

King, Stephen (STFC,RAL,ISIS)

From: Paul Kienzle <paul.kienzle@nist.gov>
Sent: 10 July 2014 18:16
To: King, Stephen (STFC,RAL,ISIS)
Cc: sasview-developers@lists.sourceforge.net
Subject: Re: [Sasview-developers] FW: Error estimates from fits
Attachments: sim.py

On Jul 10, 2014, at 10:34 AM, <stephen.king@stfc.ac.uk> <stephen.king@stfc.ac.uk> wrote:

> One of my users has been fitting some polymer scattering using the PolyGaussCoil model.
>
> For some reason they decided to compare the simple and complex fitting engines. The good news is they got the same fit parameters! But the parameter uncertainties are significantly different for some parameters.
>
> SIMPLE
> Background 0.046288 +/- 0.0011378
> Scale 81.882 +/- 0.22307
> Poly_m 1.1
> Rg 89.727 +/- 0.15731
>
> ChiSq/Npts 21.939
>
> COMPLEX
> Background 0.046288 +/- 0.00097874
> Scale 81.882 +/- 1.9189
> Poly_m 1.1
> Rg 89.727 +/- 1.3533
>
> ChiSq/Npts 21.939
>
> Their question is three-fold:
> - Is this to be expected?
> - Why the difference in uncertainties?
> - Which uncertainties should they take as 'real'?

The uncertainties should be similar.

Park computes the Jacobian matrix using the center point formula:

$$J[i,j] = (f(p[i]+h) - f(p[i]-h)) / 2h \quad \text{for parameter } p[i], \text{ data point } lq[j]$$

where

$$h = 1e-8 * (\text{max} - \text{min}) \text{ for bounds constrained parameters}$$

$$h = p[i]*1e-8 \text{ for unconstrained parameters}$$

The covariance matrix C is $\text{inv}(J'J)$, computed using **singular value decomposition** as:

$$U V S = J$$

$$C = \text{inv}(J'J) = V \text{ inv}(S S) V'$$

Similarly, scipy computes the jacobian using the forward difference formula:

$$J[i,j] = (f(p[i]+h) - f(p)) / h \text{ for parameter } p[i], \text{ data point } lq[j]$$

where

$$h = \text{abs}(p[i]) * 1e-8$$

The covariance matrix C is computed using QR decomposition as:

$$Q R = J$$

$$C = \text{inv}(J'J) = \text{inv}(R'R)$$

The center point formula should be more accurate, though twice as expensive to calculate. SVD and QR decomposition are equally stable but R'R is ill-conditioned, so I trust the SVD calculation more. Even so, I don't expect a factor of 10 difference between the two.

To determine which uncertainties are more real, you could run a forward simulation on the model with parameters pulled at random from the two uncertainties and see which is a better match to the uncertainty of the data. Drop the attached sim.py into your data directory and change to the correct file name.

Using the > Tool > Python Shell menu you can start a python shell in which you can type:

```
pwd # show what directory you are in
cd data # change to your data directory
execfile('sim.py')
```

This will show the 1-sigma uncertainty in the theory function values, which should be on the same order as the 1-sigma uncertainty on the data values. I haven't tested the script on windows, but it worked on my linux box.

- Paul