Subject: Re: The total function in IDL (RSI read please)
Posted by thompson on Sun, 08 Aug 1993 14:55:12 GMT
View Forum Message <> Reply to Message

terry@toe.CS.Berkeley.EDU (Terry Figel) writes:

> Shouldn't the total function in IDL do it's arithmetic in using a double ????
> i.e. if I have an float array, shouldn't
> f=fltarr(512,512)
> ; Fill in the array
> total(f) should = total(double(f)) ; it DOES NOT!.
> It would seem that the total routine uses double arithmetic if the array is double,
> it uses a float (which produces the wrong output if it is float)

Congradulations, you've just discovered computer round-off error. :^) The behavior you're describing is to be expected. I disagree that this is a problem with IDL. At first glance, it would appear to be nice if all floating point arithmetic was done in double precision to minimize the impact of round-off. However, on most computers, double precision arithmetic is slower (sometimes *much* slower) than doing the same operation in single precision, not to mention the extra overhead involved in converting back and forth between single and double precision.

In other words, although you may want operations like TOTAL to automatically switch over to double precision mode, that isn't necessarily what everyone wants.

This sort of behavior is endemic to all computer languages, not just IDL. For example, consider the following lines of FORTRAN code

```
 TOTAL = 0.0
 DO I = 1,N
  TOTAL = TOTAL + ARRAY(I)
 ENDDO
```

The result could be different depending on whether TOTAL was defined as single or double precision.

If you need to have the operation performed in double precision, then use TOTAL(DOUBLE(ARRAY)) instead of TOTAL(ARRAY).

Bill Thompson