

Subject: Re: DOUBLE precision no precise??
Posted by [Vincent Schut](#) on Tue, 05 Mar 2002 15:07:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Reimar Bauer wrote:

> James Tappin wrote:

 $\succ$

>> David Williams wrote:

>>

>>

>>> I've always had heaps of help from the inhabitants of this newsgroup --

>>> for which I am eternally grateful -- despite my often stupid questions.

>>> So, when a mate of mine came across this 'quirk' yesterday, and I wasn't

>>> sure how to help him out, I thought I'd ask this group.

>>>

>>> He has an array of numbers that he wants to apply a user-defined

```
>>> function to, but we're both a little disturbed by the fact that if you
```

>>> do the calculations with a pocket calculator, you get different numbers

>>> than if you perform the same calculation in IDL.

>>>

>>> To try and find where the problem is, we tried the following lines...

>>>

```
>>> IDL> a = DOUBLE(42766.080001)
```

```
>>> IDL> print,a,FORMAT='(F24.17)'
```

>>>

```
>>> 42766.078125000000000000
```

>>>

>>> As you see, the number we get out isn't the same as the number we

>>> entered. I'm guessing it's to do with the way IDL stores numbers in

>>> memory, but my understanding of low-level computational processes isn't

>>> great.

>>>

>>> Can anybody help me understand what's going on, and/or if there's a way

>>> around? I'd really appreciate whatever help is on offer, so thanks in

>>> advance.

>>

>> The problem is that 42766.080001 is a single precision constant, so what's

>> happening is that you are storing the single-precision approximation to

```
>> 42766.080001 in some scratch location, then converting that to double.
```

>>

>> What you actually want is:

```
>> a=42766.080001D0
```

≥ ≥

 $\gg -$
$$\gg + \cdots + \cdots + \cdots +$$

>> | James Tappin | School of Physics & Astronomy | O

>> | sit@star.sr.bham.ac.uk | University of Birmingham | -- V |

>> | Ph: 0121-414-6462. Fax: 0121-414-3722

>> +-----+-----+
>
>
> My vote to this answer!
>
>
> Reimar

ehm... mine too, after taking a better look at the question :-)
Was too fast with a too complex answer for this, sorry.
(It *is* a fact, though, that Borland C++ and g++ (GNU c++) give
slightly different values for a double precision pi... But this is of no
concern here.)

Vincent.
