
Subject: Re: HISTOGRAM and the Razor's Edge.
Posted by [Paul Van Delst\[1\]](#) on Thu, 12 Jun 2003 14:40:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Tim Robishaw wrote:

```
>
> IDL> print, ([-5.50,-5.45,-5.40,-5.35,-5.30,-5.25]-(-5.50))/0.05,
> format='(f21.19)'
> 0.00000000000000000000
> 1.0000038146972656250
> 1.9999980926513671875
> 3.0000019073486328125
> 3.9999961853027343750
> 5.00000000000000000000
>
> Well, there ya go. It's a roundoff error problem that results from
> trying to balance the values on a razor's edge... the subtraction and
> division knock a few values off balance. But, the result is still
> WRONG and I just don't know enough about roundoff error to know if
> this is an insurmountable problem (but my educated guess is: yes.) Is
> there some clever way to make HISTOGRAM behave properly in such
> situations?
```

The result isn't wrong. Your assumptions about the numerical accuracy are. I'm amazed you're surprised at getting a result of 1.9999980926513671875 for the expression $(-5.40 - (-5.50))/0.05$, Even double precision won't help you here:

```
IDL> print, (-5.40d0-(-5.50d0))/0.05d0, format='(f21.19)'
1.99999999999999928946
```

The subtraction and division don't knock a few values "off balance", the values are simply not representable to the precision you require - such is the nature of float point numberology. As other posters suggested, you need to preprocess the values somehow to make them integers, or take into account the precision when you do the comparisons required. It's worth writing a separate function to do floating point number comparisons. In Fortran I do the following:

```
!      ABS( x - y ) < ( ULP * SPACING( MAX(ABS(x),ABS(y)) ) )
!
!      If the result is .TRUE., the numbers are considered equal.
!
!      The intrinsic function SPACING(x) returns the absolute spacing of numbers
!      near the value of x,
!
!      { EXPONENT(x)-DIGITS(x)
!      { 2.0                for x /= 0
!      SPACING(x) = {
!      {
```

```
!           { TINY(x)           for x == 0
```

```
!
```

```
!   The ULP optional argument scales the comparison.
```

where

```
!   ULP: Unit of data precision. The acronym stands for "unit in
!       the last place," the smallest possible increment or decrement
!       that can be made using a machine's floating point arithmetic.
!       A 0.5 ulp maximum error is the best you could hope for, since
!       this corresponds to always rounding to the nearest representable
!       floating-point number. Value must be positive - if a negative
!       negative value is supplied, the absolute value is used.
!       If not specified, the default value is 1.
```

I don't know off the top of my head how to translate all that to IDL, but it would be a useful thing to do.

paulv

```
> =====
```

```
=====
```

```
> "Nothing shocks me. I'm a scientist." - Indiana Jones
```

```
> -----
```

```
> Tim Robishaw                UC Berkeley Astronomy
```

Maybe a necessary, but not sufficient, condition to stave off the shocks. :o)

--

Paul van Delst
CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7748
Fax:(301)763-8545
