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.] ;A (static)
> nwave = n_elements(wave) ;B (static)

```
>  sza = 45.0D                          ;C (static)
>  vza = 1.078D                          ;D (static)
>  dphi = 0.0D                          ;E (static)
>  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], 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
>             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
>  1.0450368       1.0480349       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
>
> 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
> 1.0447065 1.0477210 1.0543894 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