Subject: Re: Yet again, The Sky is Falling!
Posted by David Fanning on Thu, 08 Mar 2007 18:22:37 GMT
View Forum Message <> Reply to Message

yp writes:

- > Why is such discrepancy? In my problem the accuracy after 3rd decimal
- > point is not so important, however, after seeing the results I lose
- > confidence on IDL's capability on Real number arithmetic!

>

> May be I am missing something?

Well, maybe because I can't see it, but I'm immediately suspicious of what is going on in OPERATION. If you perform these two calls in the opposite order do you get the same result? That is, do you know for a fact that A, B, and F are not changing? (You have compared them before and after?)

If it was some other number, perhaps, but zero!? It seems to me all computers can represent 0 accurately. :-)

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: Yet again, The Sky is Falling!
Posted by Paul Van Delst[1] on Thu, 08 Mar 2007 18:24:56 GMT
View Forum Message <> Reply to Message

## yp wrote:

- > This is yet another floating point mystery, and I am unable to figure
- > out which is the right way to deal:

>

- > I wrote this program ("Operation") in which I made sure that all
- > calculations are done in double precision. The program accepts 6
- > mandatory arguments and returns the output to "result".

>

> Syntax: Operation, A, B, C, D, E, F, result

>

- > I get variable results (after 3rd decimal point) when I pass some of
- > the arguments as numbers and when I pass the same arguments as pre-

```
defined variables.
>
> Case#1:
> IDL> Operation, A, B, 0.0D, 0.0D, 0.0D, F, result
> IDL> help, result
> RESULT
                  DOUBLE
                             = Array[7]
> IDL> print, result
> 1.0247013
                1.0279051
                               1.0365066
                                             1.0447064
> 1.0477210
                1.0543893
                               1.0569390
>
 Case#2:
> IDL> C = (D = (E = 0.0D))
> IDL> Operation, A, B, C, D, E, F, result
> IDL> help, result
> RESULT
                  DOUBLE = Array[7]
> IDL> print, result
> 1.0250284
                               1.0367149
                1.0281385
                                             1.0450368
> 1.0480349
                1.0547703
                               1.0573193
> Why is such discrepancy? In my problem the accuracy after 3rd decimal
> point is not so important, however, after seeing the results I lose
> confidence on IDL's capability on Real number arithmetic!
> May be I am missing something?
Eliminating the most obvious possible problem:
Were A, B, or F modified in the first call and then not reinitialised before the second?
cheers.
paulv
Paul van Delst
                     Ride lots.
CIMSS @ NOAA/NCEP/EMC
                                     Eddy Merckx
```

Subject: Re: Yet again, The Sky is Falling!
Posted by Paul Van Delst[1] on Thu, 08 Mar 2007 18:27:08 GMT
View Forum Message <> Reply to Message

David Fanning wrote: > yp writes: >> Why is such discrepancy? In my problem the accuracy after 3rd decimal >> point is not so important, however, after seeing the results I lose >> confidence on IDL's capability on Real number arithmetic! >> >> May be I am missing something? > Well, maybe because I can't see it, but I'm immediately > suspicious of what is going on in OPERATION. If you > perform these two calls in the opposite order do you > get the same result? That is, do you know for a fact > that A, B, and F are not changing? (You have compared > them before and after?) > > If it was some other number, perhaps, but zero!? It seems > to me all computers can represent 0 accurately. :-) Good point. Same for other simple numbers, like 1.0. I even think you could replace "accurately" with "exactly" in your above statement and not be wrong (in a computer-y, hand-wavy sense if nothing else :o) paulv Paul van Delst Ride lots.

Eddy Merckx

Subject: Re: Yet again, The Sky is Falling! Posted by yp on Thu, 08 Mar 2007 19:11:42 GMT

View Forum Message <> Reply to Message

CIMSS @ NOAA/NCEP/EMC

On Mar 8, 6:22 pm, David Fanning <n...@dfanning.com> wrote:

> yp writes:

>

- >> Why is such discrepancy? In my problem the accuracy after 3rd decimal
- >> point is not so important, however, after seeing the results I lose
- >> confidence on IDL's capability on Real number arithmetic!
- >> May be I am missing something?
- > Well, maybe because I can't see it, but I'm immediately
- > suspicious of what is going on in OPERATION. If you
- > perform these two calls in the opposite order do you
- > get the same result? That is, do you know for a fact
- > that A, B, and F are not changing? (You have compared
- > them before and after?)

```
If it was some other number, perhaps, but zero!? It seems
to me all computers can represent 0 accurately. :-)
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.")
```

Thanks David, for your suggestion. I am pretty sure that none of the argunet values change before or after the "Operation". And yes, the discrepancy occurs both ways...

Here is the section from my running script.

```
·_____
PRO test brdf
wave = [412.5, 442.5, 490., 510., 560., 620., 660.]; A (static)
nwave = n_elements(wave) ;B (static)
               :C (static)
sza = 45.0D
vza = 1.078D ;D (static)
dphi = 0.0D
              ;E (static)
              ;F (static)
chl = 0.03D
null = 0.0D
print, 'BEFORE: ', wave, nwave, sza, vza, dphi, chl
foq = (foq0 = (dblarr(nwave)))
 for i=0, n_elements(chl)-1 do begin
  int LUT, wave, nwave, 0.0D, 0.0D, 0.0D, chl[i], fog0 ;Case1
  ; int_LUT, wave, nwave, null, null, null, chl[i], fog0 ;Case2
  int LUT, wave, nwave, sza[i], vza[i], dphi[i], chl[i], fog
  print, 'AFTER: ', wave, nwave, sza, vza, dphi, chl
  help,BRDF
  print,'BRDF: ',double(foq0[*]) / double(foq[*])
 endfor
END
:-----
#1
```

-----

IDL> test\_brdf

BEFORE: 412.500 442.500 490.000 510.000

560.000 620.000 660.000

7 45.000000 1.0780000 0.00000000

0.030000000

Loading f/Q table

AFTER: 412.500 442.500 490.000 510.000

560.000 620.000 660.000

7 45.000000 1.0780000 0.00000000

0.030000000

f/Q: 0.087899996 0.092399998 0.10349999

0.10879999 0.11449999 0.11319999 0.11339999

BRDF: 1.0250284 1.0281385 1.0367150

FOQ DOUBLE = Array[7]

-----

#2

-----

IDL> test\_brdf

BEFORE: 412.500 442.500 490.000 510.000

560.000 620.000 660.000

7 45.000000 1.0780000 0.000000000

0.030000000

Loading f/Q table

AFTER: 412.500 442.500 490.000 510.000

560.000 620.000 660.000

7 45.000000 1.0780000 0.00000000

0.030000000

f/Q: 0.087899996 0.092399998 0.10349999

0.10879999 0.11449999 0.11319999 0.11339999

BRDF: 1.0247013 1.0279051 1.0365066

FOQ DOUBLE = Array[7]

-----

In my previous example, "Operation" = int\_LUT and it does not change

any of the variables during execution or after. I don't suspect that anything wrong happening inside "int\_LUT". For any one case and for same combination of the arguments:- if I run the code for several times, I get same and consistent result each time. But when I switch between passing the argument by value and by variable, I see the discrepancy. Weird!

Subject: Re: Yet again, The Sky is Falling! Posted by yp on Thu, 08 Mar 2007 19:18:25 GMT View Forum Message <> Reply to Message

```
On Mar 8, 7:11 pm, "yp" < Yaswant.Prad...@gmail.com> wrote:
> On Mar 8, 6:22 pm, David Fanning <n...@dfanning.com> wrote:
>
>
>
>
>> yp writes:
>>> Why is such discrepancy? In my problem the accuracy after 3rd decimal
>>> point is not so important, however, after seeing the results I lose
>>> confidence on IDL's capability on Real number arithmetic!
>>> May be I am missing something?
>> Well, maybe because I can't see it, but I'm immediately
>> suspicious of what is going on in OPERATION. If you
>> perform these two calls in the opposite order do you
>> get the same result? That is, do you know for a fact
>> that A, B, and F are not changing? (You have compared
>> them before and after?)
>> If it was some other number, perhaps, but zero!? It seems
>> to me all computers can represent 0 accurately. :-)
>> 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.")
>
> Thanks David, for your suggestion. I am pretty sure that none of the
> argunet values change before or after the "Operation". And yes, the
> discrepancy occurs both ways...
```

```
>
> Here is the section from my running script.
 :-----
 PRO test brdf
> wave = [412.5, 442.5, 490., 510., 560., 620., 660.]
> nwave = n_elements(wave)
                                                 ;B (static)
> sza = 45.0D
                                     ;C (static)
> vza = 1.078D
                                     ;D (static)
> dphi = 0.0D
                                    ;E (static)
                                    ;F (static)
> chl = 0.03D
> \text{null} = 0.0D
>
> print, 'BEFORE: ', wave, nwave, sza, vza, dphi, chl
> foq = (foq0 = (dblarr(nwave)))
>
   for i=0, n_elements(chl)-1 do begin
>
    int LUT, wave, nwave, 0.0D, 0.0D, 0.0D, chl[i], fog0
                                                            :Case1
>
    ; int_LUT, wave, nwave, null, null, null, chl[i], foq0
                                                         ;Case2
>
    int_LUT, wave, nwave, sza[i], vza[i], dphi[i], chl[i], foq
>
>
    print, 'AFTER: ', wave, nwave, sza, vza, dphi, chl
>
    help,BRDF
>
>
    print,'BRDF: ',double(foq0[*]) / double(foq[*])
>
   endfor
>
  END
  ;-----
>
 #1
> IDL> test_brdf
 BEFORE:
                412.500
                            442.500
                                       490.000
                                                   510.000
> 560.000
              620.000
                          660.000
              45.000000
                             1.0780000
                                           0.00000000
         7
 0.030000000
>
  Loading f/Q table
> AFTER:
              412.500
                          442.500
                                      490.000
                                                 510.000
  560.000
              620.000
                          660.000
         7
              45.000000
                             1.0780000
                                           0.00000000
>
> 0.030000000
>
> f/Q:
         0.087899996
                         0.092399998
                                          0.10349999
> 0.10879999
                 0.11449999
                                 0.11319999
                                                0.11339999
>
```

```
> BRDF:
            1.0250284
                         1.0281385
                                     1.0367150
1.0547704
                                        1.0573193
> FOQ
             DOUBLE = Array[7]
>
> #2
  -----
 IDL> test_brdf
>
 BEFORE:
              412.500
                        442.500
                                  490.000
                                            510.000
 560.000
            620.000
                      660.000
        7
            45.000000
                         1.0780000
                                     0.00000000
>
 0.030000000
>
  Loading f/Q table
>
                      442.500
                                 490.000
 AFTER:
            412.500
                                           510.000
  560.000
            620,000
                      660.000
        7
            45.000000
                         1.0780000
                                     0.00000000
  0.030000000
>
> f/Q:
        0.087899996
                      0.092399998
                                    0.10349999
  0.10879999
               0.11449999
                            0.11319999
                                          0.11339999
>
 BRDF:
            1.0247013
                         1.0279051
                                     1.0365066
1.0569390
> FOQ
             DOUBLE = Array[7]
>
> In my previous example, "Operation" = int_LUT and it does not change
> any of the variables during execution or after. I don't suspect that
> anything wrong happening inside "int_LUT". For any one case and for
> same combination of the arguments:- if I run the code for several
> times, I get same and consistent result each time. But when I switch
> between passing the argument by value and by variable, I see the
 discrepancy. Weird!- Hide quoted text -
>
> - Show quoted text -
... ooops! forgot to add the main culprit:
Case#1
FOQ0 = 0.090099994
                     0.094999995
                                   0.10729999
0.11370000
             0.11999999
                          0.11940000
                                        0.11990000
Case#2
FOQ0 = 0.090071241
                     0.094978428
                                   0.10727842
0.11366406
             0.11996405
                          0.11935687
                                       0.11985687
```

Subject: Re: Yet again, The Sky is Falling! Posted by David Fanning on Thu, 08 Mar 2007 19:27:32 GMT

View Forum Message <> Reply to Message

### yp writes:

- > In my previous example, "Operation" = int\_LUT and it does not change
- > any of the variables during execution or after. I don't suspect that
- > anything wrong happening inside "int\_LUT". For any one case and for
- > same combination of the arguments:- if I run the code for several
- > times, I get same and consistent result each time. But when I switch
- > between passing the argument by value and by variable, I see the
- > discrepancy. Weird!

# Extremely weird!

Just to humor me, could you please subscript your A, B, and F variables, so they are passed by value rather than by reference:

IDL> Operation, a[0], b[0], c, d, e, f[0], result

and tell us what happens both ways then? What if all parameters are passed by value?

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: Yet again, The Sky is Falling!
Posted by news.qwest.net on Thu, 08 Mar 2007 19:30:34 GMT
View Forum Message <> Reply to Message

"yp" <Yaswant.Pradhan@gmail.com> wrote in message news:1173381505.441539.306710@c51g2000cwc.googlegroups.com...

...

>> sza = 45.0D ;C (static) >> vza = 1.078D ;D (static) >> dphi = 0.0D ;E (static)

- >> int\_LUT, wave, nwave, 0.0D, 0.0D, 0.0D, chl[i], foq0 ;Case1
- >> int\_LUT, wave, nwave, sza[i], vza[i], dphi[i], chl[i], foq

```
> Case#1
> FOQ0 = 0.090099994
                       0.094999995
                                      0.10729999
 0.11370000
               0.11999999
                             0.11940000
                                          0.11990000
> Case#2
> FOQ0 = 0.090071241
                       0.094978428
                                      0.10727842
> 0.11366406
               0.11996405
                            0.11935687
                                          0.11985687
```

I am jumping in late, and don't really follow what you think the problem is.

BUT, in 'case 1 you are passing zeros, and in case 2 you are passing 45,1.078, 0.

Cheers, bob

Subject: Re: Yet again, The Sky is Falling! Posted by yp on Thu, 08 Mar 2007 19:58:31 GMT

View Forum Message <> Reply to Message

```
On Mar 8, 7:30 pm, "R.G. Stockwell" <n...@email.please> wrote:
 "yp" <Yaswant.Prad...@gmail.com> wrote in message
>
  news:1173381505.441539.306710@c51g2000cwc.googlegroups.com...
>
>
>
                                      ;C (static)
>>> sza = 45.0D
>>> vza = 1.078D
                                      ;D (static)
>>> dphi = 0.0D
                                      ;E (static)
       int_LUT, wave, nwave, 0.0D, 0.0D, 0.0D, chl[i], foq0
                                                            :Case1
       int_LUT, wave, nwave, sza[i], vza[i], dphi[i], chl[i], foq
>>>
>> Case#1
>> FOQ0 = 0.090099994
                           0.094999995
                                           0.10729999
>> 0.11370000
                  0.11999999
                                 0.11940000
                                                0.11990000
>
>> Case#2
>> FOQ0 = 0.090071241
                           0.094978428
                                           0.10727842
>> 0.11366406
                  0.11996405
                                 0.11935687
                                                0.11985687
> I am jumping in late, and don't really follow what you think the
> problem is.
```

```
> BUT, in 'case 1 you are passing zeros, and in case 2 you are passing
> 45,1.078, 0.
> Cheers.
> bob
Hi Bob.
> BUT, in 'case 1 you are passing zeros, and in case 2 you are passing
> 45.1.078.0.
Not really.
In Case 1 the arguments are passed by value (which is shown in the
code)
in Case 2 (comment the line left to Case1 and uncomment the line left
to Case2) the arguments are passed by regerence.
Subject: Re: Yet again, The Sky is Falling!
Posted by yp on Thu, 08 Mar 2007 20:02:54 GMT
View Forum Message <> Reply to Message
On Mar 8, 7:27 pm, David Fanning <n...@dfanning.com> wrote:
> yp writes:
>> In my previous example, "Operation" = int LUT and it does not change
>> any of the variables during execution or after. I don't suspect that
>> anything wrong happening inside "int_LUT". For any one case and for
>> same combination of the arguments:- if I run the code for several
>> times, I get same and consistent result each time. But when I switch
>> between passing the argument by value and by variable. I see the
>> discrepancy. Weird!
> Extremely weird!
>
  Just to humor me, could you please subscript your A, B, and F
  variables, so they are passed by value rather than by reference:
>
    IDL> Operation, a[0], b[0], c, d, e, f[0], result
>
>
  and tell us what happens both ways then? What if all parameters
  are passed by value?
>
  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.")

Hi David,

IDL> Operation, a[0], b[0], c, d, e, f[0], result

This will not work as b = n\_elements(a)
So if I subscript A as a[0], I must set b[0]=1

Anyway, will give it a go with a[0], f[0] and b=1

Subject: Re: Yet again, The Sky is Falling! Posted by yp on Thu, 08 Mar 2007 20:22:22 GMT

View Forum Message <> Reply to Message

```
On Mar 8, 7:27 pm, David Fanning <n...@dfanning.com> wrote:
> vp writes:
>> In my previous example, "Operation" = int_LUT and it does not change
>> any of the variables during execution or after. I don't suspect that
>> anything wrong happening inside "int_LUT". For any one case and for
>> same combination of the arguments:- if I run the code for several
>> times, I get same and consistent result each time. But when I switch
>> between passing the argument by value and by variable, I see the
>> discrepancy. Weird!
> Extremely weird!
>
> Just to humor me, could you please subscript your A, B, and F
 variables, so they are passed by value rather than by reference:
>
    IDL> Operation, a[0], b[0], c, d, e, f[0], result
>
>
> and tell us what happens both ways then? What if all parameters
  are passed by value?
>
>
> 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.")
```

David,

```
result1=(result2=(result3=(dblarr(1))))
int_LUT,wave[0],1,0.0D,0.0D,0.0D,chl[0],result1
int_LUT,wave[0],1,null,null,null,chl[0],result2
int LUT,412.5,1,0.0D,0.0D,0.0D,0.03D,result3
print,'RESULT 1: ', double(result1[*])
print,'RESULT 3: ', double(result2[*])
print, 'RESULT 3: ', double(result3[*])
RESULT 1:
              0.090099994
RESULT 3:
              0.090071241
RESULT 3:
              0.090099994
Subject: Re: Yet again, The Sky is Falling!
Posted by Paul Van Delst[1] on Thu, 08 Mar 2007 20:22:57 GMT
View Forum Message <> Reply to Message
yp wrote:
> On Mar 8, 6:22 pm, David Fanning <n...@dfanning.com> wrote:
>> yp writes:
>>> Why is such discrepancy? In my problem the accuracy after 3rd decimal
>>> point is not so important, however, after seeing the results I lose
>>> confidence on IDL's capability on Real number arithmetic!
>>> May be I am missing something?
>> Well, maybe because I can't see it, but I'm immediately
>> suspicious of what is going on in OPERATION. If you
>> perform these two calls in the opposite order do you
>> get the same result? That is, do you know for a fact
>> that A, B, and F are not changing? (You have compared
>> them before and after?)
>>
>> If it was some other number, perhaps, but zero!? It seems
>> to me all computers can represent 0 accurately. :-)
>>
>> 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.")
```

- > Thanks David, for your suggestion. I am pretty sure that none of the
- > argunet values change before or after the "Operation". And yes, the

>

```
> Here is the section from my running script.
>
> ;-----
> PRO test_brdf
>
> wave = [412.5, 442.5, 490., 510., 560., 620., 660.]; A (static)
> nwave = n elements(wave)
                                ;B (static)
> sza = 45.0D
                 ;C (static)
> vza = 1.078D
                 ;D (static)
                 ;E (static)
> dphi = 0.0D
> chl = 0.03D
                ;F (static)
> null = 0.0D
> print, 'BEFORE: ', wave, nwave, sza, vza, dphi, chl
> foq = (foq0 = (dblarr(nwave)))
>
   for i=0, n elements(chl)-1 do begin
>
    int_LUT, wave, nwave, 0.0D, 0.0D, 0.0D, chl[i], foq0 ;Case1
>
    ; int_LUT, wave, nwave, null, null, null, chl[i], foq0 ;Case2
>
    int LUT, wave, nwave, sza[i], vza[i], dphi[i], chl[i], fog
>
>
    print, 'AFTER: ', wave, nwave, sza, vza, dphi, chl
>
>
    help,BRDF
>
    print,'BRDF: ',double(foq0[*]) / double(foq[*])
why are you using DOUBLE in the above line?
It seems all your numbers are exactly the same until you print the resultant foq0/foq. All
the (internal to int_LUN I assume) f/Q numbers are the same.
Why not just do,
   print, 'BRDF: ', foq0/foq
?
Paul van Delst
                      Ride lots.
CIMSS @ NOAA/NCEP/EMC
                                       Eddy Merckx
Ph: (301)763-8000 x7748
Fax:(301)763-8545
```

Subject: Re: Yet again, The Sky is Falling!

> discrepancy occurs both ways...

View Forum Message <> Reply to Message

Paul,

> why are you using DOUBLE in the above line?

I know its of no use as both foq0 and foq are returned as double, but just in case...

- > It seems all your numbers are exactly the same until you print the resultant foq0/foq. All
- > the (internal to int\_LUN I assume) f/Q numbers are the same.

More precisely, it shows the difference for foq0 (a hypothetical situation) when the arguments are passed by value and by reference.

```
> Why not just do,> print, 'BRDF: ',foq0/foq> ?Not really helpful! Same result.
```

Subject: Re: Yet again, The Sky is Falling!
Posted by news.qwest.net on Thu, 08 Mar 2007 22:34:26 GMT
View Forum Message <> Reply to Message

```
"yp" <Yaswant.Pradhan@gmail.com> wrote in message news:1173383911.537437.293170@h3g2000cwc.googlegroups.com...
>>>> sza = 45.0D ;C (static)

>>>> int_LUT, wave, nwave, 0.0D, 0.0D, 0.0D, chl[i], foq0 ;Case1
>>>> int_LUT, wave, nwave, sza[i], vza[i], dphi[i], chl[i], foq
...
> Not really.
> In Case 1 the arguments are passed by value (which is shown in the code)
> in Case 2 (comment the line left to Case1 and uncomment the line left to Case2) the arguments are passed by regerence.
```

I still do not follow what you mean. If you look at the three lines above, you have sza = 45.0D.

Then you execute the following 2 procedure calls: IDL> int\_LUT, wave, nwave, 0.0D, 0.0D, 0.0D, chl[i], foq0 the next call you have shown is:

IDL> int\_LUT, wave, nwave, sza[i], vza[i], dphi[i], chl[i], foq

In the first case, the 3rd parameter is 0.0D. In the second case, the 3rd parameter is sza[0] = 45.0D.

Anyways, I assume what you actually are doing is something different than what is shown in the messages you posted.

Why not reduce this problem to a very short example, and post the code here so we can take a look at it.

Cheers, bob

Subject: Re: Yet again, The Sky is Falling! Posted by yp on Thu, 08 Mar 2007 22:57:42 GMT View Forum Message <> Reply to Message

```
On Mar 8, 10:34 pm, "R.G. Stockwell" <n...@email.please> wrote:
> "yp" <Yaswant.Prad...@gmail.com> wrote in message
>
  news:1173383911.537437.293170@h3q2000cwc.googlegroups.com...
>
>
>>>> >  sza = 45.0D
                                          :C (static)
          int_LUT, wave, nwave, 0.0D, 0.0D, 0.0D, chl[i], foq0
                                                                  :Case1
          int_LUT, wave, nwave, sza[i], vza[i], dphi[i], chl[i], foq
>>>> >
> ...
>> Not really.
>> In Case 1 the arguments are passed by value (which is shown in the
>> code)
>> in Case 2 (comment the line left to Case1 and uncomment the line left
>> to Case2) the arguments are passed by regerence.
> I still do not follow what you mean. If you look at the three lines
  above, you have sza = 45.0D.
>
> Then you execute the following 2 procedure calls:
> IDL> int_LUT, wave, nwave, 0.0D, 0.0D, 0.0D, chl[i], foq0
> the next call you have shown is:
> IDL> int_LUT, wave, nwave, sza[i], vza[i], dphi[i], chl[i], foq
> In the first case, the 3rd parameter is 0.0D.
> In the second case, the 3rd parameter is sza[0] = 45.0D.
```

```
>
> Anyways, I assume what you actually are doing is something
> different than what is shown in the messages you posted.
> Why not reduce this problem to a very short example, and
> post the code here so we can take a look at it.
>
> Cheers,
> bob
Hi Bob.
Sorry about the confusion. Here I go again...
null=0.0D
int_LUT, 412.5, 1, 0.0D, 0.0D, 0.0D, 0.03D, result1
int_LUT, 412.5, 1, null, null, null, 0.03D, result2
print,'RESULT 1: ', (result1[*])
print,'RESULT 2: ', (result2[*])
RESULT 1:
              0.090099994
RESULT 2:
              0.090071241
I hope this is clearer now.
Subject: Re: Yet again, The Sky is Falling!
Posted by news.gwest.net on Thu, 08 Mar 2007 23:04:40 GMT
View Forum Message <> Reply to Message
"yp" <Yaswant.Pradhan@gmail.com> wrote in message
news:1173394662.260164.22540@s48g2000cws.googlegroups.com...
> Hi Bob.
> Sorry about the confusion. Here I go again...
> null=0.0D
int_LUT, 412.5, 1, 0.0D, 0.0D, 0.0D, 0.03D, result1
> int LUT, 412.5, 1, null, null, null, 0.03D, result2
>
> print,'RESULT 1: ', (result1[*])
> print, 'RESULT 2: ', (result2[*])
```

> RESULT 1: 0.090099994 > RESULT 2: 0.090071241 > I hope this is clearer now.

Thanks, that clears it up.
Can you post the routine int\_LUT, or post it on a webpage somewhere?

This is a facinating problem!

Cheers, bob

Subject: Re: Yet again, The Sky is Falling! Posted by yp on Fri, 09 Mar 2007 00:21:04 GMT

View Forum Message <> Reply to Message

```
On Mar 8, 11:04 pm, "R.G. Stockwell" <n...@email.please> wrote:
> "yp" <Yaswant.Prad...@gmail.com> wrote in message
  news:1173394662.260164.22540@s48g2000cws.googlegroups.com...
>
>
>
>
>
>> Hi Bob.
>> Sorry about the confusion. Here I go again...
>> ------
>> null=0.0D
>> int_LUT, 412.5, 1, 0.0D, 0.0D, 0.0D, 0.03D, result1
>> int_LUT, 412.5, 1, null, null, null, 0.03D, result2
>> print,'RESULT 1: ', (result1[*])
>> print,'RESULT 2: ', (result2[*])
>> ------
>
>> RESULT 1:
                  0.090099994
>> RESULT 2:
                  0.090071241
>> I hope this is clearer now.
> Thanks, that clears it up.
```

- > Can you post the routine int\_LUT, or post it on a webpage
- > somewhere?
- > This is a facinating problem!

>

- > Cheers.
- > bob- Hide quoted text -

>

> - Show quoted text -

Bob, David, Paul, Thank you all for sparing your time!

You can download the code from www.research.plymouth.ac.uk/casix/temp/test\_morel\_LUT.pro

You will also need the LUT (morel\_fq.dat) to run the code. www.research.plymouth.ac.uk/casix/temp/MOREL\_FQ.DAT

Best, yas

Subject: Re: Yet again, The Sky is Falling!
Posted by David Fanning on Fri, 09 Mar 2007 04:52:57 GMT
View Forum Message <> Reply to Message

#### yp writes:

>

- > You can download the code from www.research.plymouth.ac.uk/casix/temp/test\_morel\_LUT.pro
- > You will also need the LUT (morel\_fq.dat) to run the code.
- > www.research.plymouth.ac.uk/casix/temp/MOREL\_FQ.DAT

Well, I haven't tracked the actual culprit down yet, but the calculations in INT\_LUT are doing \*something\* to the variables that are being passed in by reference. If I pass everything in by value, then there is no difference:

```
IDL> int_LUT,412.5D, 1D, 0.0D, 0.0D, 0.0D, 0.03D, result3
IDL> int_LUT,412.5D+0, 1D+0, nul+0, nul+0, nul+0, 0.03D+0, result4
IDL> print,'RESULT 3: ', result3[*]
IDL> print,'RESULT 4: ', result4[*]
```

RESULT 4: 0.090100000 0.00000000 0.000000000

0.00000000 0.00000000 0.00000000 0.00000000

I suspect there is a calculation in there that sets one of the values very nearly to what you expect it to be, but not quite.

Still weird, but more believable. :-)

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: Yet again, The Sky is Falling!
Posted by R.Bauer on Fri, 09 Mar 2007 08:31:29 GMT

View Forum Message <> Reply to Message

```
yp wrote:
> On Mar 8, 11:04 pm, "R.G. Stockwell" <n...@email.please> wrote:
>> "yp" <Yaswant.Prad...@gmail.com> wrote in message
>>
>> news:1173394662.260164.22540@s48g2000cws.googlegroups.com...
>>
>>
>>
>>
>>
>>
>>> Hi Bob,
>>> Sorry about the confusion. Here I go again...
>>> ------
>>> null=0.0D
>>> int LUT, 412.5, 1, 0.0D, 0.0D, 0.0D, 0.03D, result1
>>> int_LUT, 412.5, 1, null, null, null, 0.03D, result2
>>> print,'RESULT 1: ', (result1[*])
>>> print,'RESULT 2: ', (result2[*])
>>> -----
>>
```

```
>>> RESULT 1:
                  0.090099994
>>> RESULT 2:
                   0.090071241
>>
>>> I hope this is clearer now.
>>
>> Thanks, that clears it up.
>> Can you post the routine int_LUT, or post it on a webpage
>> somewhere?
>> This is a facinating problem!
>>
>> Cheers,
>> bob- Hide quoted text -
>>
>> - Show quoted text -
>
> Bob, David, Paul,
  Thank you all for sparing your time!
> You can download the code from
www.research.plymouth.ac.uk/casix/temp/test_morel_LUT.pro
  You will also need the LUT (morel_fq.dat) to run the code.
  www.research.plymouth.ac.uk/casix/temp/MOREL_FQ.DAT
> Best.
> yas
Hi
I got an error using your example
IDL> int_LUT, 412.5, 1, 0.0D, 0.0D, 0.0D, 0.03D, result1
% Variable is undefined: BRDF.
% Execution halted at: INT_LUT
                                     102 test_morel_LUT.pro
                              102 test_morel_LUT.pro
%
               INT LUT
%
               $MAIN$
cheers
Reimar
Reimar Bauer
Institut fuer Stratosphaerische Chemie (ICG-1)
```

Forschungszentrum Juelich email: R.Bauer@fz-juelich.de

-----

a IDL library at ForschungsZentrum Juelich http://www.fz-juelich.de/icq/icq-i/idl icqlib/idl lib intro. html

\_\_\_\_\_\_

Subject: Re: Yet again, The Sky is Falling!
Posted by Paolo Grigis on Fri, 09 Mar 2007 09:25:41 GMT
View Forum Message <> Reply to Message

Here's a simple example of a routine that returns different values if it is called by value or by reference.

pro pg,a,b b=arg\_present(a) end

IDL> x=1 IDL> pg,x,b IDL> print,b 1

IDL> pg,x+0,b IDL> print,b 0

IDL> pg,x[0],b IDL> print,b 0

Ciao, Paolo

David Fanning wrote:

> yp writes:

>> You can download the code from www.research.plymouth.ac.uk/casix/temp/test\_morel\_LUT.pro >>

>> You will also need the LUT (morel\_fq.dat) to run the code.

>> www.research.plymouth.ac.uk/casix/temp/MOREL\_FQ.DAT

```
>
> Well, I haven't tracked the actual culprit down yet, but
> the calculations in INT_LUT are doing *something* to the
> variables that are being passed in by reference. If I
  pass everything in by value, then there is no difference:
>
>
> IDL>
          int_LUT,412.5D, 1D, 0.0D, 0.0D, 0.0D, 0.03D, result3
          int LUT,412.5D+0, 1D+0, nul+0, nul+0, nul+0, 0.03D+0, result4
> IDL>
> IDL>
          print, 'RESULT 3: ', result3[*]
> IDL>
          print, 'RESULT 4: ', result4[*]
> RESULT 3:
                 0.090100000
                                 0.00000000
                                                0.00000000
> 0.00000000
                 0.00000000
                                 0.00000000
                                                0.0000000
> RESULT 4:
                 0.090100000
                                 0.00000000
                                                0.00000000
> 0.00000000
                 0.00000000
                                 0.00000000
                                                0.0000000
> I suspect there is a calculation in there that sets one
> of the values very nearly to what you expect it to be,
 but not quite.
>
  Still weird, but more believable. :-)
>
> Cheers,
> David
```

Subject: Re: Yet again, The Sky is Falling!
Posted by Carsten Lechte on Fri, 09 Mar 2007 10:06:19 GMT
View Forum Message <> Reply to Message

Hi,

what happens if you do

int\_LUT, 412.5, 1, 0.0D, 0.0D, 0.0D, 0.03D, result1 int\_LUT, 412.5, 1, null, null, null, 0.03D, result2 int\_LUT, 412.5, 1, 0.0D, 0.0D, 0.0D, 0.03D, result3

is result1 eq result3?

Is there any code in int\_LUT that references files or COMMON blocks? Could that be called even though it is not supposed to be?

Subject: Re: Yet again, The Sky is Falling! Posted by yp on Fri, 09 Mar 2007 11:26:30 GMT

View Forum Message <> Reply to Message

Reimar,

Define the variable 'result1' before casting.

result1=[flt|dbl]arr(nwaves); in this case 1

It is not very convenient though I think I'll change the code to take care of this. But that's another subject. The main issue is similar to what Paolo Grigis has just mentioned.

```
> Hi
>
 I got an error using your example
>
IDL> int_LUT, 412.5, 1, 0.0D, 0.0D, 0.0D, 0.03D, result1
 % Variable is undefined: BRDF.
> % Execution halted at: INT_LUT
                                        102 test_morel_LUT.pro
> %
                 INT LUT
                                 102 test morel LUT.pro
> %
                 $MAIN$
>
> cheers
 Reimar
> Reimar Bauer
> Institut fuer Stratosphaerische Chemie (ICG-1)
> Forschungszentrum Juelich
> email: R.Ba...@fz-juelich.de
       a IDL library at ForschungsZentrum Juelich
   http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html
Hide quoted text -
> - Show quoted text -
```

Reimar, You have to define

# Subject: Re: Yet again, The Sky is Falling! Posted by R.Bauer on Fri, 09 Mar 2007 12:18:59 GMT

View Forum Message <> Reply to Message

Hi Paolo

that is correct. It shows perfectly how arg\_present works

cheers Reimar

```
Paolo Grigis wrote:
> Here's a simple example of a routine that returns different
> values if it is called by value or by reference.
>
> pro pg,a,b
>
> b=arg_present(a)
>
> end
>
> IDL> x=1
> IDL> pg,x,b
> IDL> print,b
       1
>
> IDL> pg,x+0,b
> IDL> print,b
      0
>
> IDL> pg,x[0],b
> IDL> print,b
      0
>
>
> Ciao,
> Paolo
>
> David Fanning wrote:
>> yp writes:
>>
>>> You can download the code from
>>> www.research.plymouth.ac.uk/casix/temp/test_morel_LUT.pro
>>>
```

```
>>> You will also need the LUT (morel_fq.dat) to run the code.
>>> www.research.plymouth.ac.uk/casix/temp/MOREL FQ.DAT
>>
>>
>> Well, I haven't tracked the actual culprit down yet, but
>> the calculations in INT_LUT are doing *something* to the
>> variables that are being passed in by reference. If I pass everything
>> in by value, then there is no difference:
>>
>>
>> IDL>
           int_LUT,412.5D, 1D, 0.0D, 0.0D, 0.0D, 0.03D, result3
           int LUT,412.5D+0, 1D+0, nul+0, nul+0, nul+0, 0.03D+0, result4
>> IDL>
           print,'RESULT 3: ', result3[*]
>> IDL>
>> IDL>
           print, 'RESULT 4: ', result4[*]
>>
                                                  0.00000000
>> RESULT 3:
                  0.090100000
                                   0.00000000
>> 0.00000000
                  0.00000000
                                  0.00000000
                                                 0.00000000
>>
>> RESULT 4:
                  0.090100000
                                  0.00000000
                                                  0.00000000
>> 0.00000000
                  0.00000000
                                  0.00000000
                                                 0.00000000
>>
>> I suspect there is a calculation in there that sets one
>> of the values very nearly to what you expect it to be,
>> but not quite.
>>
>> Still weird, but more believable. :-)
>>
>> Cheers,
>>
>> David
Reimar Bauer
Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
    a IDL library at ForschungsZentrum Juelich
 http://www.fz-juelich.de/icg/icg-i/idl icglib/idl lib intro. html
```

Subject: Re: Yet again, The Sky is Falling! Posted by yp on Fri, 09 Mar 2007 16:38:41 GMT

View Forum Message <> Reply to Message

## David,

- > Well, I haven't tracked the actual culprit down yet, but
- > the calculations in INT\_LUT are doing \*something\* to the
- > variables that are being passed in by reference. If I
- > pass everything in by value, then there is no difference:
- > I suspect there is a calculation in there that sets one
- > of the values very nearly to what you expect it to be,
- > but not quite.

Your guess is right. I caught the culprit - I had presumed the first value of thetaV to be zero (which is not) and when I pass the parameter by reference it picks the nearest value of the thetaV tab (1.078) and it does exactly what you suspected.

I did not realise that when I first wrote this last year... a classic example of reckless coding :( sorry to bother you all. But your suggestions were really helpful to figure this out :)

Thanks to all, vas

0

Subject: Re: Yet again, The Sky is Falling!
Posted by JD Smith on Tue, 13 Mar 2007 00:25:53 GMT
View Forum Message <> Reply to Message

On Fri, 09 Mar 2007 10:25:41 +0100, Paolo Grigis wrote:

```
> Here's a simple example of a routine that returns different values if it
> is called by value or by reference.
> pro pg,a,b
> b=arg_present(a)
> end
> IDL> x=1
> IDL> pg,x,b
> IDL> print,b
> IDL> print,b
> IDL> pg,x+0,b
> IDL> print,b
```

```
> IDL> pg,x[0],b > IDL> print,b > 0
```

That's a bit of a contrived example, given that the whole reason I originally lobbied for ARG\_PRESENT was so you could differentiate between by-value and by-reference arguments (which I observed IDL's own internal routines could do, so clearly it was functionality waiting to be exposed).

JD