Manual for WitsPhysicsReport

Warren A. Carlson

February 6, 2017

Abstract

This document demonstrates usage of the WitsPhysicsReport package for undergraduate report LaTeX-markup at the School of Physics at the University of the Witwatersrand, Johannesburg.

Contents

1 Introduction **1**
- 1.1 Basic Package . . . 2
- 1.2 Prerequisites . . . 2
- 1.3 Basic Usage . . . 3
- 1.4 Configuring for Use . . . 3

2 Pictures and Graphics **4**

3 Plotting Data **5**
- 3.1 Known Issues . . . 5
- 3.2 Simple Plots . . . 5
- 3.3 Decorated Plots . . . 7
  - 3.3.1 Plot Decorations . . . 7
  - 3.3.2 Simple Decorated Plots . . . 8
- 3.4 Advanced Data Plotting . . . 9
  - 3.4.1 Plotting Trend Lines . . . 9
  - 3.4.2 Plotting Collections . . . 10

4 Tabulating Data **11**
- 4.1 Tabulating Free Form Data . . . 12
- 4.2 Tabulating Regression Data . . . 12

1 Introduction

The School of Physics has created a LaTeX template designed to be used for undergraduate reports. This template is under development and newer versions will be made available from time to time. These updates will include newer and more refined styling and functions. The template and this document describing the template were inspired by and borrow from the School of Mechanical, Industrial & Aeronautical Engineering report and thesis Template at the University of the Wiwatersrand, written by Randal T. Paton.

The WitsPhysicsReport document class provides a template which specifies page and margin formatting, paragraph spacing, font sizes and types, colouring and element numbering. Document details may be set using a collection of convenience macros. Once these document details are set, a standard report title page is generated, with supporting report preamble contents such as tables of contents, lists of figures and tables and so on. Reports generated with WitsPhysicsReport are subject to the following limitations,

1. Documents must comprise fewer than 16 pages including the title page, preamble pages and bibliography. This is a hard page limit. Any document exceeding this limit will fail to compile and the system will throw a package error.

2. Additional packages may be used except when inclusion of these packages causes any conflict with those packages already included by WitsPhysicsReport .

The WitsPhysicsReportUtil utility package can generate high-quality data plots and tabulations, and provides consistant font size, and figure and tabular appearance and formatting. Simple data visualisation, presentation and formatting are systematised for documents generated with the WitsPhysicsReportUtil . This package makes use of pgfplots and gnuplot methods and uses human readable, comma separated value data files to quickly and easily generate documents. This package provides macros for data plotting, smoothing and labelling, and tabulation. Convenience macros are also available to add content to reports. Reports generated with WitsPhysicsReportUtil are subject to the following limitations,

1. Data files passed to convenience macros must be properly formatted comma separated value files which satisfy the internal structure requirements of the macro which uses them.

2. Convenience macro parameter lists must be completed in full to function correctly.

3. A unique, whitespace free, character string value must be provided to any macro specifying a reference label as an input.

Macros provided by this package are for convenience purposes and users are encouraged to make use of the pgfplots package used by WitsPhysicsReport to generate document contests. The document class and utility package may be used together or separately for document preparation.

1.1 Basic Package

The School of Physics report template is setup for the preparation of short reports and assignments. With the exception of some required packages, this template has been prepared to be mostly software system agnostic. This system, although using the LaTeX document markup and generation system, need not be used with any specific LaTeX development environents or other external software packages or system. Simple, human readable, comma separated value files and images with common file formats are used to generate document graphical contents.

The WitsPhysicsReport contains the WitsPhysicsReport document class template and WitsPhysicsReportUtil utility package and university logo. The template contained within the package comprises a document class file `WitsPhysicsReport.cls` and a university logo file. The utility package is contained in the style file `WitsPhysicsReportUtil.sty`. These files should be kept in the same direcory as the main document `.tex` file is located

1.2 Prerequisites

WitsPhysicsReport document class requires only a working LaTeX distribution to be installed. The utility package WitsPhysicsReportUtil requires a working LaTeX distribution with the pgfplots package and gnuplot to be installed. The pgfplots package is used to assign plotting parameters. The actual plot generation handled by gnuplot which must be installed separately. gnuplot may be installed in various ways on different systems (installation instructions for pgfplots and gnuplot are outside the scope of this document). pgfplots is available as part of the LaTeX document generation system, and at http://pgfplots.sourceforge.net. gnuplot is available at http://www.gnuplot.info.

To ensure that LaTeX is able to make use of gnuplot during compilation, the gnuplot command line executable must be available in the system path. This can be set at installation time on most systems. Additionally, the `-shell-escape` option must be added to the LaTeX compilation command. This con be done within a development environment if one chooses to use one, or on the command-line interface for command line execution. An example of such a command-line execution command is shown below,

```
pdflatex -shell-escacpe document.tex
```

where `pdflatex` is the standard PDF LaTeX compilation command and `document.tex` is the TeX/LaTeX document to compile.

1.3 Basic Usage

The WitsPhysicsReport package requires no explicit installation to the TEX/LATEX system, and may be used by simply copying the contents of th package into the main `.tex` file to be processed. Then, adding the directive

```
\documentclass{WitsPhysicsReport}
```

to the document preamble will make the WitsPhysicsReport package available to the system during document compilation. The WitsPhysicsReportUtil package is made available for use by adding the directive

```
\usepackage{WitsPhysicsReportUtil}
```

to the document preamble.

1.4 Configuring for Use

Once the WitsPhysicsReport is assigned as the `documentclass` is the preamble of the `.tex` document, specific details of the document may be set using a collection of helper macros. Title page details must be set before the `\maketitle` command is called to generate the title page correctly. Details of the commands used to assign title page details are given below.

`\Title{`⟨*title*⟩`}`

Assign the document title as ⟨ *title* ⟩. If the document title is not set then the following error message is replaced for the document title, PLEASE ENTER A REPORT TITILE.

`\CourseCode{`⟨*code*⟩`}`

Assign the course code as ⟨*code*⟩. If the course code is not set then the following error message is replaced for the course code, PLEASE ENTER THE COURSE CODE.

`\CourseName{`⟨*name*⟩`}`

Assign the course name as ⟨*name*⟩. If the course name is not set then the following error message is replaced for the course name, PLEASE ENTER THE COURSE NAME.

`\StudentName{`⟨*name*⟩`}`

Assign the student name as ⟨*name*⟩. If the student name is not set then the following error message is replaced for the student name, PLEASE ENTER YOUR NAME.

`\StudentNumber{`⟨*number*⟩`}`

Assign the student number as ⟨*number*⟩. If the student number is not set then the following error message is replaced for the student number, PLEASE ENTER YOUR STUDENT NUMBER.

`\StudentEmail{`⟨*address*⟩`}`

Assign the email address as ⟨*address*⟩. If the email address is not set then the following error message is replaced for the email address, PLEASE ENTER YOUR EMAIL ADDRESS.

The preamble be generated by adding the directive `\maketitle` to the main `.tex` document immediatly following the `\begin{document}` directive as in the example below.

```
\documentclass{WitsPhysicsReport}
\usepackage{WitsPhysicsReportUtil}

\Title{X-Ray Diffraction}
\CourseCode{PHYS3006}
\CourseName{Experimental Physics III}
\StudentName{Joe Nobody}
\StudentNumber{1234567890}
\StudentEmail{fake.em@il.com}

\begin{document}
```

```
\maketitle

...

\bibliography{bibliography}
\end{document}
```

2 Pictures and Graphics

It is common to add images, diagrams and photographs to documents. These elements can be added using the convenience macros provided by WitsPhysicsReportUtil package. Correct usage of these macros is described below.

All image files processed by this package should be placed in a directory named `pictures` in the directory containing the main `.tex` file to be processed. The following file type are acceptable `.jpg`, `.png`, `.pdf`, `.eps`. Each picture added this way may be referenced within the LaTeX document using `\ref{fig:`⟨*refer*⟩`}` where ⟨*refer*⟩ is the, *whitespace free*, reference name supplied to the macro. The `fig:` prefix is added to automatically to distinguish the referenced figure from other object references.

`\PrintImage{`⟨*image*⟩`}{`⟨*refer*⟩`}{`⟨*cap*⟩`}`

> Add the picture in the image file ⟨*image*⟩ to the document. This picture is referenced by ⟨*refer*⟩ and has caption ⟨*cap*⟩. Images included this way have a fixed width to height ratio, where the maximum height of any image is set to 56% of the textwidth.

The following code example generates the output shown in Figure 1.

```
\PrintImage{rubber-duck.jpg}
           {rubber-duck}
           {This is \textsc{not} a duck.}
```

In this example, the listed figure is cross-referenced using `\ref{fig:rubber-duck}`.

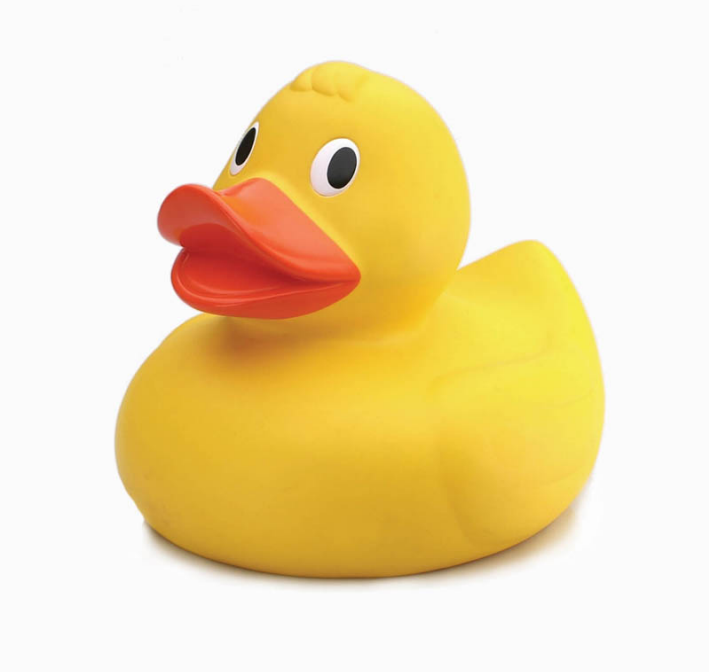

Figure 1: This is NOT a duck.

3 Plotting Data

The WitsPhysicsReport package contains a collection of two dimensional data visualisation convenience macros . Each macro has collection of parameters to customise the plot. Plot parameters are determined from the data provided to the system. In each case, data file names are passed to the desired macro and the macro generates the appropriate graph corresponding to the maximum and minimum range and domain of the data provided. All data files processed by this package should be placed in a directory named `data` in the directory containing the main `.tex` file to be processed.

Plot creation is then driven by PGFPLOTS and GNUPLOT to produce high quality graphics for inclusion on LaTeX documents. These plot are gerenated from data files with a specific format. Single data points are stored as tuples of values, separated by commas. A single tuple is written to a single line, such that that each row in the data file contains exactly one data point. The first column in such a data file contains the first co-ordinate of each tuple, the second column contains the second co-ordinate and so on, such that the i-th column contains a list of the i-th co-ordinate values. In this way, the i-th co-ordinate of the j-th data point is contained in the i-th column on the j-th row of the data file. An example of such a data file is presented below

```
38.52, 0.1088,
44.76, 0.1450,
65.14, 0.2898,
78.26, 0.3983,
82.47, 0.4345,
99.11, 0.5792,
```

Each figure generated using these macro may be referenced within the LaTeX document using the `\ref{fig:`⟨*refer*⟩`}` and where ⟨*refer*⟩ is the, *whitespace free*, reference name supplied to the macro. The `fig:` prefix is added to automatically to distinguish the referenced figure from other object references. Tabulars generated this way may be similarly referenced using `tab:`⟨*refer*⟩.

Compilation time may be shortened by adding the `\tikzexternalize` command macro to the preamble of the main `.tex` file. This will instruct the system to generate figures as separate files which may then be included without regeneration. Output files generated with `\tikzexternalize` enabled take the name ⟨*refer*⟩. WitsPhysicsReport will place these externally generated files in a subdirectory called `tikz` in the directory containing the main `.tex` file.

3.1 Known Issues

The system will output files to the `tikz` subdirectory when `\tikzexternalize` is used. This subdirectory is located in directory where the main `.tex` file is found. While some systems will automatically created this directory, some do not. If this directory is missing, the system will generate a collection of errors stating that some files could not be written or read. Creating this subdirectory will resolve this issue.

The LaTeX compiler may throw an error the first time it is run with the `\tikzexternalize` directive. This occurs because the external execution directive evaluates all externall commands together. Some of these commands require the ouput files generated by other commands. When these ouput files are not found, the system throws an error. After the first run, many of these files will have been written. Running the compiler again will resolve this issue.

3.2 Simple Plots

The most commonly used set of convenience macros for data visualisation are presented below. Each macro has a collection of parameters that must be set. In each case only a single data set is plotted. Special axis scale modifier versions are available for each of the functions which allow for plotting on logarithmic scale axis. Each such function is distinguished by suffix. The `SemiLogScaleX` suffix denotes logarithmic x-axis scale. The `SemiLogScaleY` suffix denotes logarithmic y-axis scale. The `LogLogScale` suffix denotes logarithmic x- and y-axis scales.

`\PlotPoints{`⟨*dat*⟩`}{`⟨*refer*⟩`}{`⟨*cap*⟩`}{`⟨*xlabel*⟩`}{`⟨*xunit*⟩`}{`⟨*ylabel*⟩`}{`⟨*yunit*⟩`}`

> Generate a scatter plot of x vs y data in the data file ⟨*dat*⟩. The plot is referenced by ⟨*refer*⟩ and hascaption ⟨*cap*⟩. The parameters ⟨*xlabel*⟩, ⟨*xunit*⟩, ⟨*ylabel*⟩ and ⟨*yunit*⟩ are the x-axis label and unit specifications, and y-axis label and unit specifications, respectively.

Modified: \PlotPointsLogLogScale , \PlotPointsSemiLogScaleX , \PlotPointsSemiLogScaleY .

\PlotLines{⟨*dat*⟩}{⟨*refer*⟩}{⟨*cap*⟩}{⟨*xlabel*⟩}{⟨*xunit*⟩}{⟨*ylabel*⟩}{⟨*yunit*⟩}

Generate a line plot of x vs y data in the data file ⟨*dat*⟩. The plot is referenced by ⟨*refer*⟩ and has caption ⟨*cap*⟩. The parameters ⟨*xlabel*⟩, ⟨*xunit*⟩, ⟨*ylabel*⟩ and ⟨*yunit*⟩ are the x-axis label and unit specifications, and y-axis label and unit specifications, respectively.

Modified: \PlotLinesLogLogScale , \PlotLinesSemiLogScaleX , \PlotLinesSemiLogScaleY

\PlotSteps{⟨*dat*⟩}{⟨*refer*⟩}{⟨*cap*⟩}{⟨*xlabel*⟩}{⟨*xunit*⟩}{⟨*ylabel*⟩}{⟨*yunit*⟩}

Generate a step plot histogram plot of pre-binned x vs y data in the data file ⟨*dat*⟩. The plot is referenced by ⟨*refer*⟩ and hascaption ⟨*cap*⟩. The parameters ⟨*xlabel*⟩, ⟨*xunit*⟩, ⟨*ylabel*⟩ and ⟨*yunit*⟩ are the x-axis label and unit specifications, and y-axis label and unit specifications, respectively.

Modified: \PlotStepsLogLogScale , \PlotStepsSemiLogScaleX , \PlotStepsSemiLogScaleY .

\PlotBoxes{⟨*dat*⟩}{⟨*refer*⟩}{⟨*cap*⟩}{⟨*xlabel*⟩}{⟨*xunit*⟩}{⟨*ylabel*⟩}{⟨*yunit*⟩}

Generate a bar plot histogram plot of pre-binned of x vs y data in the data file ⟨*dat*⟩. The plot is referenced by ⟨*refer*⟩ and hascaption ⟨*cap*⟩. The parameters ⟨*xlabel*⟩, ⟨*xunit*⟩, ⟨*ylabel*⟩ and ⟨*yunit*⟩ are the x-axis label and unit specifications, and y-axis label and unit specifications, respectively.

Modified: \PlotBoxesLogLogScale , \PlotBoxesSemiLogScaleX , \PlotBoxesSemiLogScaleY .

The following code example generates the output shown in Figure 2.

```
\PlotPoints{powder.csv}
           {plain-points-plot}
           {Scatterplot of data set.}
           {Data}
           {$2 \theta$}
           {\si{\degree}}
           {Intensity}
           {counts}
```

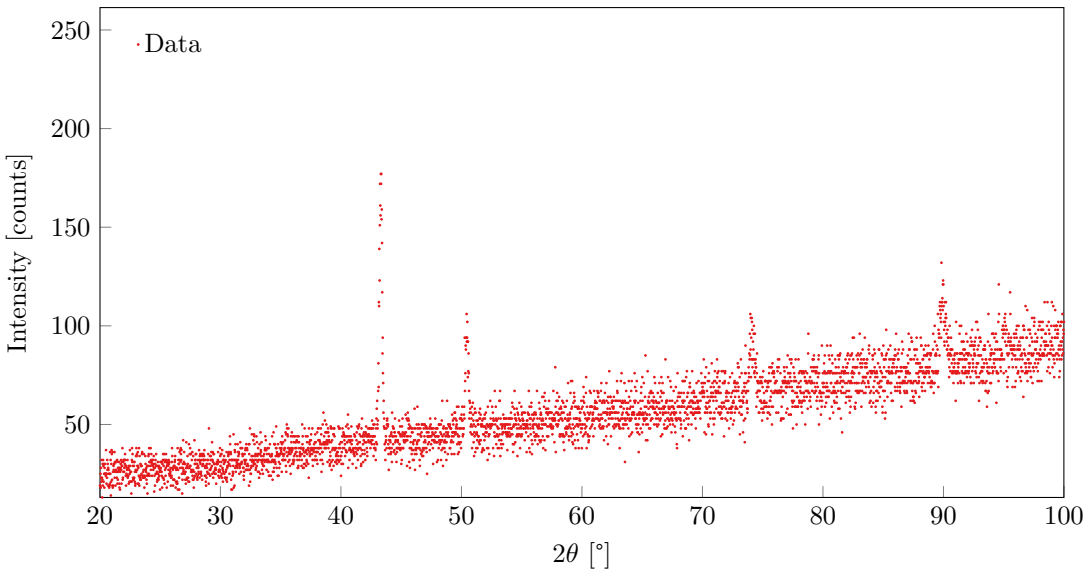

Figure 2: Scatter plot of data set.

The following code example generates the output shown in Figure 3.

```
\PlotSteps{normal-random.csv}
          {plain-boxes-plot}
          {Histogram box plot of data set.}
          {Data}
          {$x$}
          {\si{\meter}}
          {Count}
          {counts}
```

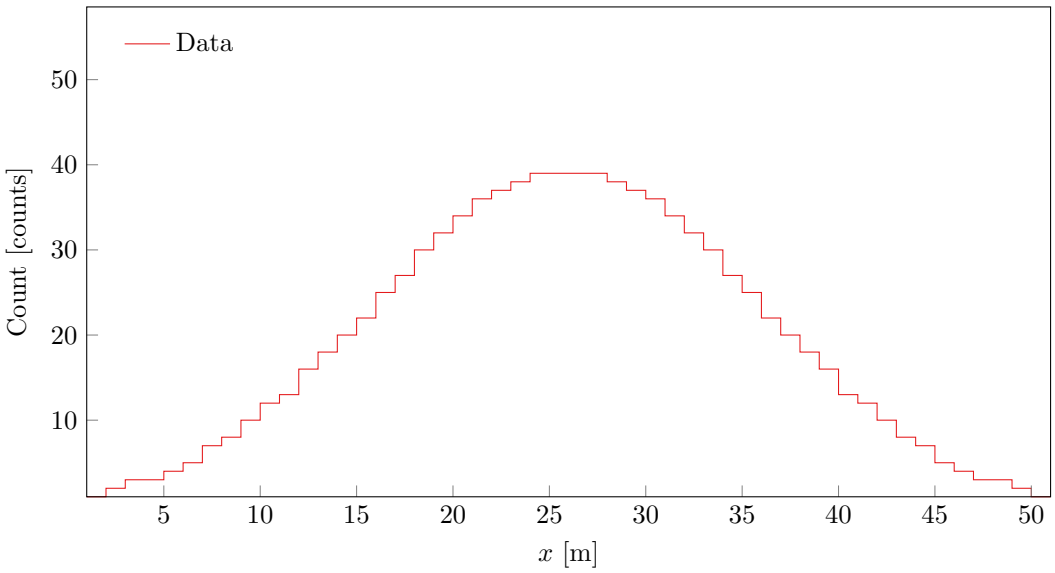

Figure 3: Histogram box plot of data set.

The following code example generates the output shown in Figure **??**.

```
\PlotSteps{random.csv}
          {plain-steps-plot}
          {Histogram step plot of data set.}
          {Data}
          {$x$}
          {\si{\meter}}
          {Count}
          {counts}
```

3.3 Decorated Plots

At times it is useful to add docorations to plots. The WitsPhysicsReport package makes available a collection of convinience macros to add labels and decorations to plots.

3.3.1 Plot Decorations

Plot decorations are added using a collection of macros which add labelled horizontal and vertical markers and labels. A synopsis of thes macros is given below.

`\AddLabel{⟨label⟩}{⟨x⟩}{⟨y⟩}`

Adds a vertically alligned marker pin with vertically alligned text ⟨*label*⟩ at the plot co-ordinate (⟨*x*⟩,⟨*y*⟩).

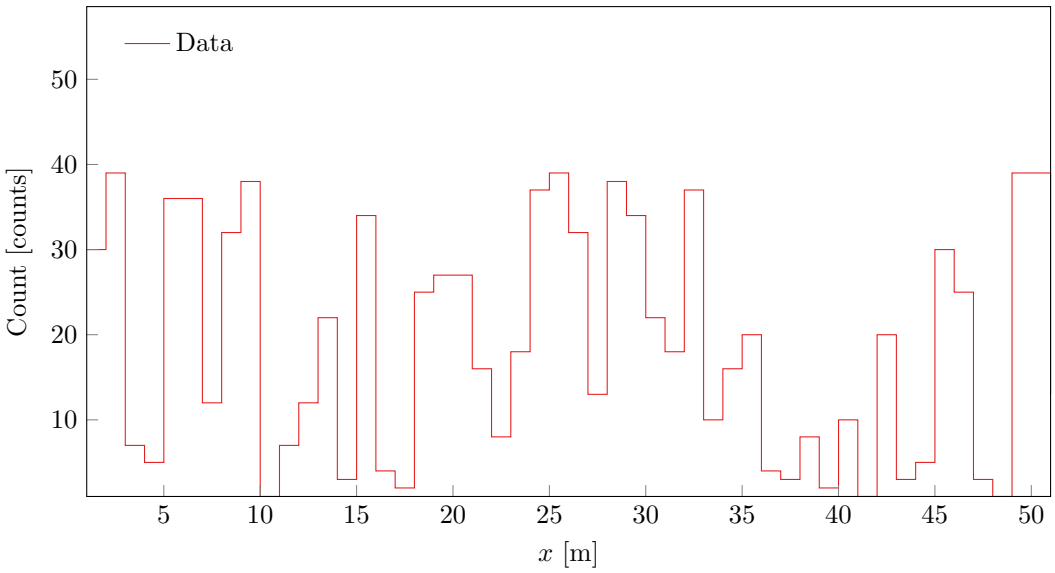

Figure 4: Histogram step plot of data set.

`\AddVerticalMarker{`⟨*label*⟩`}{`⟨*x*⟩`}`

Adds a dashed line parallel to the y-axis at the x-axis co-ordinate ⟨*x*⟩ and a x-axis label ⟨*label*⟩.

`\AddHorizontalMarker{`⟨*label*⟩`}{`⟨*y*⟩`}`

Adds a dashed line parallel to the x-axis at the y-axis co-ordinate ⟨*y*⟩ and a y-axis label ⟨*label*⟩.

3.3.2 Simple Decorated Plots

A collection of convenience macros which add decorations to commonly used plotting macros is presented below. These macros can be added as parameters to the given macros as a list. This list should comprise `\AddLabel`, `\AddHorizontalMarker` and `\AddVerticalMarker` macros.

`\PlotLabelledLine{`⟨*dat*⟩`}{`⟨*refer*⟩`}{`⟨*cap*⟩`}{`⟨*xlabel*⟩`}{`⟨*xunit*⟩`}{`⟨*ylabel*⟩`}{`⟨*yunit*⟩`}{`⟨*markers*⟩`}`

Generate a line plot of x vs y data in the data file ⟨*dat*⟩. The plot is referenced by ⟨*refer*⟩ and caption ⟨*cap*⟩. The parameters ⟨*xlabel*⟩, ⟨*xunit*⟩, ⟨*ylabel*⟩ and ⟨*yunit*⟩ are the x-axis label and unit specifications, and y-axis label and unit specifications, respectively. Additional decorations may be added to this plot in the ⟨*markers*⟩ list parameter.

Modified: `\PlotLabelledLineLogLogScale` , `\PlotLabelledLineSemiLogScaleX` , `\PlotLabelledLineSemiLogScaleY` .

`\PlotSmoothLabelledPoints{`⟨*dat*⟩`}{`⟨*refer*⟩`}{`⟨*cap*⟩`}{`⟨*xlabel*⟩`}{`⟨*xunit*⟩`}{`⟨*ylabel*⟩`}{`⟨*yunit*⟩`}{`⟨*markers*⟩`}`

Generate a scatter plot of x vs y data in the data file ⟨*dat*⟩. The plot is referenced by ⟨*refer*⟩ and caption ⟨*cap*⟩. The parameters ⟨*xlabel*⟩, ⟨*xunit*⟩, ⟨*ylabel*⟩ and ⟨*yunit*⟩ are the x-axis label and unit specifications, and y-axis label and unit specifications, respectively. Additional decorations may be added to this plot in the ⟨*markers*⟩ list parameter.

Modified: `\PlotSmoothLabelledPointsLogLogScale` , `\PlotSmoothLabelledPointsSemiLogScaleX` , `\PlotSmoothLabelledPointsSemiLogScaleY` .

The following code example generates the output shown in Figure 5.

```
\PlotSmoothLabelledPointsSemiLogScaleY{narrow-spectrum-30kV-no-filter.csv}
                                      {narrow-spectrum-30kV-no-filter}
                                      {Narrow Spectrum 30kV no Filter}
                                      {Data}
                                      {$2 \theta$}
                                      {\si{\degree}}
```

```
                                {Intensity}
                                {counts}
                                {
                                    \AddVerticalMarker{8.4}
                                    \AddHorizontalMarker{22}
                                }
```

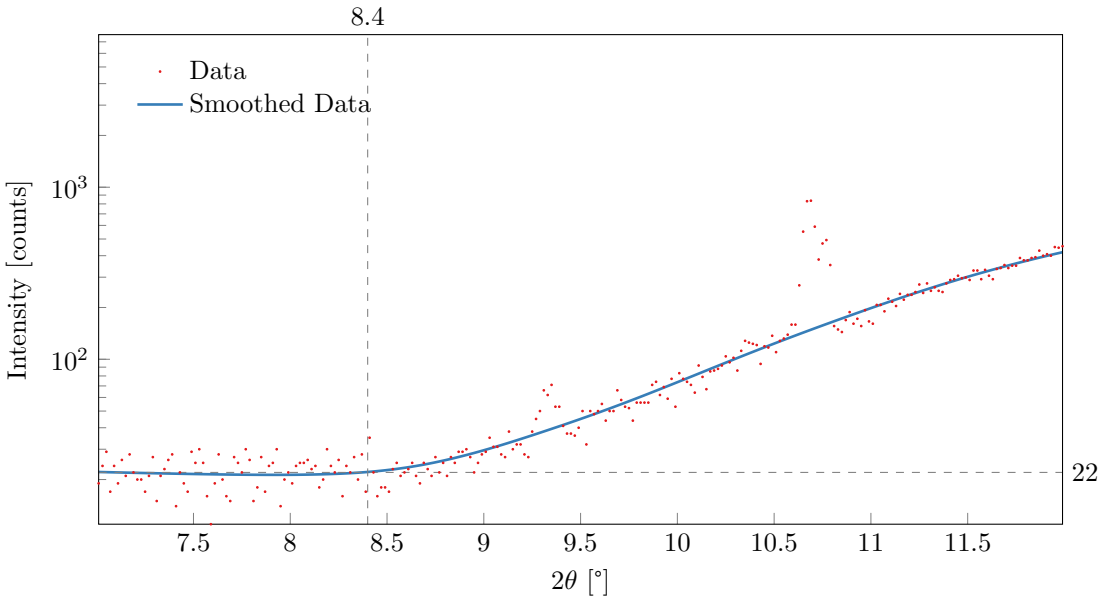

Figure 5: Narrow Spectrum 30kV no Filter

3.4 Advanced Data Plotting

More advanced plotting macros are described below.

3.4.1 Plotting Trend Lines

Data plots with trend lines may be generated using two linear regression fitting convenience macros listed below.

`\PlotOneParameterLinearRegression{⟨dat⟩}{⟨refer⟩}{⟨cap⟩}{⟨xlabel⟩}{⟨xunit⟩}{⟨ylabel⟩}{⟨yunit⟩}`

Generate a one parameter linear regression line plot of x vs y data in the data file ⟨*dat*⟩. The plot is referenced by ⟨*refer*⟩ and caption ⟨*cap*⟩. The parameters ⟨*xlabel*⟩, ⟨*xunit*⟩, ⟨*ylabel*⟩ and ⟨*yunit*⟩ are the x-axis label and unit specifications, and y-axis label and unit specifications, respectively.

`\PlotTwoParameterLinearRegression{⟨dat⟩}{⟨refer⟩}{⟨cap⟩}{⟨xlabel⟩}{⟨xunit⟩}{⟨ylabel⟩}{⟨yunit⟩}`

Generate a one parameter linear regression line plot of x vs y data in the data file ⟨*dat*⟩. The plot is referenced by ⟨*refer*⟩ and caption ⟨*cap*⟩. The parameters ⟨*xlabel*⟩, ⟨*xunit*⟩, ⟨*ylabel*⟩ and ⟨*yunit*⟩ are the x-axis label and unit specifications, and y-axis label and unit specifications, respectively.

The following code example generates the output shown in Figure 6.

```
\PlotTwoParameterLinearRegression{linear.csv}
                                 {two-param-linear-regression}
                                 {Two parameter linear regression data plot.}
                                 {Data}
                                 {$\lambda^{-1}_{min}$}
                                 {\si{\per\angstrom}}
                                 {Potential}
                                 {\si{\kilo\volt}}
```

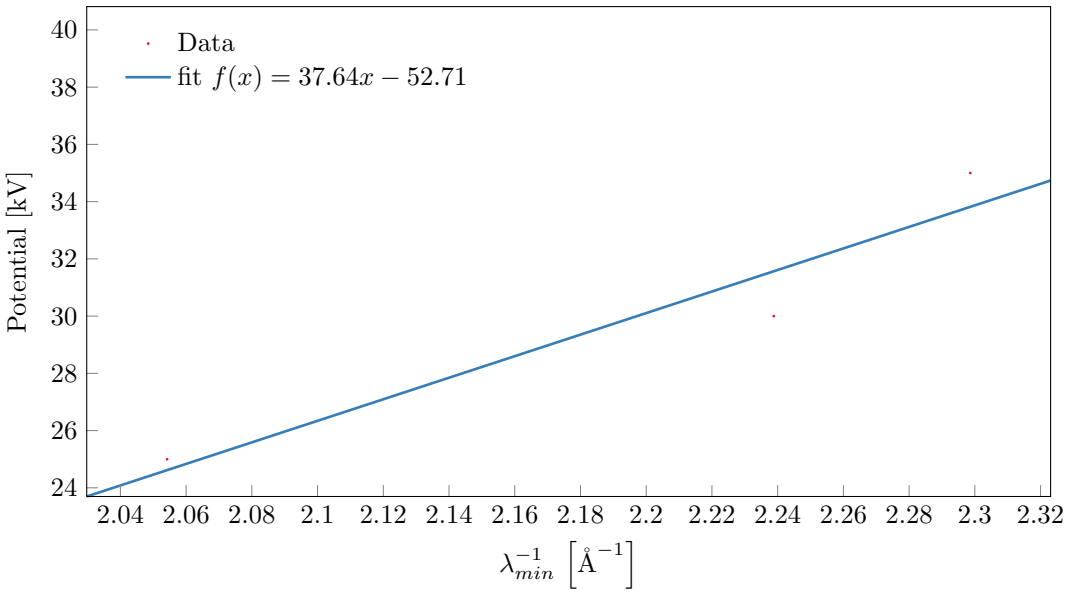

Figure 6: Two parameter linear regression data plot.

3.4.2 Plotting Collections

Elaborate collections of data and decorations may be plotted using the generic `\PlotCollection` convenience macro. Special axis scale modifier versions are available for each of the functions which allow for plotting on logarithmic scale axis and are identified by the suffixes `SemiLogScaleX`, `SemiLogScaleY`, and `LogLogScale`, as above. Collections may be generated by assembling a list of components which are described in detail later in this note.

`\PlotCollection{`⟨*refer*⟩`}{`⟨*cap*⟩`}{`⟨*xlabel*⟩`}{`⟨*xunit*⟩`}{`⟨*ylabel*⟩`}{`⟨*yunit*⟩`}{`⟨*list*⟩`}`

> Plot a collection of graphs on a single axis. The plot is referenced by ⟨*refer*⟩ and caption ⟨*cap*⟩. The parameters ⟨*xlabel*⟩, ⟨*xunit*⟩, ⟨*ylabel*⟩ and ⟨*yunit*⟩ are the x-axis label and units, and y-axis label and unit specifications, respectively. Decorations may be added to this plot in the ⟨*list*⟩ list parameter. This list should comprise `\AddLabel`, `\AddHorizontalMarker` and `\AddVerticalMarker` plot decorations, and `\AddPoints`, `\AddLines`, `\AddOneParameterLinearRegression`, `\AddTwoParameterLinearRegression`, and `\AddThreePointMeanThresholdSmoothing`.
>
> *Modified:* `\PlotCollectionLogLogScale` , `\PlotCollectionSemiLogScaleX` , `\PlotCollectionSemiLogScaleY` .

Below is a list of macros that may be used to add additional graphs to plots. Each of these macros must be used within a `tikzpicture` environment. This environment is available within the set of convenience macros described above. Each of these may be added to the a list of macros supplied to the ⟨*list*⟩ parameter described above.

`\AddPoints{`⟨*dat*⟩`}{`⟨*legend*⟩`}`

> Adds a scatter plot of x vs y data in the data file ⟨*dat*⟩ and has legend entry ⟨*legend*⟩. Data in this file are stored as two comma separated columns where the first and second columns comprise x-axis and y-axis data, respectively.

`\AddLines{`⟨*dat*⟩`}{`⟨*legend*⟩`}`

> Adds a line plot of x vs y data in the data file ⟨*dat*⟩ and has legend entry ⟨*legend*⟩. Data in this file are stored as two comma separated columns where the first and second columns comprise x-axis and y-axis data, respectively.

`\AddThreePointMeanThresholdSmoothing{⟨dat⟩}{⟨limit⟩}`

Adds a three point shifting window with approximated cubic spline curve of x vs y data in the data file ⟨*dat*⟩ where the threshold limit of the shifting window is ⟨*limit*⟩. Data in this file are stored as two comma separated columns where the first and second columns comprise x-axis and y-axis data, respectively.

`\AddOneParameterLinearRegression{⟨dat⟩}`

Add a one parameter linear regression line plot of x vs y data in the datafile ⟨*dat*⟩. Data in this file are stored as two comma separated columns where the first and second columns comprise x-axis and y-axis data, respectively. The legend entry is generated automatically as the fit function

$$y(x) = ax.$$

This tool generates two output files in the `tikz` subdirectory. The values, errors and fractional errors for the parameter a are written to the file ⟨*dat*⟩`.one.parameter.regression`. A log of the plotting data generation is written to the file ⟨*dat*⟩`.one.parameter.regression.log`.

`\AddTwoParameterLinearRegression{⟨dat⟩}`

Add a two parameter linear regression line plot of x vs y data in the datafile ⟨*dat*⟩. Data in this file are stored as two comma separated columns where the first and second columns comprise x-axis and y-axis data, respectively. The legend entry is generated automatically as the fit function

$$y(x) = ax + b.$$

This tool generates two output files in the `tikz` subdirectory. The values, errors and fractional errors for parameters a and b are written to the file ⟨*dat*⟩`.two.parameter.regression`. A log of the plotting data generation is written to the file ⟨*dat*⟩`two.parameter.regression.log`.

The following code example generates the output shown in Figure 7.

```
\PlotCollectionSemiLogScaleY{collection-log-scale-y}
                            {Decorated Data Plot Collection}
                            {$2 \theta$}
                            {\si{\degree}}
                            {Intensity}
                            {counts}
    {
    \AddLines{wide-spectrum-30kV-unknown-filter.csv}{Unknown}
    \AddPoints{wide-spectrum-30kV-Nickle-filter.csv}{Ni}
    \AddThreePointMeanThresholdSmoothing{wide-spectrum-30kV-Nickle-filter.csv}{Ni}{80}
    \AddVerticalMarker{32.81}
    \AddHorizontalMarker{30}
    \AddHorizontalMarker{50}
    \AddLabel{ $9.35$}{ 9.35}{ 59}
    \AddLabel{$23.77$}{23.77}{121}
    \AddLabel{$24.49$}{24.49}{151}
    \AddLabel{$26.63$}{26.63}{306}
    \AddLabel{$29.51$}{29.51}{342}
    }
```

4 Tabulating Data

The WitsPhysicsReport provides two ways to include tabulated data. In each case, data are read from a comma separated values file and presented in a tabular form.

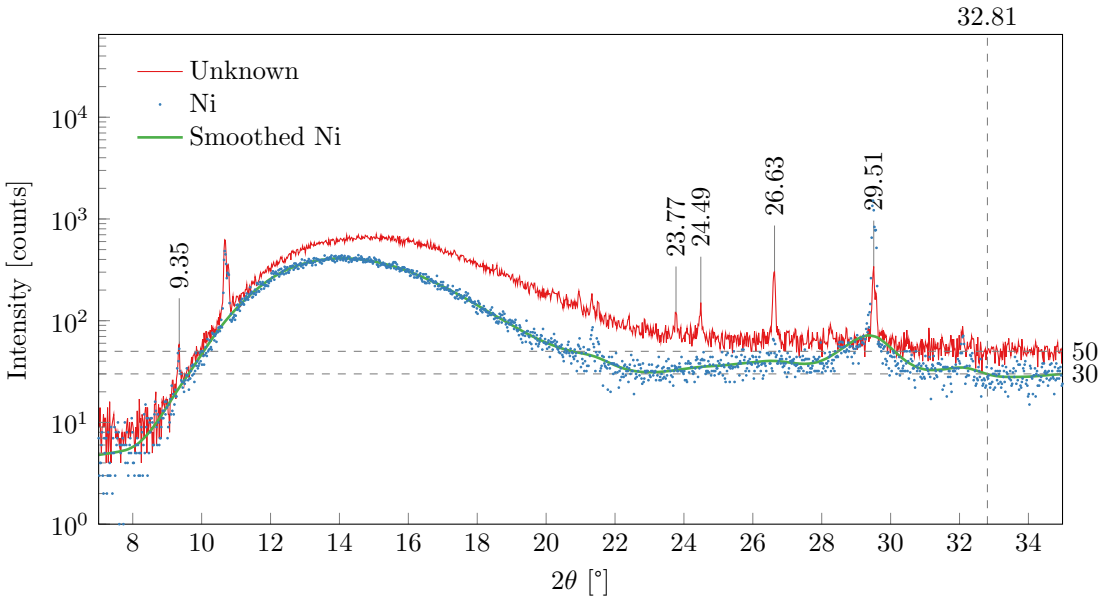

Figure 7: Decorated Data Plot Collection

4.1 Tabulating Free Form Data

Semi-automatic tabulation of free form data may be achieved using the following convenience macros.

`\TabulateData{⟨dat⟩}{⟨refer⟩}{⟨cap⟩}{⟨cols⟩}`

> Data in the multirow and multicolumn ⟨ *datafile* ⟩. The table is referenced by ⟨*reference*⟩ and has caption ⟨*cap*⟩. Column headers with LaTeX markup are added via the ampersand separated string in ⟨*cols*⟩.

Consider the data file `free-form-table.csv` with the following free form contents.

```
1, 38.52, 0.1088,  2.9974,  3, 111
2, 44.76, 0.1450,  3.9936,  4, 200
3, 65.14, 0.2898,  7.9834,  8, 220
4, 78.26, 0.3983, 10.9715, 11, 311
5, 82.47, 0.4345, 11.9691, 12, 222
6, 99.11, 0.5792, 15.9550, 16, 400
```

The following code example tabulates the contents of `free-form-table` with custom LaTeX markup column header. The output of this shown in Table 1.

```
\TabulateData{free-form-table.csv}
             {free-form-table}
             {Free-Form data table}
             {
                 Peak &
                 $2 \theta ( \si{\degree} )$ &
                 $\sin{\theta}^{2}$ &
                 $\frac{\sin{\theta}^{2}}{A}$ &
                 $h^{2} + k^{2} + l^{2}$ &
                 $(h k l)$
             }
```

4.2 Tabulating Regression Data

Regression plot data may be automatically tabulated. Data presented in the tabulated regression plot parameter tables are read from data files generated by the regression plot macros. These files take the names provided to the ⟨*refer*⟩ parameter in the regression plot macros with an additional suffix which identifies the corresponding regression model.

Table 1: Free-Form data table

Peak	$2\theta(^\circ)$	$\sin\theta^2$	$\frac{\sin\theta^2}{A}$	$h^2+k^2+l^2$	(hkl)
1	38.52	0.109	2.997	3	111
2	44.76	0.145	3.994	4	200
3	65.14	0.29	7.983	8	220
4	78.26	0.398	10.972	11	311
5	82.47	0.435	11.969	12	222
6	99.11	0.579	15.955	16	400

`\TabulateRegressionData{`⟨*dat*⟩`}{`⟨*refer*⟩`}{`⟨*cap*⟩`}`

Tabulate linear regression data generated by `\PlotOneParameterLinearRegression`, `\PlotTwoParameterLinearRegression`, `\AddOneParameterLinearRegression` or `\AddTwoParameterLinearRegression` commands for the data file ⟨*dat*⟩. The table is referenced by ⟨*refer*⟩ and has caption ⟨*cap*⟩.

The data file provided as ⟨*dat*⟩ in the `\PlotTwoParameterLinearRegression` macro parameter list is `linear.csv`. This macro generates parameter output file with name `linear.csv.two.parameter.regression` which is then passed as the ⟨*dat*⟩ parameter in `\TabulateRegressionData` macro in the code example. This code example generates the Table 2.

```
\TabulateRegressionData{linear.csv.two.parameter.regression}
                       {two-param-linear-regression}
                       {Two parameter linear regression data.}
```

Table 2: Two parameter linear regression data.

Parameter	Value	Error	Fractional Error
a	3.764×10^{1}	1.110×10^{1}	2.948×10^{-1}
b	-5.271×10^{1}	2.441×10^{1}	-4.630×10^{-1}