Subject: CONSTRAINED_MIN
Posted by rosentha on Tue, 08 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Has anyone had much joy using CONSTRAINED_MIN in idl 5.1?
First the obvious. The calling sequence in the documentation is

CONSTRAINED_MIN, Xbnd, Gbnd, Nobj, Gcomp, X, Inform

while that in the example code is

CONSTRAINED_MIN, x, xbnd, gbnd, nobj, gcomp, inform

The latter is correct.

Now the less obvious. On my first run it returns a value of the
keyword INFORM of 7, although it only documents values between -1
and 6 ! On the other hand, it apparently has found a better solution
than POWELL (my problem is currently unconstrained) in many fewer function
calls so I'm not ready to abandon ship just yet. Has anybody else
experienced anything similar?

--
Colin Rosenthal
High Altitude Observatory
Boulder, Colorado
rosentha@hao.ucar.edu

Subject: Re: CONSTRAINED_MIN
Posted by rosentha on Tue, 17 Nov 1998 08:00:00 GMT
View Forum Message <> Reply to Message

On Mon, 16 Nov 1998 09:29:21 GMT,
enric@indo.es <enric@indo.es> wrote:
> Hi!
>
> There's any more info on CONSTRAINED_MIN that the one on the help file?.
> The routine definition tells that the user can supply a function to
> compute the gradients...

I don't see that. Is it possible you read it wrong? In the help system
I'm using (v5.1.2) it says that derivatives are computed automatically
by finite differences. I agree that the documentation is poor. For example,
it states that INFORM=1 corresponds to fractional changes in the function
being smaller than EPSTOP for NSTOP consecutive iterations. However NSTOP is
nowhere defined.

--
Colin Rosenthal
High Altitude Observatory
Boulder, Colorado
rosentha@hao.ucar.edu

Subject: Re: constrained_min
Posted by Craig Markwardt on Wed, 28 Mar 2001 22:28:18 GMT
View Forum Message <> Reply to Message

Glenn Newnham <gnewnham@ses.curtin.edu.au> writes:

> I'm using constrained_min to find model parameter values which minimise
> the difference between modelled and measured data. The measured data is
> contained in various ascii files.
>
> At present my program opens the ascii file within the function to be
> minimised. this slows processing because the file is opened and closed
> every time constrained_min tries a new parameter set.
>
> Is there a way to read in these variables in the main program and make
> them global or to pass them to the function within the constrained_min
> command?

First suggestion: this is indeed a place for common blocks. It's ugly, but that's because constrained_min doesn't provide a way for you to get auxiliary data into your function (ie, by using _EXTRA, or a private parameter).

At command line:
```
common mydata, x, y, err
read_data, x, y, err
```

In function:
```
function doodad, p
  common mydata, x, y, err
  ;  .. compute function value ...
  return, f
end
```

However, I have a second suggestion: You sound like you may doing least squares curve fitting. This is really a quite specialized application which is accomplished by CURVEFIT (IDL library) and MPFIT (available from my web page). If you need to satisfy simple bounding constraints, then MPFIT is probably a good choice for you. Not only is it a quite robust algorithm, but it also allows you to set parameter constraints. It may be as simple as:

fit_params = mpfitfun('DOODAD', x, y, err, start_params)

and can be customized from there.

I've recently updated another program TNMIN, available from the same
web page. This function does more sophisticated function minimization
and also allows you to pass private data such as data values using the
FUNCTARGS keyword. It's overkill for curve fitting though.

Good luck,
Craig
Web page: http://cow.physics.wisc.edu/~craigm/idl/idl.html
(under Fitting and Tutorial)

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