Introduction
The software was originally created as a real-time visualisation program for monitoring sulphur dioxide emissions from volcanoes. However, with several different projects under way in our research group, all requiring a different set of data manipulation tools but similar user interfaces, it soon became clear that a more generic solution would be extremely useful; AvoPlot was designed to fill this role.

Essentially, AvoPlot aims to achieve two things:

• To provide a simple and intuitive graphical user interface for the matplotlib plotting library, with tools for importing common data formats.
• To provide an extensible platform that researchers can easily customise to suit their specific data visualisation and analysis requirements.

Although there are numerous open-source plotting programs already available, none (that the author is aware of) provide the same functionality as AvoPlot. Spreadsheet programs such as OpenOffice Calc [1] and Gnumeric [2] cannot load arbitrary data types, and also lack good data navigation tools (panning, zooming etc.) for their plots. The plot viewer built into matplotlib [3], provides good data navigation tools, but has very few graphical facilities for editing plot parameters. The Veusz [4] program incorporates most of the features of AvoPlot. It has a comprehensive graphical interface for creating and editing plots, as well as providing a plug-in interface to allow users to load arbitrary data types and define new analysis functions. However, Veusz implements its own plotting backend which is unlikely to be as familiar to developers as matplotlib and has a smaller community providing support. Furthermore, the scope for writing complex plug-ins is far more limited than in AvoPlot, which allows developers the freedom to use any wxPython [5] constructs when writing their plug-in interfaces.

At the time of writing, AvoPlot is still at an early stage of development. However, it is already being used as a visualisation and analysis tool by several members of the Cambridge Volcanology Group [6] and the Mount Erebus Volcano Observatory [7]. Plug-ins for working with differential optical absorption spectroscopy (DOAS) data and Fourier transform infrared (FTIR) spectra are currently under development and will be published in due course.

Implementation/architecture
AvoPlot is written in pure Python [8]. Graphical interface elements are implemented using wxPython and plots are created and displayed using matplotlib and its wxagg backend.

The software consists of two major components; the user interface (UI) and the plug-in system. The UI provides a generic graphical interface to much of matplotlib’s functionality, allowing users to change plot parameters (such as line styles, colours etc.) without needing to write any code.
It also allows users to interact with any plug-ins that have been installed. An overview of the UI is shown in Figure 1.

The plug-in system is intended to give developers a simple way both to import their specific data into the UI, and to provide tools for manipulating them. Details of how to write plug-ins are given in the AvoPlot documentation which can be found in the “doc” folder of the source distribution, or online as part of the software archive. Furthermore, example plug-ins can be found in the “examples” folder of the source distribution.

AvoPlot comes with a simple file import plug-in which will be installed automatically. Currently, this can only load text files containing columns of numerical data. However, it will be extended in future releases to support other file types such as CSV, Excel spreadsheets etc. Sample data for creating a plot with this plug-in are provided in the “spectrum_data.txt” file of the “examples” folder of the source distribution and step-by-step instructions for its use can be found in the AvoPlot documentation.

**Quality control**
Functional testing has been carried out on both Linux and Microsoft Windows (XP and 7) using Python 2.7. AvoPlot behaved as expected in both cases.

**Support**
Support is provided for users through an email list. There is a separate email list for developers which aims to provide support for people writing plug-ins. Both lists can be found at the code repository site.
(3) Reuse potential

As a standalone program, AvoPlot provides an easy to use, generic plotting tool capable of producing publication-quality figures. Although the current feature set is relatively limited, future development will focus on exposing more of matplotlib's functionality through AvoPlot's UI. This will give users access to matplotlib's extensive capabilities without requiring them to write Python scripts. In this respect, the program has a high reuse potential “as is” for a wide spectrum of end-users regardless of coding experience.

However, the main reuse potential for AvoPlot comes from its plug-in based architecture. With just a few lines of relatively simple code, researchers can write a plug-in that will plot their specific data set (which may be in a format difficult to import with mainstream plotting programs) in the AvoPlot UI. They can then take advantage of a feature-rich plot editing environment to customise their plot without having to edit or run scripts. Furthermore, AvoPlot plug-ins are distributed using Python’s distutils system [9], making it simple for other researchers (who may have no programming experience) to install plug-ins written by their peers.

In addition to data import, AvoPlot plug-ins also allow researchers to easily integrate their own data manipulation and analysis tools into the AvoPlot UI, enabling them to create a graphical data processing software without having to write extensive UI code of their own. The aim is to encourage researchers to make the analysis tools which they create user-friendly and accessible for others, who may not be programmers.

Acknowledgements

The author wishes to thank his supervisor Prof. Clive Oppenheimer for his support of the project.

References

1. Apache OpenOffice Calc. Available at http://www.openoffice.org/product/calc.html [Last accessed 19 July 2013]
2. GNOME Office / Gnumeric - Welcome to Gnumeric!. Available at https://projects.gnome.org/gnumeric/ [Last accessed 19 July 2013]
3. Hunter, J D 2007 Matplotlib: A 2D graphics environment. Computing in Science & Engineering , 9(3): 90–95. DOI: http://dx.doi.org/10.1109/MCSE.2007.55
4. Veusz - A Scientific Plotting Package. Available at http://home.gna.org/veusz/ [Last accessed 19 July 2013]
5. Dunn, R 2003 wxpython: wxwindows for python. Available at http://www.wxpython.org
6. CVG – Cambridge Volcanology Group | Undertaking pioneering research into the processes, monitoring, and impacts of volcanism. Available at http://www.volcano.group.cam.ac.uk/ [Last accessed 18 July 2013]
7. MEVO | Mount Erebus Volcano Observatory. Available at http://erebus.nmt.edu/ [Last accessed 18 July 2013]
8. Python. Python Software Foundation. Available at www.python.org/
9. Distributing Python Modules — Python v2.7.5 documentation. Available at http://docs.python.org/2/distutils/ [Last accessed 21 July 2013]