Subject: Re: how to use _EXTRA convention
Posted by Craig Markwardt on Fri, 09 Oct 1998 07:00:00 GMT
View Forum Message <> Reply to Message

nospam@ll.mit.edu (Joseph Scott Stuart) writes:
>
>
> I'm a little confused about the _EXTRA convention for passing
> parameters to a function works. I'm writing a routine to use with
> mpfit (thanks to Craig B. Markwardt), and I want to pass in some extra
> arguments via the functargs command. The mpfit procedure calls the
> function that I specify like this:
>
>   fp = call_function(fcn, xp, _EXTRA=fcnargs)
>
> So, if fcnargs = { X: Xvals, Y: Yvals, Err:Evals, D:dval, G:gval }
>
> then do I define my function like this:
>
> function myfunct, P, X=x, Y=y, Err=err, D=d, G=g
>
> or like this:
>
> function myfunct, P, _EXTRA=e
> x = e.X
> y = e.Y
> err = e.Err
> d = e.D
> g = e.G
>
>
> Or something else entirely?

Hi Stuart,

You can use either approach, but the first is probably the most
appropriate and easy to use.

HOWEVER, if your model is computed simply by one IDL function (it
often is), then I recommend using MPFITFUN, which is a driver function
for MPFIT. MPFITFUN does most of the work for you, and probably more
efficiently so.

Here's how you would do it:

function myfunct, x, p, d=d, g=g
  ymodel = { compute model here, using x, p, d and g }

```
  return, ymodel
end
```

p = mpfitfun('myfunct', x, y, err, p0, functargs={d:d, g:g})

MPFITFUN automatically takes care of translating FUNCTARGS and computes the residuals for you. The "x" values may have any dimensions and the "Y" values may have any dimensions (of course Y, YMODEL and ERR must be the same size). I highly recommend using MPFITFUN or MPFITEXPR instead of MPFIT unless your model function is hard to compute.

Good luck,
Craig

--
 -- -------------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@astrog.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 -- -------------