Subject: RE: Possible crippling bug in NR_(SIMP,ROMB)
Posted by mallozzi on Wed, 03 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

> I get erroneous results when using the NR_SIMP and NR_ROMB routines
> recursively that I belive traces to the routines themselves. Here is one
> of the simplest routines I could think of to illustrate the problem...

I too ran across this problem when I was using NR_QROMB. I talked with
IDL and they verified that there was a problem. Unfortunately I deleted
their response, but I remember that the explaination wasn't too clear;
perhaps the exact cause of the problem wan't immediately known. Anyway,
I got around it by calling the functions in the 'normal' manner
(i.e. not recursively). On initial coding, the recursive calling seemed
more natural to me...

mallozzi@ssl.msfc.nasa.gov

Subject: Re: Possible crippling bug in NR_(SIMP,ROMB)
Posted by rosentha on Thu, 04 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

L. Matthew Haffner (mhaffner@fosters.astro.wisc.edu) wrote:
: I get erroneous results when using the NR_SIMP and NR_ROMB routines
: recursively that I belive traces to the routines themselves. Here is one
: of the simplest routines I could think of to illustrate the problem (the
: idea for this recursive calling comes from NR, section 4.6):

```
: =========
: FUNCTION myfunc, x
:   return, 2.0*x
: END

: FUNCTION innerint, y
:   return, nr_qromb('myfunc', 0.0, 5.0)
: END

: PRO simpleint
:   print, nr_qromb('innerint', 0.0, 2.0)
: END
: =========
```

: Now, executing a statement like 'print, nr_qromb('myfunc',0.0,5.0)'
: correctly returns 25.000. However when adding the extra integral from 0-2
: over y as shown above in 'simpleint', IDL returns 37.500 when the correct
: answer should be 50. I have run this exact calculation through the C
: versions of the NR routines and they correctly report 50.000. I have

: checked this on SGI's, Alpha's, and DecStations with the same results.

: It seems the recursive calling is mangling the results. (Strangely, using
: the y range of 0-3 gives 50!) If I replace the return statment in
: 'innerint' with 'return, 25.0' I (of course) get 50. As listed above,
: 'innerint' should be returning 25.000 every call.

: Of course the integral above doesn't need to be calculated this way, but I
: have some that I need to do using the recursive trick. Has anyone
: encountered this before and found a workaround? Am I missing something
: obvious about recursive function calling in IDL?

From memory, these types of recursive calls also fail with
Fortran versions of the Numerical Recipes routines, because the
routines QSIMP etc. assume that internal variables of QTRAPZD are
unchanged between successive calls. The recommended solution is
to use separate copies of all the routines for each level of
integration! Of course this doesn't work with IDL since you
don't have access to the source code. Good luck.
--
--Colin Rosenthal | ``Don't smell the flowers -
--rosentha@obs.aau.dk | They're an evil drug -
--http://bigcat.obs.aau.dk/~rosentha | To make you lose your mind''-
--Aarhus University, Denmark | Ronnie James Dio, 1983 -