
Subject: Re: rounding errors

Posted by [thompson](#) on Fri, 27 Apr 2001 18:34:07 GMT

[View Forum Message](#) <> [Reply to Message](#)

"Dominic R. Scales" <Dominic.Scales@aerosensing.de> writes:

> Unfortunately, the input data is a fixed format I get from an external
> source and can not change. ...

When you say that it's in a fixed format, do you mean an ASCII format (READF), or a binary format (READU)? If the former, then you can do exactly what you want by reading it in as binary in the first place, e.g.

```
A = 0.D0  
READF, UNIT, A
```

for a single value, or

```
A = DBLARR(N_VALUES)  
READF, UNIT, A
```

for an array. You will at least then retain the precision of all the digits that were printed out.

Digits not printed out, on the other hand, are lost forever. If your file contains the number 2.56989 printed out as a series of digits, it's rather naive to think of this as 2.569890000000..., it really could have been anything from 2.569885 to 2.569894999... before it was rounded off to six digits of precision.

If you're instead reading in single precision binary numbers via READU, then you've already made one approximation when you've printed it out as 2.56989. Binary data aren't stored in decimal notation.

> The case I am trying to make is this: why aren't the missing digits, i.e.
> the ones added by the cast, set to 0.? ...

They are set to 0, in binary notation, as I demonstrated in my last post. They're not really "random". It's just the conversion from binary to decimal notation that makes it look so.

> ... As I go along with my calculations,
> these random additional digits give me a hell of a headache by accumulating.

But why would you expect that treating 2.56989 as 2.5698900000000000 is any more accurate than 2.5698900222778320? It's only a human prejudice to want to write it out that way. Computers have a different prejudice to fill out with zeros in binary notation rather than decimal, but that's just as valid as the

human prejudice towards decimal notation. In fact, this particular result is two orders of magnitude closer to what you want than you have any right to expect, differing only in the 9th digit.

- > And, as Murphy leads us to expect, always in the 'wrong' direction.
- > I know, one shouldn't test floating point numbers in digital rep. against
- > one another, not even if they were double.

- > Starting with numbers that are said to have a precision of 15 digits but
- > have random digits after the 6th!?! Why should I even bother?
- > BUT: it does work for a cast via `double(string)`...

But if your data is only stored with 6 digit accuracy, how can you hope to recover 15 digit accuracy. Filling the data out with zeros in decimal (or binary) notation is really making up data out of thin air. It's not really justified, except that one has to fill it up with something.

- > Yep, it isn't idl's fault here. It only cost me some time to notice...

 - > well, cu all,
 - > Dominic
 - > --
 - > Dipl. Phys. Dominic R. Scales | Aero-Sensing Radarsysteme GmbH
 - > Tel: +49 (0)8153-90 88 90 | c/o DLR Oberpfaffenhofen
 - > Fax: +49 (0)8153-908 700 | 82234 Wessling, Germany
 - > WWW: aerosensing.de | email: Dominic.Scales@aerosensing.de
-