Subject: Re: Challenging question - array curve fitting
Posted by Craig Markwardt on Mon, 23 Apr 2007 03:50:39 GMT
View Forum Message <> Reply to Message

Here's a late follow-up to your questions.

JD Smith <jdsmith@as.arizona.edu> writes:
> Just an unrelated followup question for Craig... MPFIT is (I hate to
> say it) quite slow, in particular for problems like this, where you
> want to simultaneously fit hundreds or thousands of things to complex
> models. I know it wasn't designed for speed, and the things it does
> are amazing (I particularly like arbitrary constraint expressions),
> but I wonder, what do you think the chances of improving the speed
> are? Would it require a fundamental rewrite, or is it more sensible
> to fall back on a C or FORTRAN version of the library (I actually
> don't know how the speed compares)? Should we lobby ITTVIS to throw
> out their non-linear fitters, and incorporate MPFIT into compiled
> code? Just want to get your perspectives. Thanks again for such a
> wonderfully useful tool.

JD, at some point early on, I tried to squeeze as much performance out of MPFIT as I could. I imagine that today's processors are not the same as then, but the difference is not likely to be huge.

I did a simple test of fitting a random vector with a polynomial,

```
yy = randomn(seed,1024L*1024L)
profiler
p = mpfitfun('quad', dindgen(1024L*1024L), yy, 1d, [1d,1d,1d], perror=dp)
profiler, /report
```

and the results were 30% of the time was spent evaluating the polynomial function, 38% spent factoring the jacobian, 15% calculating vector norms. Even if we could make MPFIT run infinitely fast, it could only ever be three times faster. The factorization is the biggest proportion of time spent, but then it needs to be.

I think there are several factors to consider.

* If your function is very complicated and time consuming, then it won't matter how fast MPFIT is. My example shows that even a very simple quadratic model function with few parameters, already takes 30% of the total execution time to evaluate.

* The function evaluation speed can be significantly improved if you can compute analytical derivatives.

* If your fit takes too many iterations to converge, then you can relax the convergence criteria (ftol, xtol, gtol).

* If you're someone like the original poster that has many fits to perform, with a small number of points per fit, then it might indeed pay off to group multiple fits together, so you don't incur the setup and teardown costs of each MPFIT call.

In the meantime I have developed a C library version of the fitter. I have never tested whether it was faster, since that wasn't the point (C was a delivery requirement to a larger project).

Happy fitting!
Craig

P.S.

```
function quad, x, p
  return, p[0] + p[1]*x + p[2]*x*x
end
```

--

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