Subject: Re: Old Question
Posted by Ben Tupper on Tue, 14 Dec 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Jacques Basson wrote:

> Hi all
>
> Sorry, this has got to be an old question, but I can't seem to locate
> the answer. What is the way around the following problem?
>
> IDL> a = -1
> IDL> print, -1^(1./3)
> -1.00000
> IDL> print, a^(1./3)
> NaN
> % Program caused arithmetic error: Floating illegal operand
>
> Thanks
> Jacques

Hello,

I now know why it happens. In the documentation I see...

Exponentiation

The caret (^) is the exponentiation operator. A^B is equal to A raised to the B power.

ï¿½ If A is a real number and B is of integer type, repeated multiplication is applied.
ï¿½ If A is real and B is real (non-integer), the formula A^B = e^(B ln A) is evaluated.
ï¿½ If A is complex and B is real, the formula A^B = (re^(iq))^B = r^B * (cosBq + isinBq) (where r is the real part of A and iq is the imaginary part) is evaluated.

ï¿½ If B is complex, the formula A^B = e^(B ln A) is evaluated. If A is also complex, the natural logarithm is computed to be ln(A) = ln(re^(iq)) = ln(r) + iq (where r is the real part of A and iq is the imaginary part).
ï¿½ A^0 is defined as 1.

Your example falls into the second type of operation. I don't know how to get around that but would like to know also.

Ben

--
Ben Tupper
Pemaquid River Company
248 Lower Round Pond Road
POB 106
Bristol, ME 04539

Tel: (207) 563-1048
Email: PemaquidRiver@tidewater.net

Subject: Re: Old Question
Posted by thompson on Wed, 15 Dec 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Ben Tupper <pemaquidriver@tidewater.net> writes:

> Jacques Basson wrote:

>> Hi all
>>
>> Sorry, this has got to be an old question, but I can't seem to locate
>> the answer. What is the way around the following problem?
>>
>> IDL> a = -1
>> IDL> print, -1^(1./3)
>> -1.00000
>> IDL> print, a^(1./3)
>> NaN
>> % Program caused arithmetic error: Floating illegal operand
>>
>> Thanks
>> Jacques

Um, I don't know if you realize this, but the only reason that the first
example DOESN'T fail is because

IDL> print, -1^(1./3)

is equivalent to

IDL> print, -(1^(1./3))

In other words, the minus sign doesn't come into play until after the
exponentiation is applied. If you typed in

```
 IDL> print, (-1)^(1./3)
```

you'd get the NaN result, with the error message, as above. Note that you can test this by typing in

```
 IDL> print, -1^2, -(1^2), (-1)^2
      -1      -1       1
```

Of course, as is implied by the answer below, the proper way to evaluate your example would be

```
 IDL> a = -1
 IDL> print, a^complex(1./3, 0)
 (     0.500000,     0.866025)
```

William Thompson

> Hello,

> I now know why it happens. In the documentation I see...

> Exponentiation

> The caret (^) is the exponentiation operator. A^B is equal to A raised to
> the B power.

> * If A is a real number and B is of integer type, repeated multiplication
> is applied.
> * If A is real and B is real (non-integer), the formula A^B = e^(B ln A)
> is evaluated.
> * If A is complex and B is real, the formula A^B = (re^(iq))^B = r^B *
> (cosBq + isinBq) (where r is the real part of A and iq is the imaginary
> part) is evaluated.

> * B is complex, the formula A^B = e^(B ln A) is evaluated. If A is
> also complex, the natural logarithm is computed to be ln(A) = ln(re^(iq))
> = ln(r) + iq (where r is the real part of A and iq is the imaginary
> part).
> * A^0 is defined as 1.

> Your example falls into the second type of operation. I don't know how
> to get around that but would like to know also.

> Ben

> --

> Ben Tupper
> Pemaquid River Company
> 248 Lower Round Pond Road
> POB 106
> Bristol, ME 04539

> Tel: (207) 563-1048
> Email: PemaquidRiver@tidewater.net

Subject: Re: Old Question
Posted by thompson on Wed, 15 Dec 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Jacques Basson <jfb37@mrao.cam.ac.uk> writes:

> Ben Tupper wrote:
>>
>> Jacques Basson wrote:
>>
>>> Hi all
>>>
>>> Sorry, this has got to be an old question, but I can't seem to locate
>>> the answer. What is the way around the following problem?
>>>
>>> IDL> a = -1
>>> IDL> print, -1^(1./3)
>>> -1.00000
>>> IDL> print, a^(1./3)
>>> NaN
>>> % Program caused arithmetic error: Floating illegal operand
>>>
>>> Thanks
>>> Jacques
>>

(stuff deleted)

> I resorted to creating a simple function which basically does
> abs(a)^(1./3) * (2*(a gt 0) - 1)
> Slightly messy, but it works.

> Jacques

I agree that it doesn't generate any errors, but what is its physical or
mathematical meaning? The only justification I can think of for this would be
if negative values of A where physically meaningless, and only represented
measurement error. The above procedure would then preserve the distribution of

noise about zero without introducing any biases towards positive or negative numbers. If that's the case, then I agree that the above procedure is proper.

William Thompson

Subject: Re: Old Question
Posted by Jacques Basson on Wed, 15 Dec 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Ben Tupper wrote:
>
> Jacques Basson wrote:
>
>> Hi all
>>
>> Sorry, this has got to be an old question, but I can't seem to locate
>> the answer. What is the way around the following problem?
>>
>> IDL> a = -1
>> IDL> print, -1^(1./3)
>> -1.00000
>> IDL> print, a^(1./3)
>> NaN
>> % Program caused arithmetic error: Floating illegal operand
>>
>> Thanks
>> Jacques
>
> Hello,
>
> I now know why it happens. In the documentation I see...
>
> Exponentiation
>
> The caret (^) is the exponentiation operator. A^B is equal to A raised to
> the B power.
>
> ï¿½ If A is a real number and B is of integer type, repeated multiplication
> is applied.
> ï¿½ If A is real and B is real (non-integer), the formula A^B = e^(B ln A)
> is evaluated.
> ï¿½ If A is complex and B is real, the formula A^B = (re^(iq))^B = r^B *
> (cosBq + isinBq) (where r is the real part of A and iq is the imaginary
> part) is evaluated.
>
> ï¿½ If B is complex, the formula A^B = e^(B ln A) is evaluated. If A is
> also complex, the natural logarithm is computed to be ln(A) = ln(re^(iq))

> = ln(r) + iq (where r is the real part of A and iq is the imaginary
> part).
> ï¿½ A^0 is defined as 1.
>
> Your example falls into the second type of operation. I don't know how
> to get around that but would like to know also.
>
> Ben
>
> --
> Ben Tupper
> Pemaquid River Company
> 248 Lower Round Pond Road
> POB 106
> Bristol, ME 04539
>
> Tel: (207) 563-1048
> Email: PemaquidRiver@tidewater.net

I resorted to creating a simple function which basically does
abs(a)^(1./3) * (2*(a gt 0) - 1)
Slightly messy, but it works.

Jacques

Subject: Re: Old Question
Posted by Jacques Basson on Thu, 16 Dec 1999 08:00:00 GMT
View Forum Message <> Reply to Message

William Thompson wrote:
>
> Ben Tupper <pemaquidriver@tidewater.net> writes:
>
>> Jacques Basson wrote:
>
>>> Hi all
>>>
>>> Sorry, this has got to be an old question, but I can't seem to locate
>>> the answer. What is the way around the following problem?
>>>
>>> IDL> a = -1
>>> IDL> print, -1^(1./3)
>>> -1.00000
>>> IDL> print, a^(1./3)
>>> NaN
>>> % Program caused arithmetic error: Floating illegal operand
>>>

>>> Thanks
>>> Jacques
>
> Um, I don't know if you realize this, but the only reason that the first
> example DOESN'T fail is because
>
> IDL> print, -1^(1./3)
>
> is equivalent to
>
> IDL> print, -(1^(1./3))
>

My mistake, sorry!

> In other words, the minus sign doesn't come into play until after the
> exponentiation is applied. If you typed in
>
> IDL> print, (-1)^(1./3)
>
> you'd get the NaN result, with the error message, as above. Note that you can
> test this by typing in
>
> IDL> print, -1^2, -(1^2), (-1)^2
> -1 -1 1
>
> Of course, as is implied by the answer below, the proper way to evaluate your
> example would be
>
> IDL> a = -1
> IDL> print, a^complex(1./3, 0)
> (0.500000, 0.866025)
>

I was trying to find out if IDL had some routine for dealing with
rational exponents. If the exponent is irrational then complex numbers
are required as you showed above. In some cases for rational exponents
(as in the case of a^(1./3), a<0), it is possible to get away without
having to use complex numbers. Besides, the point given above (1/2,
sqrt(3)/2) is only 1 of the 3 roots of (-1)^1/3 since (-1)^1/3 =
(exp(pi*i);exp(-pi*i);exp(3pi*i))^1/3
=exp(pi/3*i);exp(-pi/3*i);exp(pi*i)
=(1/2,sqrt(3)/2);(1/2,-sqrt(3)/2);-1

(apologies for the sloppy mathematical layout)

I am sure that the IDL documentation mentions that only the principal
value is returned (I haven't looked that bit up), but in my case, I am

interested in the value that lies on the real axis and not the principal value. It makes sense for IDL to get the principal value, since working out all the values becomes a bit of a pain if you have a^0.0001 or even a^0.333333, which will have(?) to be treated as a^(333333/1000000). So I am not complaing about IDL, it is just the equation that I am working with that is unusual(?), and I thought that maybe IDL could somehow deal with it elegantly.

> William Thompson
>
>> Hello,
>
>> I now know why it happens. In the documentation I see...
>
>> Exponentiation
>
>> The caret (^) is the exponentiation operator. A^B is equal to A raised to
>> the B power.
>
>> * If A is a real number and B is of integer type, repeated multiplication
>> is applied.
>> * If A is real and B is real (non-integer), the formula A^B = e^(B ln A)
>> is evaluated.
>> * If A is complex and B is real, the formula A^B = (re^(iq))^B = r^B *
>> (cosBq + isinBq) (where r is the real part of A and iq is the imaginary
>> part) is evaluated.
>
>> * B is complex, the formula A^B = e^(B ln A) is evaluated. If A is
>> also complex, the natural logarithm is computed to be ln(A) = ln(re^(iq))
>> = ln(r) + iq (where r is the real part of A and iq is the imaginary
>> part).
>> * A^0 is defined as 1.
>
>> Your example falls into the second type of operation. I don't know how
>> to get around that but would like to know also.
>
>> Ben
>
>> --
>> Ben Tupper
>> Pemaquid River Company
>> 248 Lower Round Pond Road
>> POB 106
>> Bristol, ME 04539
>
>> Tel: (207) 563-1048
>> Email: PemaquidRiver@tidewater.net