

---

Subject: Re: Fractional Pixels Origin?

Posted by [JD Smith](#) on Wed, 22 Feb 2006 01:25:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Wed, 22 Feb 2006 10:44:25 +1300, Mark Hadfield wrote:

> JD Smith wrote:

>> [snip]

>>

>> I don't see your argument. Fractional pixels are useful only when  
>> calculating things which relate pixel coordinates to some other  
>> coordinate (like celestial coordinates on the sky, etc), or when  
>> computing other derived fractional pixel positions (e.g. clip two  
>> polygons). Obviously, the computer has no understanding of a  
>> fractional pixel, but only the memory indexed offset [0,0]. But the  
>> latter does not have to drive the former. In fact I'd say it's rather  
>> strange to let the layout in memory dictate a physical coordinate  
>> system. You don't need to make this distinction.

>

> Hang on. I'm not letting layout in memory dictate a physical coordinate  
> system. I'm just adopting a convention for "image plots" that is the  
> same as for other plot types.

>

> I have a data array, mydata, dimensioned [m,n]

>

> When I type

>

> contour, mydata

>

> the contour routine implicitly locates the data points at x =  
> [0,...,m-1], y = [0,...,n-1]. Ditto when I type

>

> surface, mydata

>

> Similarly for plots of 1D arrays.

>

> When I plot this data via a false-colour image using my own  
> image-plotting routine

>

> mgh\_image\_plot, mydata

>

> then I adopt the same convention. If the image is "blocky" (eg an  
> IDLgrImage with INTERPOLATE=0) then the cell centres represent the data  
> points; the image has to fill the space  $-0.5 \leq x \leq m-0.5$  (similarly  
> for y) to get them in the right location. If the image is "smooth" (eg  
> an IDLgrImage with INTERPOLATE=1) then the outer row & column of data  
> points lie on the edge of the image and it fills the space  $0 \leq x \leq$   
> m-1, etc.

My concept of fractional pixels treats them as a physical unit, like inches. If you had a plot of sound speed vs. linear density, and had an image with the same physical axes, you'd have to make special arrangements to have the occupy the same region of your screen, and it wouldn't concern you. Why are "pixels" different? The fact that choosing [0,0] as the origin to make things line up is an artifact of how plot/contour/surface/etc. work.

A good example is PSYM=10. Anyone who plots histograms will quickly realize the origin choice of PLOT,PSYM=10 is off by one-half bin, and codes around it. A histogram at position  $q$  with bin width  $w$  should be a line from  $q-w/2$  to  $q+w/2$ , not  $q$  to  $q+w$  (or whatever it uses). My point is that there are reasons for adopting a given pixel origin which are entirely separate from the convenience of falling in line with what other tools or data systems have adopted (e.g. [1,1] in FITS, [0,0] in IDL, etc.). For some uses (such as making images and plot line up), perhaps that choice is a good one.

Anyway, given that so many standards abound, I think it's an issue we'll all have to continue to deal with.

JD

---