(215,40.81,"E_1_",CENTER);
   text(215,154.16,"E_5_",CENTER);
   text(0,200,"c) Asymmetric CDQW",CENTER);
   grestore();
   set(LINECOLOUR,BLACK);
   set(FONTWIDTH,1.);
   set(FONTASPECT,2.5);
   moveto(20.91,36.9);
   arrowto(20.0,36.9);
   moveto(20.91,36.9);
   arrowto(22.0,36.9);
   moveto(25.5,33.02);
   arrowto(25.5,36.53);
   moveto(25.5,31.79);
   arrowto(25.5,27.95);
   moveto(9.9,16.90);
   arrowto(9.9,23.02);
   moveto(9.9,15.70);
   arrowto(9.9,7.93);
   moveto(21.4,15.14);
   arrowto(20.8,15.14);
   stroke();
   }

int s_qw_box()
   {
   set(LINEWIDTH,0.06);
   cmatch();
   /* draw quantum well box */
   moveto(-225.,107.27);
   lineto(-145.,107.27);
   lineto(-145.,0.);
   lineto(-45.,0.);
   lineto(-45.,107.27);
   lineto(45.,107.27);
```

```
   lineto(45.,0.);
   lineto(145.,0.);
   lineto(145.,107.27);
   lineto(225.,107.27);
   stroke();
   set(LINEWIDTH,0.02);
   }

int a_qw_box()
   {
   set(LINEWIDTH,0.06);
   cmatch();
   /* draw quantum well box */
   moveto(-225.,188.36);
   lineto(-145.,188.36);
   lineto(-145.,0.);
   lineto(-45.,0.);
   lineto(-45.,188.36);
   lineto(45.,188.36);
   /* 188.36-107.27  */
   lineto(45.,81.09);
   lineto(145.,81.09);
   lineto(145.,188.36);
   lineto(225,188.36);
   stroke();
   set(LINEWIDTH,0.02);
   }
```

5.8 landscap.spt

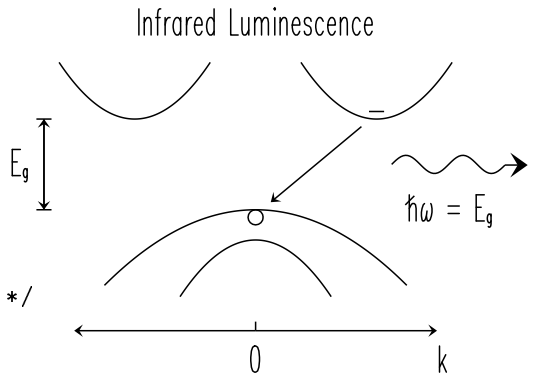
```
#include <splot.h>
double *data1,*data2;
main()
   {

   /* select landscape orientation */
   set(PAGE_ROT,ON);
   set(LINEWIDTH,0.09);
   axes_box(20,14,1.5088,0,1.515,1.03e8,3,3);

   set(TICKLENGTH,0.5);
   tickmarks(XAXES,1.510,1.515);

   set(LINEWIDTH,0.06);
   set(FONTWIDTH,0.8);
   /* use short squat letters for labels */
   set(FONTASPECT,1.1);
   set(FONT,COMPLEX);
   ticklabel(LOWER,1.510,1.515);
   set(FONTWIDTH,1.0);
   label(LEFT,"Relative PL Intensity");
   text(9.00,1.90,"Photon energy (eV)");
   text(20.00,15.00,"12 T");
   text(21.00,13.00,"a)");
   text(21.00,7.00,"b)");
   set(FONTWIDTH,0.70);
   text(13.47,9.84,"2p_0_");
   text(5.70,8.35,"3p_0_");
   text(8.80,6.46,"2s");
   text(4.80,5.40,"3d_0_");
   text(3.82,5.30,"3s");
   text(7.65,5.70,"3p_-_");
   text(11.65,5.60,"3d_-1_");
   set(FONTWIDTH,0.75);
   text(13.65,16.10,"a");
   text(14.70,14.60,"b");
   text(16.25,13.75,"b*");
   text(16.80,13.10,"c*");

   set(LINEWIDTH,0.05);
   set(TICKLENGTH,0.35);
   tickmarks(XAXES,1.509,1.511,1.512,1.513,1.514,1.516);

   /* constrain plot to axes box */
   set(AXESCLIP,ON);

   /* change x axis units to eV */
   set(XMULT,0.1239842e-3);

   /* plot first spectrum */
```

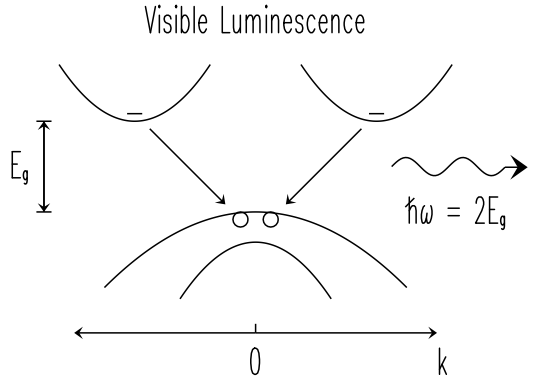


```
set(XSHIFT,0);
set(YSHIFT,47e6);
set(YMULT,300);
readdata("demo\fe21m23a.dat",data1);
set(YRANGE,0,99e6);
drawdata(data1,0,1);

/* plot second spectrum shifted up */
set(YSHIFT, 7e6);
set(YMULT,150);
readdata("demo\fe22v31a.dat",data2);
set(YRANGE,0,45e6);
drawdata(data2,0,1);
}
```

5.9 intcalc.spt

```
#include <splot.h>

/* This example file illustrates some of the internal math capabilities */
/* In this case  x,y data points for several parabolas are calculated */
/* and then drawn. Also the photon uses the sine function to draw a wave. */
/* For the photon the data is not stored in an array, rather the calculated */
/* values are added to offsets given as parameters to the photon subroutine */

/* make space for the to be generated data */
double pdat[101][2];

main()
   {
   double x0,y0;

   x0 = 10;
   y0 = 5.5;

   /* draw parabolas at x0,y0  to represent bands*/
   siband(x0,y0);

   /* draw circles to represent holes in the valence band */
   arc(x0 - 0.5,y0 - 0.25,0.25,0,360);
   stroke();
   arc(x0 + 0.5,y0 - 0.25,0.25,0,360);
   stroke();

   /* draw electrons in the conduction band */
   moveto(x0 + 4.25,y0 + 3.25);
   rlineto(-0.5,0);
   moveto(x0 - 4.25,y0 + 3.25);
   rlineto(0.5,0);
   stroke();

   /* draw arrows to indicate optical transitions */
   moveto(x0 + 3.5,y0 + 2.75);
   arrowto(x0 + 1,y0 + 0.25);
   moveto(x0 - 3.5,y0 + 2.75);
   arrowto(x0 - 1,y0 + 0.25);

   /* label the photon */
   text(x0 + 5, y0,"h!w! = 2E_g_");
   /* but bar through h */
   moveto(x0 + 4.95,y0 + 0.16);
   rlineto(0.3,0.3);
   stroke();

   /* draw bands again but higher up */
   x0 = 10;
   y0 = 19;
```

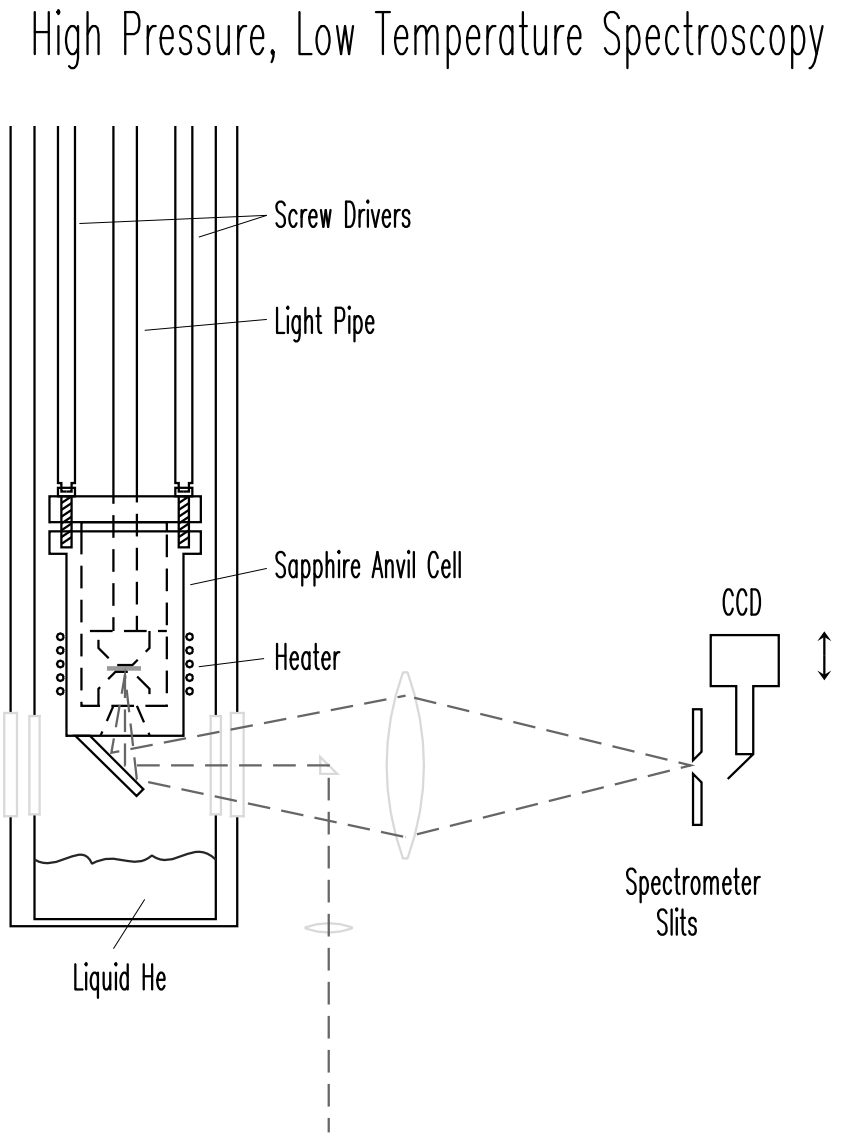




```
   siband(x0,y0);

   /* draw single hole */
   arc(x0,y0 - 0.25,0.25,0,360);
   stroke();

   /* draw single electron */
   moveto(x0 + 4.25,y0 + 3.25);
   rlineto(-0.5,0);
   stroke();
   moveto(x0 + 3.5,y0 + 2.75);
   arrowto(x0 + 0.5,y0 + 0.25);

   /* label photon */
   text(x0 + 5, y0,"h!w! = E_g_");
   moveto(x0 + 4.95,y0 + 0.16);
   rlineto(0.3,0.3);
   stroke();

   /* add titles */
   text(10.0,25.15,"Infrared Luminescence",CENTER);
   text(10.0,11.8,"Visible Luminescence",CENTER);
   }

int siband(double x0,y0)
   {

   /* draws strained SiGe band structure */
   /* as a set of parabolas */
   parab(x0,y0,0.1,-0.1);
   parab(x0,y0 - 1.0,0.05,-0.3);
   parab(x0 - 4.0,y0 + 3.0,0.05,0.3);
   parab(x0 + 4.0,y0 + 3.0,0.05,0.3);

   /* draw axes */
   moveto(x0,y0 - 4.0);
   rlineto(0,0.3);
   moveto(x0,y0 - 4.0);
   rarrowto(6,0);
   moveto(x0,y0 - 4.0);
   rarrowto(-6,0);
   stroke();
   moveto(x0 - 7, y0);
   rarrowto(0,3);
   rarrowto(0,-3);
   moveto(x0 - 7.25,y0);
   rlineto(0.5,0);
   moveto(x0 - 7.25,y0 + 3);
   rlineto(0.5,0);
   stroke();

   /* draw a photon */
```

```
   photon(x0 + 4.5,y0 + 1.5,0.3,0.3);

   /* a k vector labels */
   text(x0,y0 - 5,"0",CENTER);
   text(x0 + 6,y0 - 5,"k");
   text(x0 - 7.5,y0 + 1.5,"E_g_",RIGHT);
   }

int parab(double x0,double y0,double dx,double a)
   {
   int i;
   double x;
   /* draws a parabola by generating */
   /* a data array first */
   for (i = 0; i <= 50;i++)
      {
      x = i * dx;
      pdat[50 + i][1] = a * x * x + y0;
      pdat[50 - i][1] = a * x * x + y0;
      pdat[50 + i][0] = x0 + x;
      pdat[50 - i][0] = x0 - x;
      }
   moveto(pdat[0][0],pdat[0][1]);
   for (i = 0;i <= 100;i++)
      {
      lineto(pdat[i][0],pdat[i][1]);
      }
   stroke();
   }

int photon(double x0,double y0,double xscale,double yscale)
   {
   double x,y;
   int i;

   /* draw a photon using the sine function */
   moveto(x0,y0);
   for (i = 0;i < 50;i++)
      {
      x = i * 0.255;
      y = sin(x);
      lineto(x * xscale + x0,y * yscale + y0);
      }
   set(FONTMULT,2);
   rarrowto(2.5 * xscale,0);
   set(FONTMULT,0.5);
   stroke();
   }
```

5.10 linedraw.spt




```
/* This file illustrates how to generate a line drawing of apparatus */
/* It makes heavy use of gsave() and grestore() and axes transformations*/
/* to draw elements repeatedly in different locations and/or orientations */
/* This is just one way to do this. */

#include <splot.h>

main()
   {
   /* it turned out too big in the end so scale it back some */
   scale(0.75,0.75);
```

```
/* draw Dewar */
tail();
gsave();
/* draw a slightly bigger tail around the first */
scale(1.25,1.05);
translate(-1.05,-0.5);
tail();
grestore();

/* draw lenses */
newpath();
set(LINECOLOUR,BROWN);
gsave();
translate(12.79,10.85);
rotate(90);
scale(2,2);
lens();
grestore();
gsave();
translate(11.09,6.21);
scale(0.5,0.5);
lens();
grestore();

/* draw the prism */
gsave();
translate(10.84,10.6);
prism();
grestore();

/* draw helium level */
set(LINECOLOUR,BLUE);
moveto(2.44,8.08);
curveto(3.13,7.66,3.82,8.68,4.13,7.96);
curveto(4.89,8.38,5.32,7.72,5.89,8.20);
curveto(6.58,7.72,7.08,8.74,7.77,8.08);
stroke();

/* draw 45 degree fold mirror */
set(LINECOLOUR,BLACK);
moveto(4.07,11.72);
lineto(5.64,10.15,5.44,9.95,3.64,11.72);
closepath();

/* draw diamnond anvil cell */
moveto(3.38,11.72);
lineto(6.82,11.72,6.82,17.07,7.33,17.07,7.33,17.73,2.88,17.73,
       2.88,17.07,3.38,17.07);
closepath();
moveto(3.82,17.72);
lineto(3.82,18,6.33,18,6.33,17.72);
stroke();
```

```
/* draw plate and screws */
moveto(2.88,18.0);
lineto(7.33,18.0,7.33,18.76,2.88,18.76);
closepath();
stroke();
gsave();
translate(3.38,18.76);
screw(1.5);
translate(3.45,0);
screw(1.5);
grestore();

/* draw heater */
gsave();
translate(7.0,13.03);
heater();
translate(-3.8,0);
heater();
grestore();

/* draw slits */
gsave();
translate(21.8,11.0);
slit();
grestore();
gsave();
scale(1,-1);
translate(21.8,-10.6);
slit();
grestore();

/* draw periscope */
moveto(22.82,10.45);
lineto(23.57,11.18);
rlineto(0,2, 0.75,0, 0,1.5, -2,0, 0,-1.5, 0.75,0, 0,-2, 0.5,0);
stroke();

/* draw arrows */
moveto(25.66,14.07);
arrowto(25.66,14.79);
moveto(25.66,14.07);
arrowto(25.66,13.35);
stroke();

/* draw screw drivers and light pipe */
moveto(4.76,18.76);
lineto(4.76,30.0);
moveto(5.45,18.76);
lineto(5.45,30.0);
stroke();
gsave();
```

```
translate(3.38,18.9);
driver();
translate(3.45,0);
driver();
grestore();

/* white out some of the lines at */
/* the top that turned out too long */
white_box(1.17,29.67,8.61,31.20);
newpath();

/* draw copper gasket */
gsave();
set(LINECOLOUR,GREEN);
set(LINEWIDTH,0.10);
moveto(4.57,13.7);
lineto(5.57,13.7);
stroke();
grestore();

/* draw dashed lines in diamond cell */
set(LINESTYLE,0.5,0.25);
moveto(3.82,17.72);
lineto(3.82,12.60,6.33,12.60,6.33,17.72);
moveto(3.82,14.8);
lineto(6.33,14.8);
moveto(4.76,14.8);
lineto(4.76,18.76);
moveto(5.45,14.8);
lineto(5.45,18.76);
moveto(4.76,12.6);
lineto(4.38,11.72);
moveto(5.45,12.6);
lineto(5.83,11.72);
stroke();

/* draw the sapphire anvils */
gsave();
translate(4.32,12.6);
anvil();
translate(1.5,2.2);
rotate(180);
anvil();
grestore();
stroke();

/* draw light paths */
set(LINECOLOUR,MAGENTA);
moveto(5.1,13.57);
lineto(5.1,10.85,11.09,10.85,11.09,0.06);
moveto(13.35,12.9);
lineto(4.7,11.22,5.1,13.51,5.45,10.43,13.35,8.74);
```

```
   moveto(13.35,12.9);
   lineto(21.73,10.85,13.35,8.74);
   stroke();

   /* add comments */
   set(LINECOLOUR,BLACK);
   set(LINESTYLE,0);
   set(FONTWIDTH,1.0);
   text(2.33,32.29,"High Pressure, Low Temperature Spectroscopy");
   set(FONTWIDTH,0.6);
   text(3.57,4.58,"Liquid He");
   text(22.65,15.60,"CCD");
   text(19.81,7.39,"Spectrometer");
              text("   Slits");
   text(9.5,16.7,"Sapphire Anvil Cell");
   text(9.5,14,"Heater");
   text(9.5,23.88,"Light Pipe");
   text(9.5,27,"Screw Drivers");

   /* draw lines to labelled parts */
   set(LINEWIDTH,0.01);
   moveto(4.76,5.46);
   lineto(5.68,6.91);
   moveto(9.19,13.99);
   lineto(7.27,13.75);
   moveto(9.27,16.64);
   lineto(7.02,16.16);
   moveto(9.27,23.96);
   lineto(5.68,23.64);
   moveto(9.27,27.02);
   lineto(7.27,26.38);
   moveto(9.27,27.02);
   lineto(3.76,26.78);
   stroke();
   }

int anvil()
   {
   /* draw a sapphire anvil */
   newpath();
   moveto(0,0);
   rlineto(0,0.5,0.5,0.5,0.5,0,0.5,-0.5,0,-0.5);
   stroke();
   }

int screw(double len)
   {
   int i,j;
   /* draw a screw using a loop for */
   /* the threads */
   newpath();
   moveto(0,0);
```

```
   rlineto(0.25,0,0,0.25,-0.5,0,0,-0.25);
   closepath();
   moveto(0.15,0);
   rlineto(0,-len,-0.3,0,0,len);
   moveto(-0.15,0);
   j = (int) (len / 0.2);
   for (i=0;i < j;i++)
      {
      rmoveto(0.3,0);
      rlineto(-0.30,-0.20);
      }
   stroke();
   }

int heater()
   {
   int i;
   /* draw a heater using a loop */
   /* for the windings */
   double y = 0;
   newpath();
   moveto(0,0);
   for (i=0;i < 5;i++)
      {
      moveto(0.1,y);
      arcn(0,y,0.1,360,0);
      y = y + 0.4;
      }
   stroke();
   }

int slit()
   {
   /* draw a spectrometer slit */
   newpath();
   moveto(0,0);
   rlineto(0,1.5,0.25,0,0,-1.25);
   closepath();
   stroke();
   }

int driver()
   {
   /* draw long screw drivers */
   newpath();
   moveto(0,0);
   rlineto(0.15,0,0,0.25,0.1,0,0,10.8);
   moveto(0,0);
   rlineto(-0.15,0,0,0.25,-0.1,0,0,10.8);
   stroke();
   }
```

```
int white_box(double x1,double y1,double x2,double y2)
   {
   /* draw a white filled box for */
   /* white out purposes. The parametrs */
   /* passed in are the box corners */
   newpath();
   box(x1,y1,x2,y2);
   gsave();
   set(LINECOLOUR,WHITE);
   fill();
   grestore();
   }

int lens()
   {
   /* draw a spherical lens using arcs */
   newpath();
   arcn(0,-4,4,110,70);
   arcn(0,3.45,4,290,250);
   closepath();
   stroke();
   }

int prism()
   {
   /* draw a prism */
   newpath();
   moveto(0,0.5);
   rlineto(0,-0.5,0.5,0);
   closepath();
   stroke();
   }

int tail()
   {
   /* draw the dewar tail */
   set(LINECOLOUR,BLACK);
   moveto(2.44,30);
   lineto(2.44,6.33,7.77,6.33,7.77,30);
   stroke();
   /* draw windows */
   set(LINECOLOUR,BROWN);
   white_box(2.29,12.30,2.59,9.41);
   stroke();
   white_box(7.62,12.30,7.92,9.41);
   stroke();
   }
```