

Author: Hoi Yeung Wong
(Ocean Wong)
Director of Studies: Robin Smith
Primary Supervisor: Chantal Nobs
Organization: Culham Centre for Fusion Energy
Date: December 2019

Abstract

In this document, the method of particle spectrum unfolding using Tikhonov Regularization is introduced. Relative-entropy was used as the regularizing function.

This is the same unfolding method as the one underlying the popular unfolding code MAXED, but with several improvements, including improved accuracy of neutron yield estimation, and increased flexibility for the user to choose the Tikhonov regularization parameter to avoid overfitting.

A general neutron spectrum unfolding algorithm by regularization using relative-entropy

1 Background

Regularization algorithms aim to minimize a metric, which is a sum of two quantities: $(\chi^2)_{\text{reaction rates}}$, and another quantity measuring the deviation of the solution spectrum from the *a priori* spectrum. The mixing ratio between these two quantities is controlled by the Tikhonov parameter τ , which puts more or less weight on the relative-entropy as its value increases or decreases respectively.

The algorithm described in this paper was created for the purpose of fusion neutron spectra unfolding, as after a comprehensive review of literature, it was decided that the existing neutron spectrum unfolding methods cannot be generalized to the problem of fusion neutron measurement, which has a higher degree of under-determination, and where the possible neutron spectra differs a lot more from each other (such that the *a priori* may be very far from the true spectrum).

This is particularly useful for fusion neutron spectrum unfolding, where there are not many experimental data available, thus no reasonable parametrisation or other sophisticated method of simplifying the neutron spectrum can be used.

Therefore the mathematical derivation in creating the regularization algorithm in this paper was done with as little loss of generality as possible, such that it can be applied onto any particle spectrum unfolding problem. As long as the reaction rates, response matrix, and *a priori* is known, this algorithm is applicable.

$$\text{loss}(\phi_{\text{sol}}) = (\chi^2)_{\text{reaction rates}}(\phi_{\text{sol}}) + \tau \cdot \text{RelativeEntropy}_{\phi_0}(\phi_{\text{sol}}) \quad (1)$$

where

ϕ_0 is the *a priori* spectrum,

ϕ_{sol} is the algorithm's output spectrum,

$\text{Loss}(\phi_{\text{sol}})$ is the loss value, i.e. the value outputted by the loss function.

Due to the underdetermination condition of the unfolding problem, there exist multiple (i.e. infinitely many) solutions which will minimize χ^2 (potentially down to zero). If one wishes to find a unique solution among these multiple solutions, we can apply a condition requiring it to deviate minimally from the *a priori* spectrum.

1.1 Motivation

While some programs of particle spectra unfolding via regularization exist, not all are rigorously derived and none allows the user to easily select a regularization constant. Take MAXED for example, which is the most widely used regularization unfolding method[10]. It chooses τ by forcing $(\chi^2) = \text{number of degree of freedoms}$ [8]. This form of “regularization” leads to no insight about how far off is the *a priori* spectra, and therefore the user won’t be able to know if the spectrum is accurate or not.

Additionally, the mathematical basis is flawed: In MAXED’s derivation, it does not respect its own assumption that the solution distributions is a probability distribution by allowing it to be not normalized, i.e. $\sum_i \phi_i \neq 1$; and it also applied the relative-entropy definition backwards: the apriori distribution ϕ_0 should become the first term in the bracket of relative entropy, i.e. $D_{KL}(\phi_0||\phi_{sol})$, but instead the relative-entropy was erroneously defined as $D_{KL}(\phi_{sol}||\phi_0)$. Since the two terms in the brackets are not commutable, applying the second definition will give an incorrect result.

These minor errors are perhaps tolerable in the era of fission neutron spectrum unfolding, where the solution space is small; but as we apply the technique of neutron spectrum unfolding on fusion neutron spectra, which has a much wider range of possible solutions, these minor errors in the algorithms used may unknowingly lead to erroneous solutions.

To correct for this, the following algorithm was derived in a more rigorous manner, and has the flexibility to allow users to choose the appropriate regularization constant τ . It also does not rely on any pre-existing numerical solvers to obtain a solution (i.e. no bin-hopping or simulated annealing). A single minimum loss value in equation 1 is guaranteed, without the chance of being trapped at sub-optimal stationary points.

1.2 Loss function

The loss function chosen for this algorithm is stated in equation 1. To measure the deviation of the solution’s would-have-been reaction rate (\mathbf{Z}') from the measured reaction rate(\mathbf{Z}_0), $\chi^2 = \sum_k (\frac{Z'_k - Z_{0k}}{\sigma_k})^2$ was used. This is a widely used metric for measuring the deviation between pairs of numbers, and is generally accepted as the standard metric used to measure the deviation between reaction rates in neutron spectra unfolding [5][9].

Perhaps a more interesting question is why should we use relative-entropy as the regularizing function in the loss function.

The regularizing function, in the context of neutron spectrum unfolding, refers to the function used to compute the “deviation” of one spectrum from another (from the solution spectrum to the *a priori* spectrum).

When choosing a regularizing function, it is desirable to have the following features:

1. Can naturally obey the physical requirement of no-negative-flux without applying additional external constraint onto the set of equations;
2. For the same amount of absolute deviation $\Delta\phi$, harsher penalty should be given to bin with lower *a priori* flux.

Penalty contributed by the i^{th} bin’s deviation $>$ penalty contributed by the j^{th} bin’s deviation if the solution deviates by the same amount, i.e. $|\phi_{sol,i} - \phi_{0,i}| =$

$\Delta\phi = |\phi_{sol,j} - \phi_{0,j}|$, but the $\phi_{0,i} < \phi_{0,j}$

To satisfy both of these requirements,

1. The gradient of the loss function must approach $-\infty$ as the flux approaches 0^+ from the positive side,
2. The magnitude of the 2^{nd} order derivative of the loss function $\frac{\partial^2(\text{loss})}{\partial\phi_{sol,i}^2}$ must increase with decreasing *a priori* value $\phi_{0,i}$

Other regularizing functions considered before settling on the decision of using relative-entropy includes:

- Fractional deviation (obeys requirement 2 but not 1)
- Mean squared deviation in log-space
- Absolute deviation in log-space
- Determinant of the Fisher Information Matrix ¹

In the end, relative-entropy was chosen as it satisfies both of the requirements above, with the additional benefit of being the only unique and non-self-contradictory metric that can be used to measure deviation between probability distributions (See Appendix A in [7]). Therefore it is chosen as the metric to be used in the regularizing function in this paper.

Serendipitously, the loss function, when composed in this manner, has no local trapping suboptimal points (See appendix A), allowing to use of Lagrangian multipliers to find the point of minimum, which is detailed in section 3.

There are other implementations of the regularization method in ways that does not require an *a priori*. Take minimum (self-)entropy for example, which is the algorithm used by [2] and [4]. This is only a special implementation of the principle of MaxEnt (Maximum (relative-)Entropy), where a naïve *a priori* is assumed, i.e. it is equivalent to using a flat *a priori* in MAXED.

1.3 Definition of relative-entropy

Before proceeding any further, it is useful to know what relative-entropy is. It is also known as Kullback-Leibler divergence (D_{KL}); and has been misnamed as cross-entropy(H) in some papers[7] (See equation 5 for details). Both measures the deviation of one normalized probability distribution from another. These probability distribution can be continuous or discrete.

For example, let there be two discrete probability distributions Q and P , over x:

$$H(P, Q) = - \sum_x P(x) \log(Q(x)) \quad (2)$$

$$S(P) = - \sum_x P(x) \log(P(x)) \quad (3)$$

$$D_{KL}(P||Q) = \sum_x P(x) \log\left(\frac{P(x)}{Q(x)}\right) \quad (4)$$

$$D_{KL}(P||Q) = H(P, Q) - S(P) \quad (5)$$

¹Fisher information of a variable vector (which the neutron spectrum is) should form a MATRIX, not a scalar, unlike the papers [3][6] mentioned.

where $S(P)$ is the (self-)entropy of the distribution. (Intuitively, when analogized to drawing balls from a bag, $S(P)$ is how “surprising” the resulting ball’s colour is on average.) When P is fixed (as is the *a priori*), then $S(P)$ is a constant, so $D_{KL}(P||Q)$ and $H(P, Q)$ only differ by a fixed, constant value for any P .

Note the non-commutativity of $D_{KL}(P||Q)$: if the role of P and Q in equation 4 were reversed, a different answer will be obtained; but it will still give a loss function landscape that only has one global minimum.

The choice of the base of the \log in the equations above is arbitrary. (In the ball-drawing analogy this number is equal to the number of colours available.) In the context of information theory and computer science this is usually chosen as 2, and given the unit “shannons”; but for physics application, and for the rest of this text, it will be chosen as base e , and given the unit “nats”, where 1 shannon = 0.693 nats.

In a typical model selection problem, one would use the data/ground truth as distribution P , and postulate several different *a priori* models (AKA hypothetical models) as Q , and choose one of the *a priori* Q which has the smallest $D_{KL}(P||Q)$ as the best fit model, as that Q is considered to be the model that loses the least amount of information when it is used to approximate the ground truth P . [1].

In an underdetermined unfolding problem, the ground truth is never known. Instead, in the regularizing part of the loss-function, an *a priori* ϕ_0 will be provided by the user of the unfolding program; but the solutions. Therefore we should replace the “ground truth” with the user-provided *a priori*, and use the program’s potential output solution ϕ_{sol} in the place of the hypothetical models. This should, in theory, lead to the expression $D_{KL}(\phi_0||\phi_{sol})$.

The act of minimizing D_{KL} in the expression above is equivalent to minimizing the information lost when replacing the user-defined *a priori* with the solution distribution outputted by the program.

Another advantage of using the user-defined *a priori* as the “ground truth” is that the algorithm will still function properly even if the user-defined *a priori* contain one or more bins where the flux $\rightarrow 0^+$. This is because the $D_{KL}(\phi_0||\phi_{sol})$ function (and its gradient) remains defined even if the user-defined *a priori* has one or more bins $\rightarrow 0^+$; but the same cannot be said for $D_{KL}(\phi_{sol}||\phi_0)$

2 Assumption and notations

To keep the derivation simple, we assume that all count rates are high enough that their Poisson distributions can be approximated using Gaussian distributions. We will also ignore the systematic error, i.e. assume that the error on the cross-sections of various nuclei is zero.

The propagation of error due to systematic error (i.e. error associated with the **R** matrix) can be accounted for at a later stage by the user of Monte Carlo wrapper function around the algorithm; or can be extended using the result of 3.2 and 3.3

2.1 Notations

In previous texts, the neutron flux is usually represented with ϕ , and the reaction rates with Z . In this text, however, the neutron flux needs to be normalized to obtain a probability distribution instead. The probability distribution of a neutron being in the i^{th} bin will be indicated with f ,

$$\sum_i f_i = 1 \tag{6}$$

in line with the notations used in [7] derivation.

Note that these definitions are made with the context of activation foil neutron spectrum unfolding in mind, where some neutrons (ϕ) are incident on the foils, producing some number of nuclides (N) according to a predetermined response matrix $\underline{\underline{R}}\phi$ as a result. But it is possible to generalize it to other applications, i.e. it is equally valid to replace “Total # of nuclide k created” with “total number of pulse with height=k recorded” when using recoil/scintillation detectors; and replace ”neutron spectrum” with “neutron and γ spectrum” for detector sensitive to both, operating in a mixed field environment.

Below are some of the notations which will be used:

number of reactions = m
 so that daughter nuclide k has a range $1 \leq k \leq m$
 number of energy groups = i
 so that the i^{th} or j^{th} energy group $1 \leq i, j \leq n$
A priori probability distribution
 of neutron spectrum = \mathbf{f}^{DEF}

 probability distribution
 of neutron spectrum = \mathbf{f} (vectors are written in bold italics)

 Total # of nuclides created
 according to experimental data = \mathbf{N}

 Total # of nuclides created as
 predicted by the solution = \mathbf{N}'

 Neutron yield = Y

 Response matrix = $\underline{\underline{\mathbf{R}}}$
 And the k -th row of the response
 matrix will be written as
 a vector = \mathbf{R}_k

 element of a matrix is indicated with
 a pair of subscript indices, e.g. = R_{ki}

 element of a vector is indicated with
 a subscript index, e.g. = f_i

 Iteration of a variable is
 indicated with superscript, e.g. = \mathbf{v}^g

 Non-linear operation on a vector
 is equivalent to element-wise
 operation on the vector, e.g. : $\left(\frac{1}{\mathbf{f}}\right)_i = \frac{1}{f_i}$

 The element-wise multiplication
 is denoted by the hollow dot: \circ

R_{ki} denotes how much will the i^{th} bin's flux contribute to the k^{th} reaction rate.
 Therefore the dot product between \mathbf{R}_k and \mathbf{f} gives the reaction rate per incident neutron,

$$\mathbf{R}_k \cdot \mathbf{f} = \frac{N'_k}{Y} \quad (7)$$

Or, equivalently,

$$\mathbf{N} = Y \underline{\underline{\mathbf{R}}} \mathbf{f} \quad (8)$$

3 Algorithm

Using the constraints

$$N'_k = Y \sum_i R_{ki} f_i$$

$$\sum_i f_i \equiv \|\mathbf{f}\|_1 = 1$$

We aim to minimize the following loss value:

$$\text{loss} = \chi^2(\mathbf{f}) + \text{reg}(\mathbf{f}) + \frac{\gamma}{2}(\underline{\mathbf{R}} - \underline{\mathbf{R}}_0) \cdot \underline{\mathbf{S}}_{\mathbf{R}}^{-1}(\underline{\mathbf{R}} - \underline{\mathbf{R}}_0)$$

$$\chi^2(\mathbf{f}) = \frac{1}{2}(\mathbf{N}' - \mathbf{N}) \cdot \underline{\mathbf{S}}_{\mathbf{N}}^{-1}(\mathbf{N}' - \mathbf{N})$$

$$\text{reg}(\mathbf{f}) = \tau \mathbf{f}^{DEF} \cdot (\ln \mathbf{f}^{DEF} - \ln \mathbf{f})$$

$\underline{\mathbf{S}}_{\mathbf{N}}^{-1}$ denotes the inverse of the covariance matrix for the measured reaction rates \mathbf{N} ; if the covariance matrix is purely diagonal, i.e. no correlation between different measurements, then $\underline{\mathbf{S}}_{\mathbf{N}}^{-1}$ becomes the familiar form $S_{ii}^{-1} = \frac{1}{\sigma_i^2}$. The term with γ accounts for any possible deviation in the response matrix, where gamma controls the importance of regularizing the deviation of $\underline{\mathbf{R}}$ from $\underline{\mathbf{R}}_0$. For section 3.1, we will fix $\underline{\mathbf{R}}$ at $\underline{\mathbf{R}}_0$, and $\gamma = 0$.

Using the method of Lagrangian multipliers to minimize this loss value under the stated constraints leads to the Lagrangian expression

$$\mathcal{L}(\mathbf{f}, \mu) = \frac{1}{2}(\mathbf{Y}\underline{\mathbf{R}}\mathbf{f} - \mathbf{N}) \cdot \underline{\mathbf{S}}_{\mathbf{N}}^{-1}(\mathbf{Y}\underline{\mathbf{R}}\mathbf{f} - \mathbf{N}) - \tau \mathbf{f}^{DEF} \cdot \ln \mathbf{f} + \mu(\|\mathbf{f}\|_1 - 1)$$

At the g^{th} iteration, the value of best fit neutron yield Y that minimizes the loss value is straightforwardly found as

$$Y^g = \frac{\mathbf{N} \cdot \underline{\mathbf{S}}_{\mathbf{N}}^{-1}(\underline{\mathbf{R}}\mathbf{f}^g)}{(\underline{\mathbf{R}}\mathbf{f}^g) \cdot \underline{\mathbf{S}}_{\mathbf{N}}^{-1}(\underline{\mathbf{R}}\mathbf{f}^g)} \quad (9)$$

without any circular dependence, assuming \mathbf{f}^g is known.

However, the same cannot be said for \mathbf{f} :

$$\begin{cases} Y(\underline{\mathbf{R}}\mathbf{f}) \cdot \underline{\mathbf{S}}_{\mathbf{N}}^{-1}Y\underline{\mathbf{R}} - \mathbf{N}\underline{\mathbf{S}}_{\mathbf{N}}^{-1}Y\underline{\mathbf{R}} - \tau \mathbf{f}^{DEF} \circ \frac{1}{\mathbf{f}} = -\mu & (10) \\ \|\mathbf{f}\|_1 = 1 & (11) \end{cases}$$

Where line 10 consists of n equations, one for each component of \mathbf{f} .

To get the value for μ , one can multiply f_i onto the $1 < i^{th} < n$ equation, then sum up all n equations. Apply the condition 11 on the RHS and $\|\mathbf{f}^{DEF}\|_1 = 1$ on the LHS gives

$$\mu^g = \tau - (Y^g \underline{\mathbf{R}}\mathbf{f}^g - \mathbf{N}) \cdot \underline{\mathbf{S}}_{\mathbf{N}}^{-1}(Y^g \underline{\mathbf{R}}\mathbf{f}^g) \quad (12)$$

Applying equation 12 on 10 gives statement resembling the following

$$\nabla \chi^2(\mathbf{f}) + \nabla \text{reg}(\mathbf{f}) = \mathbf{0}$$

where $\text{reg}(\mathbf{f})$ stands for the regularization function = RelativeEntropy(\mathbf{f})

At the g^{th} iteration, the equation above may not hold true, i.e.

$$\boldsymbol{\epsilon}^g = \nabla \chi^2(\mathbf{f}^g) + \nabla \text{reg}(\mathbf{f}^g)$$

Where $\boldsymbol{\epsilon}^g$ is a vector represents “residual steepness”.

3.1 Methods to iteratively approach solution \mathbf{f}

At this stage, apart from heedlessly taking steps downhill towards the point of lowest loss value, there are two ways to find the solution:

3.1.1 Finding the optimal point using the linear approximation of the loss function gradient in the vicinity of the solution

To ensure convergence in the fewest iteration, we can adjust \mathbf{f}^g to a new position \mathbf{f}^{g+1} such that its new gradient is expected to cancel out ϵ^g .

$$\nabla\chi^2(\mathbf{f}^{g+1}) - \nabla\chi^2(\mathbf{f}^g) + \nabla\text{reg}(\mathbf{f}^{g+1}) - \nabla\text{reg}(\mathbf{f}^g) + \epsilon^g = \mathbf{0}$$

To make the equation solvable, simplification must be made. In the following equation we have done so by Taylor expanding the gradient of the regularization function to its 1st order.

Using $\mathbf{1}$ to represent a vectors of all '1's, and using $\Lambda(\mathbf{f})$ to represent an $n \times n$ diagonal matrix whose main diagonal is populated with vector \mathbf{f} , and $(\frac{\mathbf{f}^{DEF}}{\mathbf{f}^g})_i = \frac{f_i^{DEF}}{f_i^g}$, we have

$$\left[Y^g \underline{\mathbf{R}}^T \underline{\mathbf{S}}_{\mathbf{N}}^{-1} \underline{\mathbf{R}} Y^g + \tau \Lambda \left(\frac{\mathbf{f}^{DEF}}{\mathbf{f}^g} \circ \frac{1}{\mathbf{f}^g} \right) \right] d\mathbf{f}^g = \tau \frac{\mathbf{f}^{DEF}}{\mathbf{f}^g} - \mu^g \mathbf{1} - Y^g \underline{\mathbf{R}} \underline{\mathbf{S}}_{\mathbf{N}}^{-1} (Y^g \underline{\mathbf{R}} \mathbf{f}^g - \mathbf{N}) \quad (13)$$

Multiplying by the inverse of the matrix inside the $[\]$ gives $d\mathbf{f}^g$. Where $\tau \Lambda \left(\frac{\mathbf{f}^{DEF}}{\mathbf{f}^g} \circ \frac{1}{\mathbf{f}^g} \right)_{ij} = \tau \delta_{ij} \frac{f_i^{DEF}}{(f_i^g)^2}$ is a diagonal matrix. Without this, the matrix on the left inside the square brackets will be singular when $\underline{\mathbf{R}}$ is underdetermined, and cannot be inverted normally.

The drawback of a linear-approximation algorithm such as this one is that it will overshoot; and sometimes it may overshoot into the negative flux region. To avoid this unphysical result, an underrelaxation constant should be applied:

$$x^g = |\min(\frac{df_i^g}{f_i^g}) \forall df_i^g < 0| \quad (14)$$

$$w^g = \min(\frac{\alpha}{x^g}, 1) \quad (15)$$

where x is a scalar denoting largest absolute fractional error among all negative fractional errors.

$$\mathbf{f}^{g+1} = \mathbf{f}^g + w^g d\mathbf{f}^g$$

The larger α is, the faster the algorithm will converge. However the higher order terms (O) in $\frac{f_i}{f_i + df_i} = 1 - \frac{df_i^g}{f_i^g} + O\left(\frac{df_i^g}{f_i^g}\right)$ will start to dominate when α is allowed to approach 1. Therefore to keep the absolute value of the residuals less than the 1st order approximation itself, i.e. $\left| \frac{O\left(\frac{df_i^g}{f_i^g}\right)}{\left(\frac{df_i^g}{f_i^g}\right)} \right| < 1$, an $\alpha < 0.5$ was chosen.

3.1.2 Finding the optimal point in the regularization function to counteract the current loss function gradient

An alternative method, not requiring Taylor expansion, is to use cancel out the “residual steepness” by accounting for only the change induced by the the regularization function:

$$\left(\nabla \text{reg}(\mathbf{f}^g + d\mathbf{f}^g) - \nabla \text{reg}(\mathbf{f}^g) \right) + \epsilon^g = \mathbf{0} \quad (16)$$

Since $[\nabla \text{reg}(\mathbf{f})]_i$ is only dependent on f_i , we can find the exact value of ∇reg^{-1} easily even though it is a non-linear function; and we do so without applying matrix inversion operations, making it a much less resource-intensive and scalable algorithm.

$$f_i^g + df_i^g = \frac{\tau f_i^{DEF}}{\left((Y^g \underline{\mathbf{R}} \mathbf{f}^g - \mathbf{N}) \cdot \underline{\mathbf{S}}_{\mathbf{N}}^{-1} (Y^g \underline{\mathbf{R}}) \right)_i + \mu^g} \quad (17)$$

The appropriate step size that minimizes is given as w^g

$$\mathbf{f}^{g+1} = \mathbf{f}^g + w^g d\mathbf{f}^g$$

where w^g can be found by setting $\frac{d\text{loss}(\mathbf{f} + w^g d\mathbf{f})}{dw^g} = 0$, simplifying by assuming $Y^{g+1} = Y^g$ has no w^g dependence, and Taylor expanding when necessary, to obtain

$$w^g = \frac{-\sum_i \left(\frac{\tau f_i^{DEF}}{f_i^g} \right) \left(\frac{df_i^g}{f_i^g} \right) - 2(Y^g \underline{\mathbf{R}} \mathbf{f}^g - \mathbf{N}) \cdot \underline{\mathbf{S}}_{\mathbf{N}}^{-1} \underline{\mathbf{R}} d\mathbf{f}^g}{-2\sum_i \left(\frac{\tau f_i^{DEF}}{f_i^g} \right) \left(\frac{df_i^g}{f_i^g} \right)^2 + 4(Y^g)^2 d\mathbf{f} \cdot \underline{\mathbf{R}}^T \underline{\mathbf{S}}_{\mathbf{N}}^{-1} \underline{\mathbf{R}} d\mathbf{f}^g}$$

3.2 Accounting for systematic error

In order to account for the uncertainty in the response matrix due to the ambiguity of the cross-sections, we can append an extra term into the loss function, such that Lagrangian expression becomes

$$\begin{aligned} \mathcal{L}(\underline{\mathbf{R}}, \mathbf{f}, Y, \mu) = & \frac{\gamma}{2} (\underline{\mathbf{R}} - \underline{\mathbf{R}}_0) \cdot \underline{\mathbf{S}}_{\mathbf{R}}^{-1} (\underline{\mathbf{R}} - \underline{\mathbf{R}}_0) + \frac{1}{2} (Y \underline{\mathbf{R}} \mathbf{f} - \mathbf{N}) \cdot \underline{\mathbf{S}}_{\mathbf{N}}^{-1} (Y \underline{\mathbf{R}} \mathbf{f} - \mathbf{N}) \\ & + \tau \mathbf{f}^{DEF} \cdot \ln \mathbf{f} + \mu \|\mathbf{f}\|_1 \end{aligned}$$

where $\underline{\mathbf{S}}_{\mathbf{R}}^{-1}$ is a rank 4 tensor, and is the tensor inverse of the covariance tensor of $\underline{\mathbf{R}}$.

It should only be populated along the “diagonal”, i.e. $(S_R^{-1})_{k,i,k',j} = \delta_{k,k'} (S_R^{-1})_{k,i,k',j}$ where δ is the Kronecka delta.

This simply adds an extra step into the iterative calculations, which is expressed as follows in Einstein notation

$$\left(\gamma \underline{\mathbf{S}}_{\mathbf{R}}^{-1} + (Y \mathbf{f}) \otimes \underline{\mathbf{S}}_{\mathbf{N}}^{-1} \otimes (Y \mathbf{f}) \right) \underline{\mathbf{R}} = \gamma \underline{\mathbf{S}}_{\mathbf{R}}^{-1} \underline{\mathbf{R}}_0 + \mathbf{N} \cdot \underline{\mathbf{S}}_{\mathbf{N}}^{-1} \otimes (Y \mathbf{f}) \quad (18)$$

Multiplying both side by the inverse of the tensor in the bracket on the left leaves only $\underline{\mathbf{R}}$, allowing $\underline{\mathbf{R}}^{(g+1)}$ to be computed.

3.3 Error propagation

The covariance ($\underline{\underline{\mathbf{S}}}_{\mathbf{f}}$) on the solution vector(\mathbf{f}) of a loss-function minimization problem, such as this one, is simply the inverse of the Hessian matrix of the loss function:

$$\underline{\underline{\mathbf{S}}}_{\mathbf{f}} = \underline{\underline{\mathbf{H}}}^{-1}$$

$$H_{ij} = \frac{\partial^2(\text{loss})}{\partial f_i \partial f_j} = -\tau \delta_{ij} \frac{f_i^{DEF}}{f_i^2} + \left(Y(\mathbf{R}^T)_i + \frac{\partial Y}{\partial f_i} \underline{\underline{\mathbf{R}}} \mathbf{f} \right) \cdot \underline{\underline{\mathbf{S}}}_{\mathbf{N}}^{-1} \left(Y(\mathbf{R}^T)_j + \frac{\partial Y}{\partial f_j} \underline{\underline{\mathbf{R}}} \mathbf{f} \right)$$

where

$$\frac{\partial Y}{\partial f_i} = \frac{\left[N \underline{\underline{\mathbf{S}}}_{\mathbf{N}}^{-1} \mathbf{R} \right]_i}{(\underline{\underline{\mathbf{R}}} \mathbf{f}) \cdot \underline{\underline{\mathbf{S}}}_{\mathbf{N}}^{-1} (\underline{\underline{\mathbf{R}}} \mathbf{f})} - \frac{2 \left(N \cdot \underline{\underline{\mathbf{S}}}_{\mathbf{N}}^{-1} (\underline{\underline{\mathbf{R}}} \mathbf{f}) \right) \left[(\underline{\underline{\mathbf{R}}} \mathbf{f}) \cdot \underline{\underline{\mathbf{S}}}_{\mathbf{N}}^{-1} \underline{\underline{\mathbf{R}}} \right]_i}{\left((\underline{\underline{\mathbf{R}}} \mathbf{f}) \cdot \underline{\underline{\mathbf{S}}}_{\mathbf{N}}^{-1} (\underline{\underline{\mathbf{R}}} \mathbf{f}) \right)^2}$$

Alternatively, if one is concerned about the total flux ϕ instead:

$$H_{ij} = \frac{\partial^2 \text{loss}}{\partial \phi_i \partial \phi_j} = \delta_{ij} \frac{f_i^{DEF}}{Y^2} \frac{\tau}{f_i^2} + (S_{\phi}^{-1})_{ij} \quad (19)$$

where $\phi = Y \mathbf{f}$ and $\underline{\underline{\mathbf{S}}}_{\phi}^{-1} = \underline{\underline{\mathbf{R}}} \cdot \underline{\underline{\mathbf{S}}}_{\mathbf{N}}^{-1} \underline{\underline{\mathbf{R}}}$

When $\underline{\underline{\mathbf{R}}}$ is also allowed to vary, the Hessian matrix extends from having $n \times n$ dimension to having $(n + m \times n) \times (n + m \times n)$ dimension. The extra elements in the Hessian is written as follows:

$$\frac{\partial^2 \text{loss}}{\partial R_{ki} \partial f_j} = \quad (20)$$

$$\frac{\partial^2 \text{loss}}{\partial R_{ki} \partial R_{k'j}} = \quad (21)$$

↑ Those equations will be filled in later.

The inverse of the Hessian matrix will then return the covariance.

4 Usage

Note to self:

- L-curve
- If the *a priori* coincides with the true spectrum, then there will be exactly zero drift as the τ increase.
- The level of drift increases wrt. deviation of the τ .
- can observe using the L-curve.
- we should not stop at $\chi^2 = 1$

5 Benchmarking/Verification

6 Using multiple a priori/assigning covariance onto a priori

References

- [1] Kenneth P Burnham, Kenneth P. Burnham, and David Raymond Anderson. Model selection and multimodel inference : a practical information-theoretic approach. Springer, New York, London, 2nd ed edition, 2002.
- [2] J Cvachovec and F Cvachovec. Maximum likelihood estimation of a neutron spectrum and associated uncertainties. Advances in Military Technology, 1(2):67–79, 2008.
- [3] Fatma Zohra Dehimi, Abdeslam Seghour, and Saddik El Hak Abaidia. Unfolding of neutron energy spectra with fisher regularisation. IEEE Transactions on Nuclear Science, 57(2):768–774, 2010.
- [4] Shigetaka Maeda and Tetsuo Iguchi. A new unfolding code combining maximum entropy and maximum likelihood for neutron spectrum measurement. Journal of Nuclear Science and Technology, 50(4):381–386, 2013.
- [5] M. Matzke. Unfolding procedures. Radiation Protection Dosimetry, 107(1-3):155–174, 2003.
- [6] Jan Mlynar, John M. Adams, Luciano Bertalot, and Sean Conroy. First results of minimum fisher regularisation as unfolding method for jet ne213 liquid scintillator neutron spectrometry. Fusion Engineering and Design, 74(1):781 – 786, 2005. Proceedings of the 23rd Symposium of Fusion Technology.
- [7] M. Reginatto and P. Goldhagen. Maxed, a computer code for maximum entropy deconvolution of multisphere neutron spectrometer data. Health Phys., 77(5):579–83, 1999.
- [8] Marcel Reginatto, Paul Goldhagen, and Sonja Neumann. Spectrum unfolding, sensitivity analysis and propagation of uncertainties with the maximum entropy deconvolution code maxed. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 476(1-2):242–246, 2002.
- [9] Wolfgang von der Linden. Maximum-entropy data analysis. Applied Physics A, 60(2):155–165, 1995.
- [10] Ross Worrall, Chantal Nobs, and Lee W Packer. A review and development of unfolding methodologies with respect to nuclear fusion based activation detectors. UKAEA.

Appendices

A Uniqueness of global minimum