Subject: choosing parameters for curvefit
Posted by nospam on Thu, 01 Oct 1998 07:00:00 GMT
View Forum Message <> Reply to Message

I'm using curvefit to fit a function with a bunch of parameters to a dataset. I'd like to be able to easily specify that some parameters should be held constant on a particular run, while others are fit. I want to do this to explore how well the curvefit is going and to explore the parameter space. The only way I've come up with so far to do this is using commons blocks as below. If you can think of another way to do this, let me know. I'd like to avoid common blocks on aesthetic grounds, but there doesn't seem to be any other way to pass extra information to the function that curvefit calls.

Here is how I have implemented it so far:

```
;; The function to fit, parameters in array A
pro func, X, A, Y, pder

 ; declare the common block
 common FITPARAMS, all_params, mask

 ; put the adjusted parameters from curvefit into the common
 ; block
 v = where(mask EQ 1)
 all_params[v] = A

 ; compute the function values and partial derivatives
 ; using the parameter values from all_params[]

 ....

end

;; in the main program:

 ; declare the common block
 common FITPARAMS, all_params, mask

 ; set up the parameters I want to fit this time around
 ; by putting a 1 in the corresponding place in the mask
 mask = [ 1, 0, 0, 1, 1, 0, 1]

 ; set up the initial values of the parameters
 all_params = [ ... ]

 ; read the independent data into X and the dependant
 ; data into Y
```

```
...
yfit = curvefit( X, Y, 1.0/Y, sigmas)

; analyze the results
...
```

--
Scott Stuart
stuart at ll mit edu
