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

Subject: Re: The total function in IDL (RSI read please)
Posted by gurman on Sun, 08 Aug 1993 20:20:30 GMT
View Forum Message <> Reply to Message

thompson@serts.gsfc.nasa.gov (William Thompson) writes:

> 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.

Bill -

I don't know what happens on SPARCstations, but interestingly enough, on Alphas running OpenVMS, running the total loop you suggest in FORTRAN takes exactly twice as long in double precision as in single. At first sight, that might seem kind of strange, since the Alpha is supposedly a "64-bit machine," but it is able to do 32-bit operations, 2 per cycle. Thus, a large number of 32-bit operations take half as long as the same number of 64-bit operations. On the IBM Power/RISC chip, I understand that there can be 6 operations per cycle, so things probably still scale simply.

For what it's worth, the total time to add a million REAL*4 numbers on a DEC 3000/400 was 0.095 s, vs. 0.190 s for a million REAL*8's. In a year or two, when that becomes one of the slower machines IDL will be running on, I don't see any reason to default to FLOAT instead of DOUBLE for TOTAL operations in IDL, do you?

Joe Gurman

--
J.B. Gurman / Solar Physics Branch / NASA Goddard Space Flight Center /
Greenbelt MD 20771 USA / gurman@uvsp.gsfc.nasa.gov
| Federal employees are prohibited from holding opinions under the Hatch Act.|
| Therefore, any opinions expressed herein are somebody else's. |