Logic of the Code

Aryana Haghjoo

March 14, 2023

1 Derivation of the Covariance Matrix

To find the analytic form of the covariance matrix, we need to find the second derivative of the chi-square with respect to the parameters of the model. Thus, we start with the matrix definition of the chi-square:

$$\chi^2 = (d - A(m))^T N^{-1}(d - A(m)) \tag{1}$$

Where d is an array containing the data, A is the theoretical model, which is dependent on the values of parameters (the dependency can be non-linear), and N is the noise matrix. If the noise is uncorrelated, the diagonal values of the noise matrix are just the square of the error bar on each data point.

$$\frac{d\chi^2}{dm} = -(\frac{dA(m)}{dm})^T N^{-1}(d - A(m)) - (d - A(m))^T N^{-1}\frac{dA(m)}{dm} \tag{2}$$

Since we know that:

$$(N^{-1})^T = N^{-1} \tag{3}$$

$$[\frac{dA(m)}{dm}N^{-1}(d - A(m))]^T = (d - A(m))^T N^{-1}\frac{dA(m)}{dm} \tag{4}$$

Substituting in 2, we get:

$$\frac{d\chi^2}{dm} = -2(\frac{dA(m)}{dm})^T N^{-1}(d - A(m)) \tag{5}$$

$$\tag{6}$$

Thus, we can calculate the second derivative:

$$\frac{d^2\chi^2}{dm^2} = -2(\frac{d^2A(m)}{dm^2})^T N^{-1}(d - A(m)) - 2(\frac{dA(m)}{dm})^T N^{-1}(-\frac{d\chi^2}{dm}) \tag{7}$$

$$\tag{8}$$

We can neglect the first term because of following two reasons:

1. First: The $(d - A(m))$ component, which is the residual, can take both negative and positive values; Thus on average, it will be close to zero.

2. Second: The fitted model should be very close to the data. Therefore, we expect that $(d - A(m)) \approx 0$

Finally, we are left with:

$$\frac{d^2\chi^2}{dm^2} = 2(\frac{dA(m)}{dm})^T N^{-1}(\frac{dA(m)}{dm}) \tag{9}$$

Which is the definition of the **Covariance Matrix**

2 The Levenberg-Marquardt Algorithm

The Levenberg-Marquardt (LM), also known as the damped least-squares (DLS) method, is a fitting algorithm used for non-linear least-squares problems. This algorithm is iterative, and on each step, the parameters m will be replaced by $m + \delta m$. We aim to find δm by minimizing the chi-square calculated based on the perturbed parameters.

We knew that:

$$\chi^2(m) = (d - A(m))^T N^{-1}(d - A(m)) \tag{10}$$

$$\chi^2(m + \delta m) = \chi^2(m) + \frac{d\chi^2}{dm}\delta m \tag{11}$$

$$\frac{d\chi^2(m + \delta m)}{dm} = \frac{d}{dm}(\chi^2) + \frac{d}{dm}(\frac{d\chi^2}{dm}\delta m) \tag{12}$$

Since we calculated the first order derivative of chi-square in 2, we have:

$$\frac{d\chi^2(m + \delta m)}{dm} = -2(\frac{dA(m)}{dm})^T N^{-1}(d - A(m)) + \frac{d^2\chi^2}{dm^2}\delta m + \frac{d\chi^2}{dm}\frac{d}{dm}(\delta m) \tag{13}$$

Where the last term is zero since δm does not have any dependencies on m, and we have already calculated the seond derivative of chi-square in 9. Thus, we are left with:

$$\frac{d\chi^2(m + \delta m)}{dm} = -2(\frac{dA(m)}{dm})^T N^{-1}(d - A(m)) + 2(\frac{dA(m)}{dm})^T N^{-1}(\frac{dA(m)}{dm}) \tag{14}$$

We define $d - A(m) \equiv r$, and $\frac{dA(m)}{dm} \equiv A'$, we get:

$$A'^T N^{-1} A' \delta m = A'^T N^{-1} r \tag{15}$$

$$\delta m = (A'^T N^{-1} A')^{-1} A'^T N^{-1} r \tag{16}$$

Which is the basis for **Newton's method**. However, this method does not always converge, especially on complicated likelihood surfaces. To solve this issue, we add a new term to the left-hand side of the above equation, which involves a control parameter named Λ that is updated on each iteration.

$$(A'^T N^{-1} A' + \Lambda I)\delta m = A'^T N^{-1} r \tag{17}$$

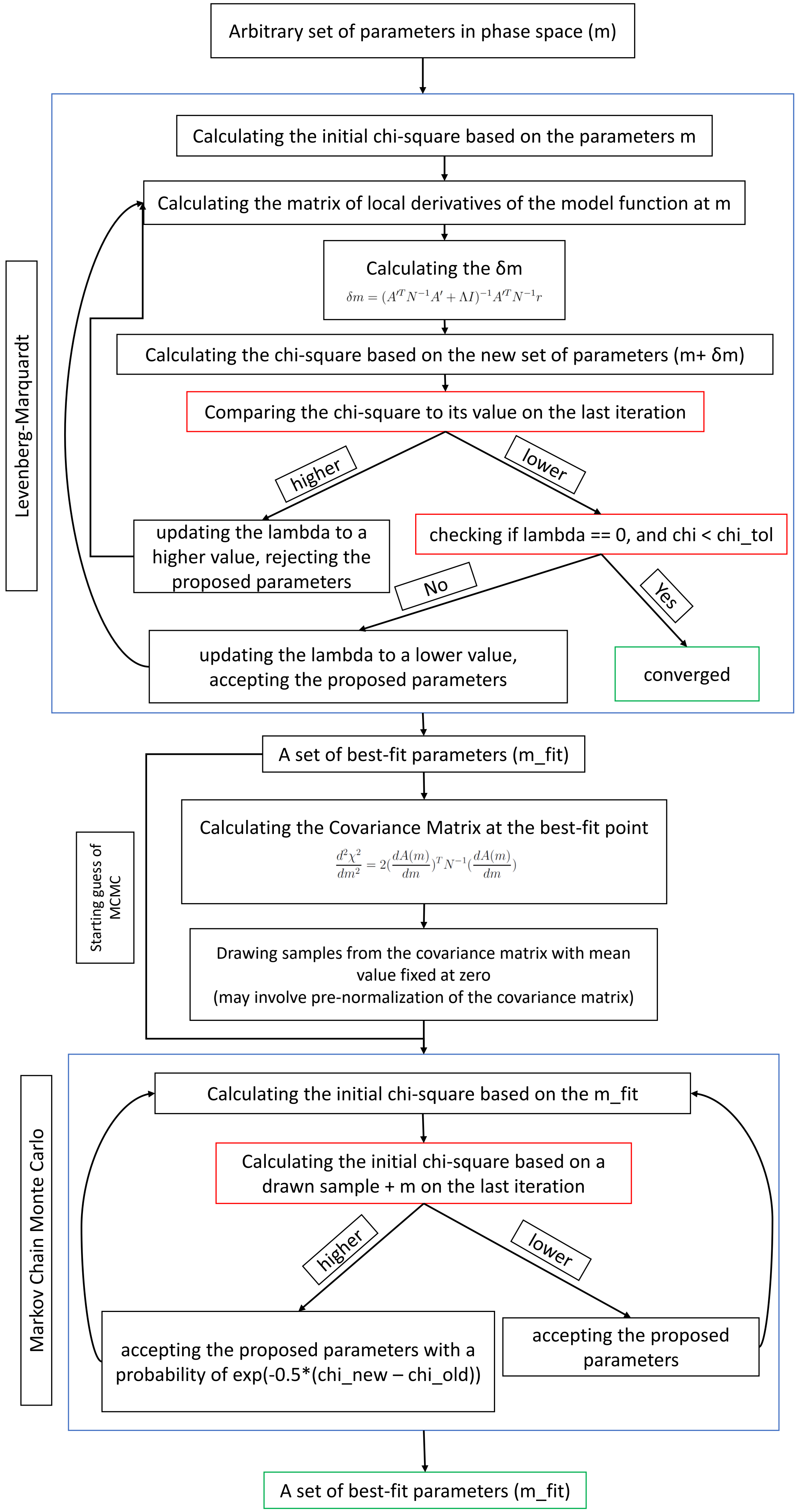
$$\delta m = (A'^T N^{-1} A' + \Lambda I)^{-1} A'^T N^{-1} r \tag{18}$$

On each iteration, we compare the chi-square to its value in the last step. If we encounter a higher value, the Λ will be multiplied to a constant arbitrary number (> 1). Otherwise, it will be divided with another constant value (> 1). For practical purposes, if Λ takes a value lower than a constant small arbitrary number, we set it equal to zero. If the Λ is zero, and the chi-square is less than an arbitrary threshold value, we declare the convergence of the algorithm.

3 Combining the Levenberg-Marquardt and Markov Chain Monte Carlo

There are various ways to force a MCMC to converge faster. One of which, involves using a posterior distribution that accounts for correlations between different parameters. To generate this posterior distribution, we first use a Levenberg-Marquardt fitter to obtain a set of best-fit parameters and the covariance matrix of parameters based on the local derivatives of the likelihood at the best-fit point. We then draw samples from this covariance matrix to generate the posterior distribution.

The logic of the above-mentioned algorithm can be summarized in the following chart.



Arbitrary set of parameters in phase space (m)
Levenberg-Marquardt
Calculating the initial chi-square based on the parameters m
Calculating the matrix of local derivatives of the model function at m
Calculating the δm
$\delta m = (A'^T N^{-1} A' + \Lambda I)^{-1} A'^T N^{-1} r$
Calculating the chi-square based on the new set of parameters (m+ δm)
Comparing the chi-square to its value on the last iteration
higher
lower
updating the lambda to a higher value, rejecting the proposed parameters
checking if lambda == 0, and chi < chi_tol
No
Yes
updating the lambda to a lower value, accepting the proposed parameters
converged
A set of best-fit parameters (m_fit)
Starting guess of MCMC
Calculating the Covariance Matrix at the best-fit point
$\frac{d^2\chi^2}{dm^2} = 2(\frac{dA(m)}{dm})^T N^{-1} (\frac{dA(m)}{dm})$
Drawing samples from the covariance matrix with mean value fixed at zero
(may involve pre-normalization of the covariance matrix)
Markov Chain Monte Carlo
Calculating the initial chi-square based on the m_fit
Calculating the initial chi-square based on a drawn sample + m on the last iteration
higher
lower
accepting the proposed parameters with a probability of exp(-0.5*(chi_new – chi_old))
accepting the proposed parameters
A set of best-fit parameters (m_fit)