Subject: Re: Interesting Filled Contour Problem
Posted by David Fanning on Mon, 16 Apr 2012 19:28:21 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> Anyone who can explain that Min(data) thing will get an
> automatic invite to the IDL Expert Programming Association
> annual meeting this Fall.

I should point out (for those of you who respond
to incentives) that this is now a $250 value!

Coyote raised the price the other day when he
decided to celebrate Spring with a small Dairy
Queen vanilla cone and he had to pay $2. God
only knows how much this tiny thing would cost
in Edinburgh or Melbourne! :-(

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Interesting Filled Contour Problem
Posted by David Fanning on Mon, 16 Apr 2012 20:57:40 GMT
View Forum Message <> Reply to Message

David Fanning writes:

>> Anyone who can explain that Min(data) thing will get an
>> automatic invite to the IDL Expert Programming Association
>> annual meeting this Fall.
>
> I should point out (for those of you who respond
> to incentives) that this is now a $250 value!

We have a winner, and from a MOST unexpected source!
Josh, a Technical Support Engineer at Excelis, has
correctly identified the problem as being caused

by NaNs in the original data set. When I do this:

  data = data > 0

I apparently set the NaN values to zero, too.
(Not sure this is totally kosher, but it does
seem to work!)

Oddly, enough, the notion of NaNs got me thinking about
a problem I ran into with contour plots in my last
book, and a quick look-though found the note on page
161 that would have alerted me to the problem. Assuming
I had read the book, of course. :-)

All things come around again, I guess, and especially
as you get older. :-)

Thanks, Josh!

Cheers,

David

P.S. The location of your ticket is here: 8d8rw03893wkrh3w3lwr323l.
A secret courier will be in contact with you in the next week or
so to provide the decoder ring that will allow you to decode the
location. Don't tell anyone the location of the ticket or the dinner,
under pain of ..., well, just don't tell anyone. Lord knows if
word of this leaks out, we'll have to get special bouncers to
handle all the women who will want to come!

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

---

Subject: Re: Interesting Filled Contour Problem
Posted by Craig Markwardt on Tue, 17 Apr 2012 03:21:02 GMT
View Forum Message <> Reply to Message

On Monday, April 16, 2012 4:57:40 PM UTC-4, David Fanning wrote:
> We have a winner, and from a MOST unexpected source!
> Josh, a Technical Support Engineer at Excelis, has
> correctly identified the problem as being caused

> by NaNs in the original data set. When I do this:
>
>    data = data > 0
>
> I apparently set the NaN values to zero, too.
> (Not sure this is totally kosher, but it does
> seem to work!)

Whaaa? I was under the impression that
  NAN = !values.d_nan
  NAN OP X
will always result in a value of NAN, for all operations OP and for all operands X. In other words, I don't think IDL's behavior is kosher.

Strangely,
 NAN < 0
produces NAN, so apparently NAN is a negative number (!!!).

Craig

---

Subject: Re: Interesting Filled Contour Problem
Posted by David Fanning on Tue, 17 Apr 2012 03:51:17 GMT
View Forum Message <> Reply to Message

Craig Markwardt writes:

> Whaaa? I was under the impression that
>    NAN = !values.d_nan
>    NAN OP X
> will always result in a value of NAN, for all operations OP and for all operands X. In other words, I don't think IDL's behavior is kosher.
>
> Strangely,
>   NAN < 0
> produces NAN, so apparently NAN is a negative number (!!!).

I guess. I'm as surprised as you are. Inconsistent,
to say the least. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

---

Subject: Re: Interesting Filled Contour Problem
Posted by Carsten Lechte on Tue, 17 Apr 2012 11:54:27 GMT
View Forum Message <> Reply to Message

On 17/04/12 05:21, Craig Markwardt wrote:
> NAN OP X
> will always result in a value of NAN, for all operations OP and for all operands X

Unless that OP is a comparison operator; then the result is always
false, i.e. NAN eq NAN is false etc.

> Strangely,
> NAN< 0
> produces NAN, so apparently NAN is a negative number (!!!).

You jest, but what is happening is this:

```
IDL> print, asdf, asdf GT 0, asdf LT 0, asdf < 0, asdf > 0
      -1.0000000          -NaN           NaN       1.0000000
   0   0   0   1
   1   0   0   0
      -1.0000000          -NaN           NaN       0.0000000
       0.0000000       0.0000000       0.0000000       1.0000000
```

It seems like "asdf < 0" zeros those elements WHERE( asdf gt 0), i.e.
element 3, while "asdf > 0" uses the COMPLEMENT of WHERE( asdf gt 0),
i.e. [0,1,2], instead of zeroing WHERE( asdf LT 0), i.e. [0].

Unexpected, inconsistent, yet conforming to IEEE.

chl

---

Subject: Re: Interesting Filled Contour Problem
Posted by Kenneth P. Bowman on Tue, 17 Apr 2012 16:00:30 GMT
View Forum Message <> Reply to Message

In article < 9436440.509.1334632862216.JavaMail.geo-discussion-forums@vbv d13 >,
Craig Markwardt <craig.markwardt@gmail.com> wrote:

> On Monday, April 16, 2012 4:57:40 PM UTC-4, David Fanning wrote:
>> We have a winner, and from a MOST unexpected source!

>> Josh, a Technical Support Engineer at Excelis, has
>> correctly identified the problem as being caused
>> by NaNs in the original data set. When I do this:
>>
>>  data = data > 0
>>
>> I apparently set the NaN values to zero, too.
>> (Not sure this is totally kosher, but it does
>> seem to work!)
>
> Whaaa? I was under the impression that
>  NAN = !values.d_nan
>  NAN OP X
> will always result in a value of NAN, for all operations OP and for all
> operands X. In other words, I don't think IDL's behavior is kosher.
>
> Strangely,
>  NAN < 0
> produces NAN, so apparently NAN is a negative number (!!!).
>
> Craig

IEEE 754 says this

5.11 Details of comparison predicates 5.1.0

For every supported arithmetic format, it shall be possible to compare one floating-point datum to another in that format (see 5.6.1). Additionally, floating-point data represented in different formats shall be comparable as long

Four mutually exclusive relations are possible: less than, equal, greater than, and unordered. The last case arises when at least one operand is NaN. Every NaN shall compare unordered with everything, including itself. Comparisons shall ignore the sign of zero (so +0 = ?0). Infinite operands of the same sign shall compare equal.

Languages define how the result of a comparison shall be delivered, in one of two ways: either as a relation identifying one of the four relations listed above, or as a true-false response to a predicate that names the specific comparison desired.

Assuming that the floating-point hardware returns an "unordered" for any comparison involving a NaN, the question is, what does IDL do with it?

For a logical comparison, it seems to properly return false

```
IDL> PRINT, 5.0 EQ !Values.F_NaN
   0
IDL> PRINT, 5.0 LT !Values.F_NaN
   0
IDL> PRINT, 5.0 GT !Values.F_NaN
   0
```

The < and > operators, however, mean "return the lesser or greater of" the two arguments. It seems to me that by definition the result is undefined when the relation is "unordered". That is, it is neither less than, greater than, nor equal, and it looks like IDL is inconsistent in its handling of that case. At least it does generate a floating-point exception.

```
IDL> PRINT, 1.0 < !Values.F_NaN
      1.00000
% Program caused arithmetic error: Floating illegal operand
IDL> PRINT, 1.0 > !Values.F_NaN
         NaN
% Program caused arithmetic error: Floating illegal operand
```

Looks like an implementation issue.

Ken

---