Subject: Re: converting floats to doubles
Posted by Michael Wallace on Fri, 20 May 2005 20:52:03 GMT
View Forum Message <> Reply to Message

First, let me assure you that when you convert any number from a float to a double, there is absolutely no change in value. When you print out a value and do not supply a specific format, IDL will show a different number of decimal places depending on whether the number is a float or double. Here's an example of what I mean:

```
IDL> print, float(3), double(3)
      3.00000       3.0000000
```

The above results are just because IDL's default format for floats are different than the default format for doubles.

Floating point numbers do not represent a number exactly. Floating point numbers are composed of a sign bit, exponent and a mantissa. These three values are then fed into an equation which then produces the actual floating point number we see. This why I can say that converting a number from a float to a double doesn't change anything. The mantissa, exponent and sign bit of the float are copied directly into sign bit, mantissa and exponent of the double. The extra bits in the double are just left at 0.

dblarr(n) is the same as double(fltarr(n)). The fltarr(n) will create n many floating point numbers, all of which are 0. Converting all these floats into doubles will yield a double array where all values are 0 and this is the very same thing as dblarr(n).

sqrt(dblarr(n)) is not the same thing as double(sqrt(fltarr(n)). In the latter, you are taking the sqrt of the floating point numbers first. There will be precision lost because the float mantissa is only capable of storing so much information. If you take this value and cast it into a double, the less precise float value is preserved and the other bits of the double's mantissa are left at 0. Had you done sqrt(dblarr(n)), the sqrt operation would have been calculated using double precision arithmetic and the entire mantissa of the double would be filled. Because the double's mantissa is larger than the float's mantissa, it is able to store more precision.

A lot of the gory details of IEEE 754, the specification of floating point numbers, can be found here: http://en.wikipedia.org/wiki/IEEE_754.

-Mike

Benjamin Hornberger wrote:

> Hi computation gurus,
>
> is dblarr(n) equivalent in precision to double(fltarr(n))? I know that
> in a case like sqrt(dblarr(n)) vs. double(sqrt(fltarr(n))), they are not
> equivalent (the second version is not true double precision). But I
> thought when I start with whole numbers anyway, it might be the case.
>
> In other words, when a floating point number is converted to double, are
> the additional digits always set to zero, or is it possible that they
> aren't? I tried it out by printing some numbers, and it looks like they
> add only zeroes, but I would be happy if the experts could confirm.
>
> Thanks for any insight,
>
> Benjamin