Subject: Arithmetic error from NR_MACHAR()
Posted by on Mon, 29 Jun 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Hi all,

I tried to find out the machine precision on my LINUX PC with IDL 5.0.

The result is the following:

```
IDL> help, /struc, NR_MACHAR(/DOUBLE)
** Structure DMACHAR, 13 tags, length=68:
   IBETA           LONG                 2
   IT              LONG                64
   IRND            LONG                 2
   NGRD            LONG                 0
   MACHEP          LONG               -63
   NEGEP           LONG               -64
   IEXP            LONG                 2
   MINEXP          LONG             -1075
   MAXEXP          LONG             -1073
   EPS             DOUBLE       1.0842022e-19
   EPSNEG          DOUBLE       5.4210109e-20
   XMIN            DOUBLE           0.0000000
   XMAX            DOUBLE                 Inf
% Program caused arithmetic error: Floating divide by 0
```

It seems, that MAXEXP has got the wrong value.
Also XMAX, which is according to the manual calculated as
(1-EPSNEG)*IBETA^MAXEXP, is Infinity, what may be the origin of the
error message.
But then I tried this:

```
IDL> prec = NR_MACHAR(/DOUBLE)
% Program caused arithmetic error: Floating divide by 0
IDL> print, (1-prec.EPSNEG)*prec.IBETA^prec.MAXEXP
      0.0000000
```

There I dont get 'Infinity' as result, but '0.0'.

If, however, I try the same on a UNIX-system, first everything seems ok.

```
IDL> help, /struc, NR_MACHAR(/DOUBLE)
** Structure DMACHAR, 13 tags, length=72:
   IBETA           LONG                 2
   IT              LONG                53
   IRND            LONG                 2
   NGRD            LONG                 0
```

```
MACHEP          LONG             -52
NEGEP           LONG             -53
IEXP            LONG              11
MINEXP          LONG           -1022
MAXEXP          LONG            1024
EPS             DOUBLE       2.2204460e-16
EPSNEG          DOUBLE       1.1102230e-16
XMIN            DOUBLE      2.2250739e-308
XMAX            DOUBLE      1.7976931e+308
```

But - if I calculate XMAX myself again - the result is 0.0:

```
IDL> prec = NR_MACHAR(/DOUBLE)
IDL> print, (1-prec.EPSNEG)*prec.IBETA^prec.MAXEXP
      0.0000000
```

If I use MACHAR() instead of NR_MACHAR(), exactly the same happens.
The keyword DOUBLE also is not the origin of the problem.

Maybe anybody else understands, what happens?

Thanks,
Heiko
--

Heiko Hï¿½nnefeld

HASYLAB / DESY

Notkestr. 85
22603 Hamburg
Tel.: 040 / 8998-2698
Fax.: 040 / 8998-2787
e-mail: hhuenne@desy.de

Subject: Re: Arithmetic error
Posted by Nigel Wade on Thu, 31 Jan 2002 10:11:39 GMT
View Forum Message <> Reply to Message

Ed Wright wrote:

>
> To: IDL
> From: Ed Wright, JPL
>
> I require assistance with an odd problem. My current projects involves
> creating a dlm module of some 200+ routines from a library of 1500+
> routines. Several of the IDL callable routines replicate intrinsic IDL

> calls. While testing one such routine that calculates determinate of a 3X3
> I encountered a problem. The numerical result between my determinant
> function and determn matches to 10^(-16).
>
> Now the odd thing. Under certain circumstances the use of my determinant
> function as an argument in another routine call causes an error:
>
> % Program caused arithmetic error: Floating illegal operand.
>

That error means somewhere in your code you have generated a NaN (Not a Number). The operations which generate NaNs include 0/0, Inf*0, sqrt(<0) Inf-Inf,Inf/Inf (there may be others...).

Since you don't have a preceding 'Floating divide by zero', it's a fair bet your error doesn't involve Inf. So, all you have to do is look through your code to find where you might have a 0/0 or sqrt(<0). ;-)

--

Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK
E-mail : nmw@ion.le.ac.uk
Phone : +44 (0)116 2523568, Fax : +44 (0)116 2523555

Subject: Re: Arithmetic error
Posted by Ed Wright on Thu, 31 Jan 2002 15:53:53 GMT
View Forum Message <> Reply to Message

in article a3b58p$kb2i$1@rook.le.ac.uk, Nigel Wade at nmw@ion.le.ac.uk wrote on 1/31/02 2:11 AM:

> Ed Wright wrote:
>
>>
>> To: IDL
>> From: Ed Wright, JPL
>>
>> I require assistance with an odd problem. My current projects involves
>> creating a dlm module of some 200+ routines from a library of 1500+
>> routines. Several of the IDL callable routines replicate intrinsic IDL
>> calls. While testing one such routine that calculates determinate of a 3X3
>> I encountered a problem. The numerical result between my determinant
>> function and determn matches to 10^(-16).
>>
>> Now the odd thing. Under certain circumstances the use of my determinant

>> function as an argument in another routine call causes an error:
>>
>> % Program caused arithmetic error: Floating illegal operand.
>>
>
> That error means somewhere in your code you have generated a NaN (Not a
> Number). The operations which generate NaNs include 0/0, Inf*0, sqrt(<0)
> Inf-Inf,Inf/Inf (there may be others...).

By code, should I take it to mean the external C code?

>
> Since you don't have a preceding 'Floating divide by zero', it's a fair bet
> your error doesn't involve Inf. So, all you have to do is look through your
> code to find where you might have a 0/0 or sqrt(<0). ;-)

Oh, is that all. No problem. 300k lines - let me at them.... Mr. Codebuster,
that's me.

As always,
Ed Wright

Subject: Re: Arithmetic error
Posted by Paul van Delst on Thu, 31 Jan 2002 18:41:55 GMT
View Forum Message <> Reply to Message

Ed Wright wrote:
>
> in article a3b58p$kb2i$1@rook.le.ac.uk, Nigel Wade at nmw@ion.le.ac.uk wrote
> on 1/31/02 2:11 AM:
>
>> Ed Wright wrote:
>>
>>>
>>> To: IDL
>>> From: Ed Wright, JPL
>>>
>>> I require assistance with an odd problem. My current projects involves
>>> creating a dlm module of some 200+ routines from a library of 1500+
>>> routines. Several of the IDL callable routines replicate intrinsic IDL
>>> calls. While testing one such routine that calculates determinate of a 3X3
>>> I encountered a problem. The numerical result between my determinant
>>> function and determn matches to 10^(-16).
>>>
>>> Now the odd thing. Under certain circumstances the use of my determinant
>>> function as an argument in another routine call causes an error:
>>>

>>> % Program caused arithmetic error: Floating illegal operand.
>>>
>>
>> That error means somewhere in your code you have generated a NaN (Not a
>> Number). The operations which generate NaNs include 0/0, Inf*0, sqrt(<0)
>> Inf-Inf,Inf/Inf (there may be others...).
>
> By code, should I take it to mean the external C code?
>
>>
>> Since you don't have a preceding 'Floating divide by zero', it's a fair bet
>> your error doesn't involve Inf. So, all you have to do is look through your
>> code to find where you might have a 0/0 or sqrt(<0). ;-)
>
> Oh, is that all. No problem. 300k lines - let me at them.... Mr. Codebuster,
> that's me.

Try !EXCEPT=2 and let IDL do the searching for you. Be prepared for a lot of output.....

paulv

--
Paul van Delst Religious and cultural
CIMSS @ NOAA/NCEP purity is a fundamentalist
Ph: (301)763-8000 x7274 fantasy
Fax:(301)763-8545 V.S.Naipaul

Subject: Re: Arithmetic error
Posted by Nigel Wade on Fri, 01 Feb 2002 10:12:10 GMT
View Forum Message <> Reply to Message

Ed Wright wrote:

> in article a3b58p$kb2i$1@rook.le.ac.uk, Nigel Wade at nmw@ion.le.ac.uk
> wrote on 1/31/02 2:11 AM:
>
>> Ed Wright wrote:
>>
>>>
>>> To: IDL
>>> From: Ed Wright, JPL
>>>
>>> I require assistance with an odd problem. My current projects involves
>>> creating a dlm module of some 200+ routines from a library of 1500+
>>> routines. Several of the IDL callable routines replicate intrinsic IDL
>>> calls. While testing one such routine that calculates determinate of a
>>> 3X3 I encountered a problem. The numerical result between my determinant

>>> function and determn matches to 10^(-16).
>>>
>>> Now the odd thing. Under certain circumstances the use of my determinant
>>> function as an argument in another routine call causes an error:
>>>
>>> % Program caused arithmetic error: Floating illegal operand.
>>>
>>
>> That error means somewhere in your code you have generated a NaN (Not a
>> Number). The operations which generate NaNs include 0/0, Inf*0, sqrt(<0)
>> Inf-Inf,Inf/Inf (there may be others...).
>
> By code, should I take it to mean the external C code?
>

It could be either. Both IDL and C would generate the same floating point exception. I can't offhand remember whether an exception in your C code would generate this exception in IDL. But it might be that the C code generates a Nan/Inf and then IDL generates the exception when it tries to use it.

>>
>> Since you don't have a preceding 'Floating divide by zero', it's a fair
>> bet your error doesn't involve Inf. So, all you have to do is look
>> through your code to find where you might have a 0/0 or sqrt(<0). ;-)
>
> Oh, is that all. No problem. 300k lines - let me at them.... Mr.
> Codebuster, that's me.

Debugging external code is always fun.

I forsee extensive use of printf...

--

Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK
E-mail : nmw@ion.le.ac.uk
Phone : +44 (0)116 2523568, Fax : +44 (0)116 2523555