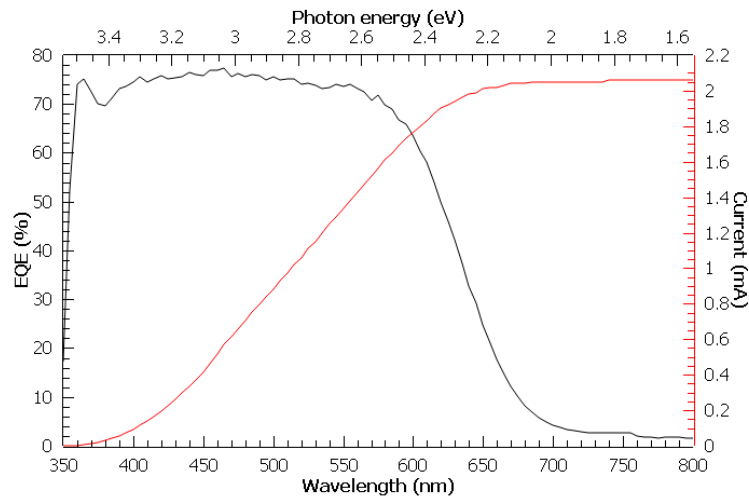


Proposal for independent axes (data ranges)

The ultimate goal for this proposal is to achieve something like this (done with Scidavis):



Here, the black curve is tied to the bottom and left axes, while the red curve is tied to the bottom and right axes. The top axis, despite having no curves tied to it, is still independently defined. Along the way, I will also propose to add some features to make a plotter's life easier, all related to the implementation of independent data ranges.

First of all, how to create independent data ranges? We can take inspiration from the system already in place for the plot's padding, which can be symmetric:

Padding
 Symmetric
Horizontal: 1, 15cm
Vertical: 0, 10cm

or asymmetric:

Padding
 Symmetric
Left: 1, 15cm
Top: 0, 10cm
Right: 0, 40cm
Bottom: 1, 10cm

We could apply the same principle to the data range definition. It could be symmetric:

Symmetric
x-Range: Auto
Format: numeric
Minimum: 100
Maximum: 2500
Scaling: linear
y-Range: Auto
Format: numeric
Minimum: -0.1
Maximum: 1.9
Scaling: linear

Or asymmetric:

Symmetric

x1- Range Auto

Format: numeric

Minimum: 100

Maximum: 2500

Scaling: linear

y1- Range Auto

Format: numeric

Minimum: -0.1

Maximum: 1.9

Scaling: linear

x2- Range Auto

Format: numeric

Minimum: 100

Maximum: 2500

Scaling: linear

y2- Range Auto

Format: numeric

Minimum: -0.1

Maximum: 1.9

Scaling: linear

I am aware that in Labplot you can add as many axes as you want, but the outer ones are still the main ones, so I think this still applies. For the extra ones you create, you will need to be able to choose if you want to link them to the x1 (y1) or x2 (y2) data range, as we're going to do with the curves and everything else. Nowadays they're still tied to one data range, so when you double them, it shouldn't be too difficult to give the option to decide to which data range you want to link them to.

Now for the curves (and everything else really), this is what you can do. When the data ranges are asymmetric, an additional property element can show up to make you decide to which data range to reference them:

General	Line	Symbol	Values	Filling	Error bars
Name:	tmby				
Comment:					
x-data:	Potential (mV) [X]				
y-data:	tmby [Y]				
x-axis:	x1				
y-axis:	y2				

Something similar could be done for extra axes, reference lines, etc.

Now, if you look at the initial plot, you will notice something else that Labplot can't do right now: in the top axis, the number labels don't start at the corner of the plot, but somewhere inside the axis. This is because I set as data range for the top axis the 1.5498-3.5424 range, and it would be stupid to show numbers with all those decimals. Currently in Labplot you can choose the last label to display at the end of the axis by playing with the ticks settings, but the start of the axis will always have a label, corresponding to the beginning of the data range. This can be a bad thing in some cases, when you would like to have the first label not at the beginning of the axis, but somewhere

inside (like the case above). To achieve this, we could add a numeric box option to the Ticks tab of the axis (or possibly to the Labels tab, wherever makes more sense to you), similar to the “Increment:” (or “Number:”, depending on the tick option) one, to allow the definition of the numeric value for the first label (and the associated first major tick). In practice, we have to allow minor ticks as first tick (or even no tick at all at the beginning of the axis, depending on the data range definition) and be more flexible with tick placement in general. To give a practical example of one such occurrence, I may have wanted to start the bottom axis in the initial plot at 345 nm instead of 350 nm, to distance the curve a bit from the left ticks. In this case, the axis would have begun between two minor ticks (the first major tick should still be 350 nm).

One last comment, which is a minor thing but can help to make life easier to people. When the top/bottom and left/right axes data ranges are symmetric, most of the time people want to have specular tick behaviour on them. Currently, you have to set ticks for all four axes independently. It would be nice to have an option to mirror in x2 (y2) the tick settings of x1 (y1), so that we only have to do the work once.