Subject: Re: DOUBLE precision no precise??
Posted by James Tappin on Tue, 05 Mar 2002 12:14:50 GMT
View Forum Message <> Reply to Message

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)'
>
> 42766.07812500000000000
>
> 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

--
 +----------------------+-----------------------------+-- -------+
James Tappin	School of Physics & Astronomy	O__
sjt@star.sr.bham.ac.uk	University of Birmingham	-- \/`
Ph: 0121-414-6462. Fax: 0121-414-3722	
 +---+-- -------+

Subject: Re: DOUBLE precision no precise??
Posted by Vincent Schut on Tue, 05 Mar 2002 13:01:35 GMT
View Forum Message <> Reply to Message

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)'
>
> 42766.07812500000000000
>
> 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.
>
>
> Dave
>
> ---------- ---------
> David R. Williams, | BT7 1NN, Northern Ireland.
> Astrophysics & Planetary Science, | d.williams@qub.ac.uk
> Queen's University, Belfast, | http://star.pst.qub.ac.uk/~drw/
> ========== =========

I'm no expert on math precision, but I think that this is a known 'problem', caused by the different possible ways to store floating point precision data. (For example, different C++ compilers also give different values for a double precision float constant of pi, nice eh? :-)) It might help to read the idl help section called 'accuracy & floating point operations', as a start. Maybe others in the group can give you a more specific answer, though.

cheers,

Vincent.

Subject: Re: DOUBLE precision no precise??
Posted by R.Bauer on Tue, 05 Mar 2002 14:07:37 GMT
View Forum Message <> Reply to Message

James Tappin wrote:
>
> 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)'
>>
>> 42766.07812500000000000
>>
>> 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
>
> --
> +----------------------+------------------------------+-- -------+
> | James Tappin | School of Physics & Astronomy | O__ |

> | sjt@star.sr.bham.ac.uk | University of Birmingham | -- \/` |
> | Ph: 0121-414-6462. Fax: 0121-414-3722 | | |
> +---+-- -------+

My vote to this answer!

Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
-- -------
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml
== =======

Subject: Re: DOUBLE precision no precise??
Posted by Martin Downing on Tue, 05 Mar 2002 14:30:30 GMT
View Forum Message <> Reply to Message

Hi David,

James is correct. You are attempting to assign a double constant
incorrectly, since the default for a floating point constant is a single
precision (FLOAT) number. Append with the letter "d" and all will be well.
You will find many recent discussions in the group on the consequences of
floating point arithmetic with inadequate precision, in the mean time I hope
the following helps you see where your precision was lost!

cheers

Martin
======================

IDL> a = 42766.080001

IDL> help , a

A FLOAT = 42766.1

IDL> print, a, format = '(f30.20)'

42766.078125000000000000

IDL> a = double(a)

IDL> print, a, format = '(f30.20)'

42766.078125000000000000

IDL> a = 42766.080001d

IDL> help , a

A DOUBLE = 42766.080

IDL> print, a, format = '(f30.20)'

42766.080001000002000000

==========
--

Martin Downing,
Clinical Research Physicist,
Grampian Orthopaedic RSA Research Centre,
Woodend Hospital, Aberdeen, AB15 6LS.
Tel. 01224 556055 / 07903901612
Fax. 01224 556662

m.downing@abdn.ac.uk

"David Williams" <d.williams@qub.ac.uk> wrote in message
news:3C84B20D.57963F41@qub.ac.uk...
>
> 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)'
>

> 42766.07812500000000000
>
> 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.
>
>
> Dave
>
> ---------- ---------
> David R. Williams, | BT7 1NN, Northern Ireland.
> Astrophysics & Planetary Science, | d.williams@qub.ac.uk
> Queen's University, Belfast, | http://star.pst.qub.ac.uk/~drw/
> ========== =========

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:
>
>> 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)'
>>>
>>> 42766.07812500000000000
>>>

>>> 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
>>
>> --
>> +----------------------+-----------------------------+-- -------+
>> | James Tappin | School of Physics & Astronomy | O__ |
>> | sjt@star.sr.bham.ac.uk | University of Birmingham | -- \/` |
>> | 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.

Subject: Re: DOUBLE precision no precise??
Posted by Robert Stockwell on Tue, 05 Mar 2002 15:32:36 GMT
View Forum Message <> Reply to Message

David Williams wrote:

...
> To try and find where the problem is, we tried the following lines...
>
> IDL> a = DOUBLE(42766.080001)

> IDL> print,a,FORMAT='(F24.17)'
>
> 42766.07812500000000000
>
...
> Dave
>
> ---------- ---------
> David R. Williams, | BT7 1NN, Northern Ireland.
> Astrophysics & Planetary Science, | d.williams@qub.ac.uk
> Queen's University, Belfast, | http://star.pst.qub.ac.uk/~drw/
> ========== =========
>

IDL> a = DOUBLE(42766.080001)
This statements is the same as a = DOUBLE(FLOAT(42766.080001))
since that is how it gets parsed by IDL. This is how it should work
since you did not define the constant as a double (although
people could argue that double should be the default type).

Anyways, this is what you want to do (note the "d" at the end
of the number).

IDL> print,42766.080001d,format='(f24.12)'
42766.080001000002

Cheers,
bob

Subject: Re: DOUBLE precision no precise??
Posted by William Clodius on Tue, 05 Mar 2002 16:11:54 GMT
View Forum Message <> Reply to Message

David Williams wrote:<snip>

> To try and find where the problem is, we tried the following lines...
>
> IDL> a = DOUBLE(42766.080001)
> IDL> print,a,FORMAT='(F24.17)'
>
> 42766.07812500000000000
>

> 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.
>

<snip>

All computers IDL is available for store numbers in memory using a binary representation. This representation comes in at least two forms, single (float) and double precision. Both representations can be thought of as typically representing a number by an integer multiplied by a scale factor (exponent) that is an integer power of two. Double uses twice as many bits as float to allow a larger range of integers and scale factors. Because of the finite range of the integers, and because the exponent is a power of two and not a power of ten, only an infinitesimal fraction of the numbers that can be written exactly in decimal can be represented exactly in a finite binary representation. This is a common source of confusion for users of most programming languages. (There are some languages that use less efficient representation such as decimal or rational arithmetic, but such languages, in addition to their inefficiencies, often provide only the simplest mathematical operations.)

In addition to this common source of confusion, your code has an additional problem that is almost as common among such languages. You apparently don't understand the lexical conventions used to distinguish between literals that represent single and double precision numbers. IDL iginores your DOUBLE in deciding this. Instead it interprets your 42766.08001 as a single precision literal, and finds the nearest representable value, which is only accurate to about 7 decimal places. If you want a literal to be interpretted as a double precision, it mus have D# (or d#) as a suffix, where # is an appropriate decimal exponent, i.e. you could represent 42766.08001 as any of
42766.08001D0
42766.08001 d0
42.76608001 D3
4.276608001 D4
0.4276608001 D5
...
to have it interpretted as a double precision number.

Subject: Re: DOUBLE precision no precise??

Posted by James Kuyper on Tue, 05 Mar 2002 16:29:10 GMT
View Forum Message <> Reply to Message

Vincent Schut wrote:

> David Williams wrote:
...

>> To try and find where the problem is, we tried the following lines...
>>
>> IDL> a = DOUBLE(42766.080001)
>> IDL> print,a,FORMAT='(F24.17)'
>>
>> 42766.07812500000000000
...

> I'm no expert on math precision, but I think that this is a known
> 'problem', caused by the different possible ways to store floating point
> precision data. (For example, different C++ compilers also give
> different values for a double precision float constant of pi, nice eh?

It's not quite the same problem. The C++ problem is due to the fact that different implementations of C++ are free to implement double precision math with different degrees of precision. There's a minimum required precision, but it's not a very strict requirement. And since the C standard library isn't even required to provide a value of pi, implementations are free to provide it with whatever precision they choose.

The IDL problem is due to the fact that the default precision in IDL is 'FLOAT', rather than 'double', which is the default precision for C/C++. Thus,

IDL	C/C++
float	
42766.080001	42766.080001F
double	
42766.080001D	42766.080001
long double N/A	42766.080001L

Subject: Re: DOUBLE precision no precise??
Posted by dave_r_williams on Tue, 05 Mar 2002 17:20:58 GMT
View Forum Message <> Reply to Message

> Anyways, this is what you want to do (note the "d" at the end
> of the number).
>
>

>
> IDL> print,42766.080001d,format='(f24.12)'
> 42766.080001000002

James, Reimar, Vincent, Martin, Bob: Thanks a million.

Once again, another stupid question asked, another patient answer given!

Much obliged to you all.

Dave

Subject: Re: DOUBLE precision no precise??
Posted by Pete[1] on Tue, 05 Mar 2002 21:11:11 GMT
View Forum Message <> Reply to Message

"Dave Williams" <dave_r_williams@hotmail.com> wrote:
>> Anyways, this is what you want to do (note the "d" at the end
>> of the number).
>> IDL> print,42766.080001d,format='(f24.12)'
>> 42766.080001000002
>
> James, Reimar, Vincent, Martin, Bob: Thanks a million.
>
> Once again, another stupid question asked, another patient answer given!
>
> Much obliged to you all.
>
> Dave

Even if you do use double precision in IDL, you'll probably find that your
pocket calculator is more accurate than your computer. A decent calculator
will use high precision or BCD or something weird like that. It has time
on its side.

Pedantically,
Peter Mason
