

CU-QMW-TN-0016
Date: 15 October 2001

Issue: 0
Rev. : 0
Page: i

Plotting in QSAS2

Steve Schwartz
Queen Mary, University of London

Document Status Sheet			
1. Document Title: Plotting in QSAS2			
2. Document Reference Number: CU-QMW-TN-0016			
3. Issue	4. Revision	5. Date	6. Reason for Change
0	0	12 October 2001	First draft; for discussion

Contents

1	Introduction	1
1.1	Background	1
1.2	Plot Basics	1
2	Elements of the Plot	1
3	Examples	2
4	Plottable Objects	2
4.1	1-D	2
4.2	1.5-D	3
4.3	2-D	3
4.3.1	Examples	3
4.3.2	Plot Types	3
4.4	3-D	3
4.4.1	Examples	3
4.4.2	Plot Types	4
4.4.3	Plot Options	4
4.4.4	Axes and Bin Limits	4
5	GUI Aspects	4
6	Issues and Questions	5

1 Introduction

1.1 Background

The plot user interface, and underlying capabilities, in QSAS v1 are effective but limited in terms of layout, flexibility, and repeated operations (options, etc.). Additionally, there is no provision for array handling, spectrograms, and similar higher dimensional objects now available in QSAS via QDOS2. Save and Restore is vital here to meet user requirements. Finally, the move to Qt provides the right time to re-think and re-code the user interface. While other GUIs in QSAS v1 can be translated easily from Motif to Qt (even if they are eventually re-designed from scratch later) the plot interface is sufficiently complicated that it would waste too much time to replicate the QSAS v1 PUI in Qt. Hence the PUI is an item of major importance.

This document is an attempt to record my thoughts and ponderings on the subject, rather than lay down rigid requirements. I won't repeat at length much of the functionality of QSAS1, save to say that it is essentially all needed in QSAS2 (multi-traces, stacks, autoscaling, time-interval control, interactive value return, printing, ...).

1.2 Plot Basics

I use the PGPLOT terminology throughout here

View Surface is the physical page, x-window, etc.

Non-dimensional Device Coordinates ("ndc") run from (0,0) at bottom left to (1,1) at top right of the plottable surface. This surface is usually the "page" but it is also possible to divide the physical page into sub-pages. I think in this case, the ndc's run from 0 to 1 within a single sub-page. Most QSAS work is probably best done by handling the page sub-division ourselves in any case.

Viewport is the rectangle within the View Surface which contains a single "graph" - i.e., the rectangle which corresponds to the plot axes.

World Coordinates ("wc") are the values in real physical units (e.g., nanoTeslas for a magnetic field).

Window is the rectangle in world coordinates which corresponds to the Viewport. Plots get clipped at the boundaries.

2 Elements of the Plot

At the basic level, I think to the user the plot consists of the following roughly hierarchical collection:

Page or Pages which are defined in terms of

1. Size and/or aspect ratio of plottable surface
2. Margins
3. Title
4. User identification/date, filename, ...

(Groups of) Panels defined by and/or need

1. ($xndc$, $yndc$) of viewport (assumed rectangular, with option SQUARE)
2. Groups of Panels (may) share common:
 - (a) Layout Options
 - Arrange in columns, rows, matrix, loose, overplot
 - Touching or gap (x and/or y) width
 - Equal $\Delta xndc$ and/or $\Delta yndc$
 - (b) Scale Options
 - Common Limits (x and/or y)
 - Common World Coordinate Ranges (Δxwc and/or Δywc (and/or Δzwc for spectrogram plots))
 - (c) Plot Objects (e.g., same object(s) will be used in all panels, such as a vector (though different panels may plot different components of this common object)).
 - (d) Panel "Aesthetics" (see below)
 - (e) Plot Options
 - Line style, colour, thickness, symbols for individual traces
 - Colour Table
3. Plottable Objects (see below): all of same type within a single panel. To combine types use Overplot.

There is an issue here as to how and when to propagate these properties to members within the group and how to override and/or lock individual deviations to fine tune one group member. Are they defined at the group level and propagated down or are they/could they be defined for one member and then shared?

4. Panel Aesthetics

- Title (font type, size, placement)
- Box/Axis line thickness and colour
- Major tick length, in/out/grid, tick interval, decimal/degrees
- Minor tick ...
- Draw zero line
- Axis title position (l,r,t,b), font type and size, and orientation
- Axis numeric labels position (l,r,t,b), font type and size, and orientation
- Linear or Log scale
- Legend for traces: placement, contents, opaque, framed, font
- Colour bar for Spectrograms: placement, title, font
- Colour bar ...: numeric labels position, font, orientation

Plottable Objects are described more fully in a separate section, but are one of

1-D i.e., just like an axis

1.5-D Extra tick labels

2-D e.g., XY or TY

3-D e.g., $Z(X,Y)$ or $Z(T,Y)$

Panels themselves?

These objects can be derived from other objects, so that, for example, the x -component of a vector time series is a 2-D plot object derived from an array.

3 Examples

I'll try to generate some examples, including:

- Samples from existing QSAS plots: these go a long way to meeting the average users' needs.
- Hodograms: side-by-side square plots with the same world coordinate ranges (though not necessarily values), as done in my minimum variance plots.
- Vector Multi-panel. Can be made within QSAS v1 but gets tedious. Needs common Δywc range and common panel height. Also for several traces, common colour scheme, etc. for all panels.

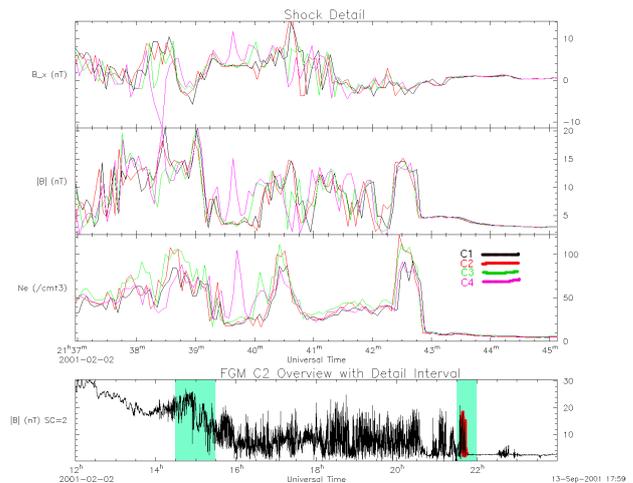


Figure 1: Example: Classic stack. If members included different components of a vector, the vertical scales should represent the same range, e.g., $0 \rightarrow 20\text{nT}$, $-10 \rightarrow +10\text{nT}$, etc. Note the shaded regions corresponding to 1-D time interval objects. If used in the stack, such shading should be possible for all panels. This could be achieved by repeating it (automagically) as a “trace” in each one or overplotting it (perhaps even overplotting it with the entire group).

- Energy-time spectrograms
- Energy-time spectrogram with overplot of a line trace (e.g., spacecraft potential).

4 Plottable Objects

4.1 1-D

A 1-D plottable object is anything which only requires a single set of numbers to plot. I can think of at least two examples:

Events are (arrays of) single values

Ranges are (arrays of) Δ objects which can be used to delineate events or ranges of interest.

The most typical examples are times or time intervals corresponding to events. These can be plotted/shown as

- Shading (colour and intensity options; opacity?)
- Lines (vertical if along horizontal time axes) with options for line style and colour.

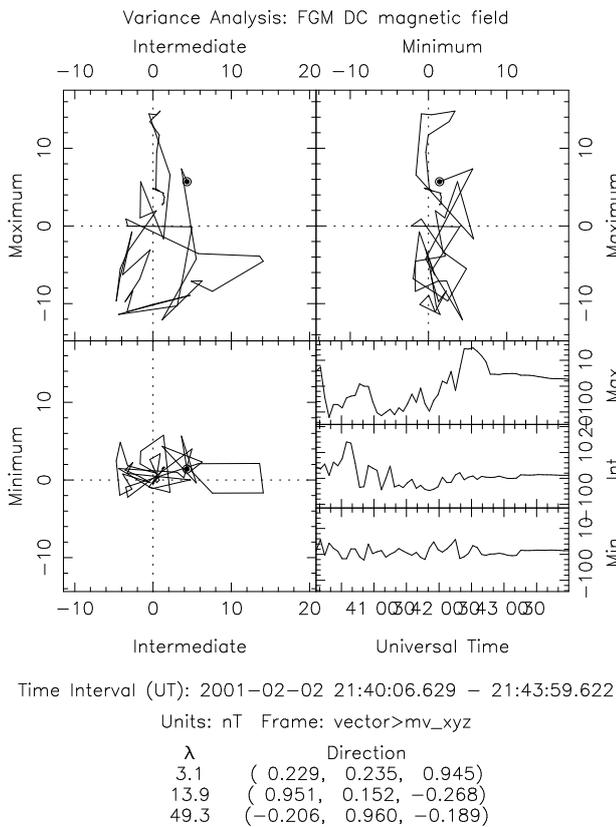


Figure 2: Example: Hodograms - and time series plots - from minimum variance. QSAS2 should provide just about enough capability to do this itself, though perhaps the annotations at the bottom are a bit trickier. Note the equal world coordinate ranges though different limits.

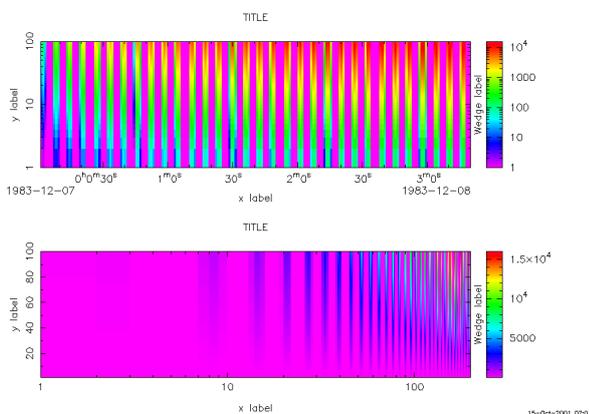


Figure 3: Example: Spectrograms. This taken from my tests of my pplot spectrogram routines showing the various elements.

4.2 1.5-D

Space physicists like to label time axes with, e.g., spacecraft position, magnetic longitude, or other information. In effect this involves drawing a real (or virtual) axes with ticks labelled not by t but by $X(t)$. Options may include the number of ticks to label, whether to draw a separate axis (with perhaps different ticks to the original but correctly aligned) and the associated aesthetics of font type and size, etc.

4.3 2-D

Any object reducible to $X - Y$ or $T - Y$ series, that is, any object from which a scalar series can be extracted.

4.3.1 Examples

1. Scalar series Y_i vs X_i
2. Time series Y_i vs its timetabs T_i
3. Vector components vs one another (hodogram) or time.
4. Array element $A(I, J, i, K, L, \dots)$ vs T_i where
 - I, J, K, L, \dots are specified and fixed
 - Some or all of I, J, K, L, \dots are summed, averaged, or otherwise combined.

Gap-handling is an important aspect here.

4.3.2 Plot Types

These are essentially those of the present QSAS Trace type, i.e., lines and/or symbols, with options to deal with colour, thickness, style, symbol, etc.

4.4 3-D

Any object reducible to $Z(X, Y)$ or $Z(T, Y)$.

4.4.1 Examples

1. 2D arrays $Z(i, j)$
2. Higher-D arrays $Z(I, i, j, K, L, \dots)$ where
 - I, J, K, L, \dots are specified and fixed
 - Some or all of I, J, K, L, \dots are summed, averaged, or otherwise combined.

Gap-handling is an important aspect here.

4.4.2 Plot Types

Spectrogram Z values are colour coded in traditional spectrogram fashion.

Contour Z values as colour filled or contour lines, with/without contour labelling.

Multi-trace $Z(i, J)$ trace for each J . This could be done via the 2-D plotting, but may be better generated automatically here. Useful for data with large dynamic range in Z . Different values of J to be distinguished by a sequence of colours and/or line styles.

4.4.3 Plot Options

The colour table, linear/log scaling, and limits need to be specified. Also values and labels for X, Y or T, Y axes need to be specified; default to index?

Plot options also include the colour “wedge” providing the translation of colour to Z -value.

4.4.4 Axes and Bin Limits

Horizontal and vertical axes need to get information relating to the XY bin corresponding to $Z(i, j)$. For $X(i)$, say, these could be:

1. An $N \times 1$ array of start, centre, or end values for the i^{th} bin. An algorithm to specify the stop, width, or start values, respectively, needs to be employed to determine the N^{th} stop value, width, or 1st start value respectively.
2. $N \times 2$ arrays or $2 N \times 1$ arrays of start and stop values. Note that cef is moving to the latter to ease conformity with cdf.
3. i itself, centred with width 0.5.

This bin information can be used either for labelling or for actual axis scaling/bin size on plot, or both.

5 GUI Aspects

There is a real question as to whether a new panel is created by specifying its type (1-D, 2-D, 3-D, etc.) and then dropping a specific QDOS object into that panel, or dropping a specific QDOS object into a creator slot and then choosing the type. The former allows one to create without confusion a panel containing the density

and particle flux in 100eV bin. The latter makes it easier to drop a vector and instantly get a group of 4 panels with x, y, z , and magnitude respectively. Such a group could, of course be saved, but the memory of the vector component selection needs to be retained, especially if on restore the vector object was replaced with another vector (or a scalar!).

The question can be re-cast as: Do I want to create a plot of a vector component, or do I want to create a line plot? Can we offer both ways of building a plot? What happens when we go to change the plot object under either mode of building? When we restore? When we have multiple traces taken from different types of data objects?

Perhaps there should be wild-card placeholders, such as `__VECTOR1`, `__ARRAY[5D]`, which can be re-assigned once. This would be handy if there is some restored plot which used `B_mag_ampte_gse` and which you want to use for `B_FGM.C1%gse_xyz`; equivalently a global substitute box.

I have thought about the following design, with 2 columns: a ListView of (groups of) panels/plot types and their contents, and an Options Selector, which is probably an index card tabbed region. The left one is used to create and layout the plot; groups of panels are held in sub-elements of the ListView, and can be nested as deep as required. Underneath any panel group live only other panels. Underneath all panels live the objects to be plotted in that panel. Selecting any item from this ListView brings up the Options corresponding to that item. For Panels and Panel Groups, that includes placement, box characteristics, etc. AND also object options (line styles, colours, etc.). For objects it includes line styles, vector/array component selection, etc. Information such as line styles, axis ranges, etc. propagate down from top group to object level. The ability to lock an option to prevent it from being replaced by newly propagated properties is needed, as is the need to offer an over-ride to force it anyway. It may also propagate up/around to other group members.

Right-clicking on a Panel Group, or highlighting it an selecting New Panel from a pull-down menu, enables new panels and or panel groups to be added. Selecting several panels should provide an option to group them. Individual or selected items can be drag'n'dropped to change their position, including dropping on a (collapsed) group to put something inside it. I suspect that most plot layout will be done by relative positioning and fractions of available space, so

the order items appear will correspond to the order in the resulting plot.

Actually, at some point in the past, I had imagined splitting the above into 3 columns, the middle showing the plot objects and the left reserved for groups and panels. On a wish list is an interactive layout tablet to handle the page layout.

Plot output will include xwindows, ps, and gif (I don't know how/if pgplot deals with multiple page gifs yet). If we keep several xwindows open, interactive value return becomes more complicated and we'll need to keep track of the windows in use.

6 Issues and Questions

1. Gap handling in various n-D objects
2. Missing Values and FILLVALs, especially in 3-D objects
3. Dimension Variables
4. Interactive Value Return, especially if we have multi-page plotting. There may also be some complications here with 3-D data if the bin indices are used as world coordinates.