Subject: Re: Specify the degree of accuracy of a floating point number
Posted by David Fanning on Thu, 22 Feb 2007 06:22:13 GMT
View Forum Message <> Reply to Message

loknath writes:

> Is there a way to convert a single precision or double precision
> floating point number to an accuracy of specified decimal places? In
> other words, if I want to write 0.268954 or 233.256 to an accuracy of
> 2 decimal places, i.e. 0.27 and 233.26, is there a way to do it?

You could try NUMBER_FORMATTER:

http://www.dfanning.com/programs/number_formatter.pro

Cheers,

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

Subject: Re: Specify the degree of accuracy of a floating point number
Posted by Allan Whiteford on Thu, 22 Feb 2007 13:05:44 GMT
View Forum Message <> Reply to Message

loknath wrote:
> Hi
> Is there a way to convert a single precision or double precision
> floating point number to an accuracy of specified decimal places? In
> other words, if I want to write 0.268954 or 233.256 to an accuracy of
> 2 decimal places, i.e. 0.27 and 233.26, is there a way to do it?
> Thanks
>

Loknath,

As well as David's suggestion you could look at the format keyword of
string() and print.

e.g.

```
IDL> number=0.268954
IDL> str=string(number,format='(F0.2)')
IDL> print,str
```

0.27
IDL> print,number,format='(F0.2)'
0.27

similarly:

IDL> number=233.256
IDL> str=string(number,format='(F0.2)')
IDL> print,str
233.26
IDL> print,number,format='(F0.2)'
233.26

Note that support for F0.N is fairly recent (doesn't work in 6.0, does work in 6.2 - don't have 6.1 handy to test).

Thanks,

Allan

Subject: Re: Specify the degree of accuracy of a floating point number
Posted by Brian Larsen on Thu, 22 Feb 2007 17:58:22 GMT
View Forum Message <> Reply to Message

Just to add more to the mix here:

I use the solarsoft routine round_off.pro (I will put it inline as its short) to do this, seems to work well and its already written, which I always like.

Brian

```
;+
; Project     : SOHO - CDS
;
; Name        : ROUND_OFF()
;
; Purpose     : To round a number to a specified accuracy
;
; Explanation : Rounds the input number to the accuracy specified.
;
; Use         : IDL> out = round_off(x,acc)
;               IDL> out = round_off(12.278,0.01) ==> out = 12.28
;
; Inputs      : x    - the number to be operated on
;               acc  - the numerical accuracy required
```

```
;
; Opt. Inputs : None
;
; Outputs     : The function value is the input rounded to the desired
accuracy
;
; Opt. Outputs: None
;
; Keywords    : None
;
; Calls       : NINT
;
; Restrictions: None
;
; Side effects: None
;
; Category    : Util, Numerical
;
; Prev. Hist. : None
;
; Written     : C D Pike, RAL, 17-May-1993
;
; Modified    : Replace calls to NINT by ROUND.  CDP, 17-Jun-95
;
; Version     : Version 2, 17-Jun-95
;-

function round_off,num,acc

;
;  check enough parameters
;
if n_params() lt 2 then begin
   bell
   print,'Use: num = round_off(input, accuracy)'
   return,0.0
endif

;
;  check for negative input
;
if num lt 0.0 then neg=1 else neg=0

;
;  round off according to the accuracy required
;
num = abs(num)
if acc lt 1.0 then begin
```

```
   factor = round(1.0/acc)
   x = long((num+0.5*acc)*factor)/double(factor)
endif else begin
   x = long((long((num+acc/2.)/acc))*acc)
endelse

;
;  return in same form received
;
case datatype(num) of
   'BYT': x = byte(x)
   'INT': x = fix(x)
   'LON': x = long(x)
   'DOU': x = double(x)
   'FLO': x = float(x)
endcase

if neg then return,-x else return,x

end
```


 ---------- ---------
Brian A. Larsen
Dept. of Physics
Space Science and Engineering Lab (SSEL)
Montana State University - Bozeman
Bozeman, MT 59717

Subject: Re: Specify the degree of accuracy of a floating point number
Posted by David Fanning on Thu, 22 Feb 2007 18:28:30 GMT
View Forum Message <> Reply to Message

Brian Larsen writes:

> Just to add more to the mix here:
>
> I use the solarsoft routine round_off.pro (I will put it inline as its
> short) to do this, seems to work well and its already written, which I
> always like.

To get this to work (I don't have the DATATYPE function), I changed
this case statement:

```
>  case datatype(num) of
>     'BYT': x = byte(x)
>     'INT': x = fix(x)
>     'LON': x = long(x)
>     'DOU': x = double(x)
>     'FLO': x = float(x)
>  endcase
```

To this:

```
case Size(num, /TName) of
   'BYTE': x = byte(x)
   'INT': x = fix(x)
   'LONG': x = long(x)
   'DOUBLE': x = double(x)
   'FLOAT': x = float(x)
endcase
```

A pretty harmless change, it seems to me. Then I tried the
program:

```
IDL> a = 432.49584738273845D
IDL> print, round_off(a, 0.0000001)
      2.9991178
```

Huh!? What did I do wrong?

Cheers,

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

Subject: Re: Specify the degree of accuracy of a floating point number
Posted by loknath on Thu, 22 Feb 2007 18:57:15 GMT
View Forum Message <> Reply to Message

On Feb 22, 11:28 am, David Fanning <n...@dfanning.com> wrote:
> Brian Larsen writes:
>> Just to add more to the mix here:

>
>> I use the solarsoft routine round_off.pro (I will put it inline as its
>> short) to do this, seems to work well and its already written, which I
>> always like.
>
> To get this to work (I don't have the DATATYPE function), I changed
> this case statement:
>
>> case datatype(num) of
>>   'BYT': x = byte(x)
>>   'INT': x = fix(x)
>>   'LON': x = long(x)
>>   'DOU': x = double(x)
>>   'FLO': x = float(x)
>> endcase
>
> To this:
>
> case Size(num, /TName) of
>   'BYTE': x = byte(x)
>   'INT': x = fix(x)
>   'LONG': x = long(x)
>   'DOUBLE': x = double(x)
>   'FLOAT': x = float(x)
> endcase
>
> A pretty harmless change, it seems to me. Then I tried the
> program:
>
> IDL> a = 432.49584738273845D
> IDL> print, round_off(a, 0.0000001)
>   2.9991178
>
> Huh!? What did I do wrong?
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

thank you,
all of you for the great suggestions.
Loknath

Subject: Re: Specify the degree of accuracy of a floating point number
Posted by Brian Larsen on Thu, 22 Feb 2007 20:43:44 GMT
View Forum Message <> Reply to Message

David,

hmmm...

>
> IDL> a = 432.49584738273845D
> IDL> print, round_off(a, 0.0000001)
> 2.9991178
>
> Huh!? What did I do wrong?

This is what it does for me (and I do have datatype)

```
IDL> a = 432.49584738273845D
IDL>  print, round_off(a, 0.0000001)
% Compiled module: ROUND_OFF.
     -214.74836
% Program caused arithmetic error: Floating illegal operand
IDL>  print, round_off(a, 0.0000001d)
     -214.74836
% Program caused arithmetic error: Floating illegal operand
IDL> help, a
A               DOUBLE    =       432.49585
IDL>  print, round_off(a, 0.0001)
      432.49580
```

I think the answer is in the warning (% Program caused arithmetic error: Floating illegal operand)
look at this chunk of the code:

```
num = abs(num)
if acc lt 1.0 then begin
  factor = round(1.0/acc)
  x = long((num+0.5*acc)*factor)/double(factor)
endif else begin
  x = long((long((num+acc/2.)/acc))*acc)
endelse
;; and add in a help statement
HELP, x, acc
```

I then get:

```
IDL> a = 432.49584738273845D
IDL> print, round_off(a, 0.0000001)
X               DOUBLE    =      -214.74836
ACC               FLOAT     =   1.00000e-07
```

-214.74836
% Program caused arithmetic error: Floating illegal operand

Is it something to do with dividing a long by a double??
x = long((num+0.5*acc)*factor)/double(factor)

Or something to do with?
http://www.dfanning.com/math_tips/sky_is_falling.html

Odd that it works with smaller number of digits...
IDL> print, round_off(a, 0.000001)
X DOUBLE = 432.49585
ACC FLOAT = 1.00000e-06
432.49585
IDL> print, round_off(a, 0.0000001)
X DOUBLE = -214.74836
ACC FLOAT = 1.00000e-07
-214.74836

I bet you are in a better position to understand this than I am
however...

Brian

---------- ---------
Brian A. Larsen
Dept. of Physics
Space Science and Engineering Lab (SSEL)
Montana State University - Bozeman
Bozeman, MT 59717

Subject: Re: Specify the degree of accuracy of a floating point number
Posted by David Fanning on Thu, 22 Feb 2007 21:18:19 GMT
View Forum Message <> Reply to Message

Brian Larsen writes:

> I bet you are in a better position to understand this than I am
> however...

Well, what I know about it is that formatting numbers in
a sensible way for display (in widget programs, for example)

is EXTREMELY annoying. And double precision numbers are
perverse. (Although not usually as perverse as with your
ROUND_OFF program. I find it amusing, however, that it barfed
so convincingly on my very first try!)

NUMBER_FORMATTER is not too bad, but even there I have
to look at it cross-eyed occasionally to maintain the
fiction that it is a beautiful program. :-)

There are a couple of articles on my web page addressing
the issue for the more curious among you.

Cheers,

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

Subject: Re: Specify the degree of accuracy of a floating point number
Posted by tochaf on Fri, 23 Feb 2007 10:01:56 GMT
View Forum Message <> Reply to Message

> thank you,
> all of you for the great suggestions.
> Loknath

so, can I know about your result? Now you have 0.268954 in 0.27 in
float format to make some aritmetics operations??? . My question is...
HOW?

thank you...

Subject: Re: Specify the degree of accuracy of a floating point number
Posted by Kenneth Bowman on Fri, 23 Feb 2007 16:43:39 GMT
View Forum Message <> Reply to Message

In article <MPG.20478eaa542e4575989e91@news.frii.com>,
David Fanning <news@dfanning.com> wrote:

> A pretty harmless change, it seems to me. Then I tried the

> program:
>
> IDL> a = 432.49584738273845D
> IDL> print, round_off(a, 0.0000001)
>   2.9991178
>
> Huh!? What did I do wrong?

This is a round-off error issue. The function works by multiplying the input by a factor, rounding to a LONG, and then dividing by the same factor to convert back to a floating-point type.

When you are trying to preserve a lot of precision (digits), you get round-off issues with the LONG. This can be avoided (in most cases), by rounding to a LONG64.

See examples below

IDL> a = 432.49584738273845D

This works for small precision
IDL> print, DOUBLE(ROUND(100.0D0*a))/100.0
      432.50000

This fails for large precision
IDL> print, DOUBLE(ROUND(100000000.0D0*a))/100000000.0
      21.474836
% Program caused arithmetic error: Floating illegal operand

But works if 64-bit integers are used
IDL> print, DOUBLE(ROUND(100000000.0D0*a, /L64))/100000000.0
      432.49585

Cheers, Ken