OK, while we're talking about this, I've got another gripe, I mean,
constructive comment about asinhscl. This time, it's about reproducibility of
scaling. I just spent an hour tracking down some strange behavior with some
plots, which I ultimately traced to how Scale_Vector works.

The scenario is this. I have a couple of images which I want to display
with the same scaling, and then I also want to display a color bar.
I want to ensure that all the images have precisely the same scaling,
so I say
 minval = min(image1) < min(image2) < min(image3)
 maxval = min(image1) > min(image2) > min(image3)

and then I can call
 tvimage,asinhscl(image1,min=minval,max=maxval)
 tvimage,asinhscl(image2,min=minval,max=maxval)
 tvimage,asinhscl(image3,min=minval,max=maxval)

However, this did *NOT* do what I expected. Pixels in the images which should
have had the exact same output greyscale ended up having slightly different
values.

I'll cut to the chase. The problem is that some images have maxes and mins
different from the overall max and min, so the first Scale_Image call in
asinhscl maps them into, say 0.03-0.98 instead of the full range 0.0-1.0. Then
the *second* call to Scale_Vector doesn't have explicit min and max set, so
it maps asinh(0.03) to 0 and asinh(0.98) to 1. Thus the color map
is slightly different for each image. Whoops!

The fix is relatively simple - figure out where asinh sends 0 and 1, and
then set those as the range for the second Scale_Vector:

 extrema = asinhscl_asinh([0.0,1.0]*nonlinearity)
 output = Scale_Vector(ASinhScl_ASinh(Temporary(output)*nonlinearity), $
    minOut, maxOut, MinValue=extrema[0], MaxValue=extrema[1], /NAN, Double=1)

I think at this point we're starting to converge on a final version.
I've also got a hacked-up version of your colorbar.pro for making
asinh scaled colorbars, but I want to polish it a bit more before releasing
it into the wild. :-)

 - Marshall