Subject: defining functions on-the-fly
Posted by mperrin+news on Tue, 22 May 2001 22:30:22 GMT
View Forum Message <> Reply to Message

I would like to be able to define some functions on-the-fly in IDL.
In other words, my program does some stuff, and computes that the desired
function is (some arbitrary function, not necessarily a polynomial).
I would then like to define the equivalent of
FUNCTION myfunction, x,y
  return, <my arbitrary expression in (x,y)>
END
and then be able to call myfunction(x,y).

Is there any easier way to do this than actually writing out a
myfunction.pro file to disk and compiling that? That way would certainly
work, but it strikes me as inelegant... At first I thought I could do this
with the EXECUTE statement, but you can't define functions using that.

 - Marshall

Subject: Re: defining functions on-the-fly
Posted by dirk on Wed, 23 May 2001 22:01:47 GMT
View Forum Message <> Reply to Message

In article <9eh94p$3ce$1@agate.berkeley.edu>,
Marshall Perrin <mperrin+news@arkham.berkeley.edu> wrote:
> Marc Schellens <m_schellens@hotmail.com> wrote:
>> try .comp:
>>
>> IDL> .comp [ENTER]
>> - function f
>> - return,42
>> - end
>> % Compiled module: F.
>> IDL> print,f()
>>   42
>> IDL>
>
> No, this doesn't work for what I have in mind - .comp is an executive command
> and so can only be used interactively. You can't use .comp in a procedure. I
> want *my software* to be able to define functions on the fly, not myself. So
> it looks like the best solution really is writing out a new .PRO file to the
> disk and compiling that.

I faced the same problem a while back, and this writing out solution was
unacceptable as well. The problem for me was that subsequent iterations
of the program would write a different equation in the .PRO file, but IDL

has already compiled a function by that name and doesn't look at or recompile the new .PRO file. Since .comp and .run are executive level, you can't force it to either.

This is what I did: (I had to interactively have N gaussians in a fitting program)

```
;prepare expressions for fitting and result plotting
plotresult = 'model=convol(exp((-1.)*(0.0'
FOR i=0, ncomp-1 DO $
  plotresult = plotresult + ' + Gauss1(v,result['+strtrim(string(3*i), 2)+ $
   ':'+strtrim(string(2+3*i), 2)+'])'
plotresult = plotresult+')), normspreadfunc, /center, /edge_truncate)'

; then
done = execute(plotresult)

; and you can do
oplot, v, model, color=200, thick=3
```

I wasn't clear on why execute wasn't working for you... Perhaps this helps.

Cheers, Dirk
