Subject: Irregularly spaced tick-marks on secondary axis.
Posted by Paul Van Delst[1] on Wed, 01 May 2002 14:20:00 GMT
View Forum Message <> Reply to Message

Hey there,

I like to plot radiance data with respect to frequency (in units of centimetres^-1). However, I would like to put a wavelength scale (in units of microns [micrometres]) on the top Xaxis. The relationship between them is very simple but non-linear:

wavelength (um) = 10000.0 / wavenumber (cm^-1)

I have been doing this sort of thing:

```
FUNCTION wticks, axis, index, value
  wavelength = 10000.0d / value
  format = '( f5.2 )'
  RETURN, STRING( wavelength, FORMAT = format )
END


  PLOT, x, y, XSTYLE = 8

  AXIS, XAXIS = 1, $
        XRANGE = !X.CRANGE ,$
        XTICKV = 10000.0d/[ 10d, 11, 12, 13, 14, 15 ], $
        XTICKS = 5, $
        XSTYLE = 1, $
        XTICKFORMAT = 'wticks'
```

As you can see, the above AXIS command assumes something about the XTICKV values. However, if I
now decide to zoom into the plot such that the x-range falls between, say, 11 and 12 microns -- no wavelength scale is shown since XTICKV doesn't contain fractional wavelength values. Is there any way to get IDL to set "nice" wavelength values (via a dynamic XTICKV for e.g.) on the top scale based on the wavenumber range on the bottom scale?

thanx,

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7274
Fax:(301)763-8545

Religious and cultural
purity is a fundamentalist
fantasy
V.S.Naipaul

Subject: Re: Irregularly spaced tick-marks on secondary axis.
Posted by Ken Mankoff on Thu, 02 May 2002 12:55:17 GMT
View Forum Message <> Reply to Message

On Wed, 1 May 2002, Paul Van Delst wrote:
> FUNCTION wticks, axis, index, value
> wavelength = 10000.0d / value
> format = '(f5.2)'
> RETURN, STRING(wavelength, FORMAT = format)
> END
>
> PLOT, x, y, XSTYLE = 8
> AXIS, XAXIS = 1, $
> XRANGE = !X.CRANGE ,$
> XTICKV = 10000.0d/[10d, 11, 12, 13, 14, 15], $
> XTICKS = 5, $
> XSTYLE = 1, $
> XTICKFORMAT = 'wticks'
>
> As you can see, the above AXIS command assumes something about the
> XTICKV values. However, if I now decide to zoom into the plot such
> that the x-range falls between, say, 11 and 12 microns -- no
> wavelength scale is shown since XTICKV doesn't contain fractional
> wavelength values. Is there any way to get IDL to set "nice"
> wavelength values (via a dynamic XTICKV for e.g.) on the top scale
> based on the wavenumber range on the bottom scale?

Can't you make the inputs to XTICKV a function call (as you say, a
"dynamic XTICKV"), and use !X.CRANGE in that function?

This seems fairly straightforward, so if this is not the solution,
please clarify a bit...

The best analogy I can think of to your problem is "MAP_GRID", which
picks nice gridlines for global views (30 degrees) and also nice
gridlines for close-up views (every .001 degrees if the view small
enough). Is this what you are trying to achieve?

-k.
--
Ken Mankoff http://lasp.colorado.edu/snoe/
http://lasp.colorado.edu/mars/
http://lasp.colorado.edu/~mankoff/ http://lasp.colorado.edu/marsrobot/

Subject: Re: Irregularly spaced tick-marks on secondary axis.

Ken Mankoff wrote:
>
> On Wed, 1 May 2002, Paul Van Delst wrote:
>> FUNCTION wticks, axis, index, value
>> wavelength = 10000.0d / value
>> format = '(f5.2)'
>> RETURN, STRING(wavelength, FORMAT = format)
>> END
>>
>> PLOT, x, y, XSTYLE = 8
>> AXIS, XAXIS = 1, $
>> XRANGE = !X.CRANGE ,$
>> XTICKV = 10000.0d/[10d, 11, 12, 13, 14, 15], $
>> XTICKS = 5, $
>> XSTYLE = 1, $
>> XTICKFORMAT = 'wticks'
>>
>> As you can see, the above AXIS command assumes something about the
>> XTICKV values. However, if I now decide to zoom into the plot such
>> that the x-range falls between, say, 11 and 12 microns -- no
>> wavelength scale is shown since XTICKV doesn't contain fractional
>> wavelength values. Is there any way to get IDL to set "nice"
>> wavelength values (via a dynamic XTICKV for e.g.) on the top scale
>> based on the wavenumber range on the bottom scale?
>
> Can't you make the inputs to XTICKV a function call (as you say, a
> "dynamic XTICKV"), and use !X.CRANGE in that function?
>
> This seems fairly straightforward, so if this is not the solution,
> please clarify a bit...
>
> The best analogy I can think of to your problem is "MAP_GRID", which
> picks nice gridlines for global views (30 degrees) and also nice
> gridlines for close-up views (every .001 degrees if the view small
> enough). Is this what you are trying to achieve?

Sort of, but I want the tickmarks on the upper axis to be, um, visually pleasing. So I would like 10, 11, 12, 13, etc microns (which will be irregularly spaced), rather than the wavelength equivalent of the frequency ticks, which would be regularly spaced but would be numbers like 8.3333.., 10, 12.5, 16.666.., and 25 microns for frequencies of 1200, 1000, 800, 600, and 400 cm^-1.

I spent about half-a-day on this yesterday and it is not an straightforward problem. What one needs is some sort of algorithm that, given the *regularly spaced* wavelength ticks [8.3333.., 10, 12.5, 16.666.., and 25] produces an array of "prettified" tick values such as [10, 11, 12, 14, 16, 18, 20, 25]. Note that the tick interval changes to allow for the

1/x spacing otherwise the tick marks will all be smooshed together at the long wavelength end of the axis. And to make it general, it also has to work when you have fractional wavelengths (e.g. for plots between 4.0 and 4.5 um). I'm sure this sort of stuff/algorithm exists (IDL has one for its plots), I just don't know where to find it and I don't want to reinvent it (coz I'll more than likely do a crappy job).

It never ceases to amaze me how these apparently simple things are so difficult to do in a general way.

paulv

--
Paul van Delst Religious and cultural
CIMSS @ NOAA/NCEP/EMC purity is a fundamentalist
Ph: (301)763-8000 x7274 fantasy
Fax:(301)763-8545 V.S.Naipaul

Subject: Re: Irregularly spaced tick-marks on secondary axis.
Posted by Ken Mankoff on Thu, 02 May 2002 14:52:49 GMT
View Forum Message <> Reply to Message

On Thu, 2 May 2002, Paul Van Delst wrote:
> Ken Mankoff wrote:
>>
>> On Wed, 1 May 2002, Paul Van Delst wrote:
>>> FUNCTION wticks, axis, index, value
>>> wavelength = 10000.0d / value
>>> format = '(f5.2)'
>>> RETURN, STRING(wavelength, FORMAT = format)
>>> END
>>>
>>> PLOT, x, y, XSTYLE = 8
>>> AXIS, XAXIS = 1, $
>>> XRANGE = !X.CRANGE ,$
>>> XTICKV = 10000.0d/[10d, 11, 12, 13, 14, 15], $
>>> XTICKS = 5, $
>>> XSTYLE = 1, $
>>> XTICKFORMAT = 'wticks'
>>>
>>> As you can see, the above AXIS command assumes something about the
>>> XTICKV values. However, if I now decide to zoom into the plot such
>>> that the x-range falls between, say, 11 and 12 microns -- no
>>> wavelength scale is shown since XTICKV doesn't contain fractional
>>> wavelength values. Is there any way to get IDL to set "nice"
>>> wavelength values (via a dynamic XTICKV for e.g.) on the top scale
>>> based on the wavenumber range on the bottom scale?
>>

>> Can't you make the inputs to XTICKV a function call (as you say, a
>> "dynamic XTICKV"), and use !X.CRANGE in that function?
>>
>> This seems fairly straightforward, so if this is not the solution,
>> please clarify a bit...
>>
> Sort of, but I want the tickmarks on the upper axis to be, um,
> visually pleasing. So I would like 10, 11, 12, 13, etc microns (
> which will be irregularly spaced), rather than the wavelength
> equivalent of the frequency ticks, which would be regularly spaced
> but would be numbers like 8.3333.., 10, 12.5, 16.666.., and 25
> microns for frequencies of 1200, 1000, 800, 600, and 400 cm^-1.
>
> I spent about half-a-day on this yesterday and it is not an
> straightforward problem. What one needs is some sort of algorithm
> that, given the *regularly spaced* wavelength ticks [8.3333.., 10,
> 12.5, 16.666.., and 25] produces an array of "prettified" tick
> values such as [10, 11, 12, 14, 16, 18, 20, 25]. Note that the tick
> interval changes to allow for the 1/x spacing otherwise the tick
> marks will all be smooshed together at the long wavelength end of
> the axis. And to make it general, it also has to work when you have
> fractional wavelengths (e.g. for plots between 4.0 and 4.5 um). I'm
> sure this sort of stuff/algorithm exists (IDL has one for its
> plots), I just don't know where to find it and I don't want to
> reinvent it (coz I'll more than likely do a crappy job).

I would do this:
1) Plot your data on frequency, from f0 to f1
2) Make a 2nd array of frequencies, from f0 to f1, with 10 sub-values
   IDL> f = makex(f0, f1, (f1-f0)/10.)
3) convert f to um.
   You now have your secondary X-AXIS, with its non-linear values
   spaced evenly 10 times over the plot. This is true regardless of
   your original axise being 10 to 1e3 or 14.001 to 14.003. The
   only problem is that these values are "ugly" fractions. At least,
   from my understanding of the problem, this is the only fault so far.
4) prettify your 'um' array via CIEL(), FLOOR() and ROUND(). For the
   case where you have all floats, a float-type ROUND() is:
   IDL> pretty = FIX(ugly * 10) / 10.
   now pass this to XTICKN and XTICKV

Step 4 is a bit (only a little bit) dynamic, in that a ROUND() will
work for a large frequency range, a FIX(ugly*10)/10. for a small
frequency range, and a FIX(ugly*100)/100. for a very small range.

EX (cut-n-paste):
NOTE that if you use the 2nd frequency (f = [14.001,14.003]), then

your "prettify" code (the um2= section) needs to change...

```
f = [ 10, 11, 12, 13, 14, 15 ]*10
;f = [ 14.001, 14.003]
plot, f, indgen(6), xst=8
f0 = f[0] & f1 = f[ n_elements(f)-1 ]
ff = makex( f0, f1, (f1-f0)/10. )
um = 10000d / ff
xr = [ um[0], um[ n_elements(um)-1 ] ]

;;; this part is a function of your "zoom" level (the range of f)
;um2 = round(um)          ; big f
um2 = fix( um*10 )/10.     ; small f
;um2 = fix( um*100 )/100.  ; teeny-tiny f

axis, /xaxis, xrange=xr, /xst, xtickv=um2, xtickn=strtrim(um2,2)
; xtickn should use a FORMAT statement to look prettier...
```

-k.

--

Ken Mankoff http://lasp.colorado.edu/snoe/
http://lasp.colorado.edu/mars/
http://lasp.colorado.edu/~mankoff/ http://lasp.colorado.edu/marsrobot/