Subject: Re: Specifying DOUBLE precision and using WHERE
Posted by Martin Schultz on Mon, 28 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Since roundoff errors keep popping up in this newsgroup: Here's an old
trick that I learned in the early 80's ;-)

 instead of comparing   (A eq B)   use (ABS(A-B) lt SMALL)
where SMALL is a small number relative to the number you are testing but
within the numerical resolution. In your example
   ind = where(ABS(data-missing) lt 1.e30

ms = -9.96921e+36
md = -9.96921d+36
print,ms eq md, abs(ms-md) lt 1.e30
   0   1

some more details below ....

Martin

Johnny Lin wrote:
>
> hi folks,
>
> i'm using WHERE to do a test for a missing value in an array of data.
> the funny thing is the result of WHERE seems to change depending on
> whether I use the function DOUBLE to set the type of the missing
> value field (or of the data i'm testing), or whether I set it using
> "d" as the exponent. is there a difference in specifying the type
> of a variable using "d" vs. DOUBLE? below is a sample of 4 different
> cases that illustrate what i'm describing.
>
> thanks!
>
> cheers,
> -Johnny Lin
>
> ---------------------------------------
> UCLA Department of Atmospheric Sciences
> E-mail: jlin@ucla.edu
> ---------------------------------------
>
> print,!version
> { sparc sunos unix 5.1 Apr 13 1998}
>
> CASE #1 (missing and data are both FLOAT):

ok

> CASE #2 (missing is DOUBLE using "d" exponent and data is FLOAT):
> missing=-9.96921d+36
> data=randomn(-34,20)
> data[where(data lt 0.1)]=missing
> print,where(data eq missing)
> ; -1
here you have a roundoff error. Try
  print,missing,format='(e20.14)'
  print,data[1],format='(e20.14)'

>
> CASE #3 (missing is made DOUBLE using DOUBLE fctn. and data is FLOAT):
> missing=-9.96921e+36
> missing=double(missing)
> data=randomn(-34,20)
> data[where(data lt 0.1)]=missing
Here you define missing as float, so it is internally the same number as
the missing data values. The DOUBLE function will just add a few zeros
to the mantissa, but the relevant digits remain the same. Thus, in
effect you are still comparing float to float.

> CASE #4 (missing is DOUBLE using "d" exponent and data is DOUBLE using
> DOUBLE fctn.):
> missing=-9.96921d+36
> data=double(randomn(-34,20))
> data[where(data lt 0.1)]=missing
This one only works because you first convert your data and then set the
values that you select to missing. That is you are comparing the same
double representation of missing. Try
  missing=-9.96921d+36
  data=randomn(-34,20)
  data[where(data lt 0.1)]=missing
  data=double(data)
and you'll see it won't work.

Martin

|||||||||||||||\\\\\\\\\\\\\\\----------////////////// //|||||||||||||||
Martin Schultz, DEAS, Harvard University, 29 Oxford St., Pierce 109,
Cambridge, MA 02138 phone (617) 496 8318 fax (617) 495 4551
e-mail mgs@io.harvard.edu web http://www-as/people/staff/mgs/