Analysis and Implementation of the Maximum Likelihood Expectation Maximization Algorithm for FIND

Angus Boyd Jameson
University of New Hampshire

Follow this and additional works at: https://scholars.unh.edu/student_research

Part of the Applied Statistics Commons, Atomic, Molecular and Optical Physics Commons, Data Science Commons, Nuclear Commons, and the Probability Commons

Recommended Citation

This Undergraduate Research Project is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Student Research Projects by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact Scholarly.Communication@ unh.edu.
ANALYSIS AND IMPLEMENTATION OF THE MAXIMUM LIKELIHOOD EXPECTATION MAXIMIZATION ALGORITHM FOR FIND

UNIVERSITY OF NEW HAMPShIRE
DEPARTMENT OF PHYSICS AND ASTRONOMY

PHYSICS BACHELOR OF SCIENCE DEGREE

Author: Angus B. Jameson
Advisor: Dr. Mark McConnell
Co-advisor: Dr. Karl Slifer

Durham, 12 2020
Abstract

This thesis presents an organized explanation and breakdown of the Maximum Likelihood Expectation Maximization image reconstruction algorithm. This background research was used to develop a means of implementing the algorithm into the imaging code for UNH’s Field Deployable Imaging Neutron Detector to improve its ability to resolve complex neutron sources. This thesis provides an overview for this implementation scheme, and include the results of a couple of reconstruction tests for the algorithm. A discussion is given on the current state of the algorithm and its integration with the neutron detector system, and suggestions are given for how the work and results of this project could be continued and expanded upon.

1 Introduction

Today the usage of radioactive and fissile material is becoming more and more widespread in industrial sectors. This means that it easier than ever for someone to obtain some of this material and rig it into something harmful like a Radiation Dispersal Device or even a nuclear weapon. The Field-Deployable Imaging Neutron Detector (FIND) developed by the University of New Hampshire (UNH) [9] seeks to give homeland security agents and similar groups access to a compact and portable device that can be used to detect rogue nuclear radiation sources that were could be the the result of hazardous nuclear waste or even attempts by a terrorist cell to jury rig nuclear devices [8].

![Diagram of the most recent version of the FIND system. The system also includes a tablet interface and display](image)

FIND’s detection system consists of two $3 \times 3$ arrays of stilbene scintillators each of which is read out by a silicon photomultipliers (SiPM’s). The method by which FIND detects incident neutrons is the double scattering technique. While this form of neutron detector is by no means novel, the FIND program aims to produce a neutron detector that can easily be carried and used during field operations. To this end the
current total weight of the detector system is 25 kg (55 lbs) and uses an easily portable tablet display [6].

The next generation of FIND, iFIND, seeks to build on this foundation by adding more scintillators to each scatter layer as well as add four additional scattering layers arranged in a cube formation giving it a near omni-directional field of view [7]. Despite these improvements the underlying physics of the detector remain the same. As a result, while the contents of this thesis were developed with FIND in mind, all code and conclusions could be applied to iFIND with relatively minor adjustments.

1.1 Back-Projection and Imaging Imitations

As mentioned previously, FIND uses the double scattering technique for neutron detection. In this technique a neutron travels some distance and enters one of the detector’s scintillators. The neutron will then n-p scatter off a proton, depositing some of its energy in the form of light which is measured by the accompanying SiPM [8]. The neutron then scatters towards the second layer of scintillators and experiences a similar reaction. Based on the location of these two scattering interactions \((r_1, r_2)\), the energies they deposit \((E_1, E_2)\) and the time of flight (TOF) between them, the incident energy of the neutron \((E_n)\), and the angle \((\phi)\) between the incident trajectory \((S_o)\) and the scattered trajectory \((S_c)\) can be reliably calculated. Since \(S_o\) can have any orientation about \(S_c\), the possible origin of the neutron reduces to points along the surface of a cone with half angle \(\phi\) (event cone) as seen in fig. 2b.

\[
E_n = E_1 + K \cdot E, \\
\sin^2 \phi = \frac{E_1}{E_1 + K \cdot E}
\]

In the accompanying imaging software for FIND, these cones are projected onto a 2D plane as event circles [6]. This image plane is parallel to scintillator planes and at a predefined distance \((z_n)\) from the detector. As more emitted neutrons double scatter events within FIND, more event circles are projected on the image plane, which should all overlap on the location of the neutron source. This method of imaging is referred as the back-projection method.

While in theory the overlap of just there of these event circles should be enough to determine the source location, systematic error in the detector as well as the presence of neutrons from background sources make it far more practical to instead generate a distribution map of event circle intersections for hundreds or even thousands of event circles. For the purpose of generating this distribution the image plane is divided into

![Figure 2: Breakdown of the double scatter and back-projection method.](image)
Figure 3: An idealized example of back projection. In practice the distribution peak would not be this sharp.

26 × 20 pixels. Note that the physical size of these pixels depends on the image plane’s distance from the detector. Fig. 4a shows an example of an image generated in this way, where the distribution peak indicates a neutron source location.

Figure 4: (a) Image result for the pure back projection of 7182 event circles. (b) Image result of the likelihood response map algorithm of a similar neutron source. Values under 10% of the maximum were zeroed (white color). Note that these two images are not created from the same data set even though their sources (Cf-252) were the same. At the top top of each image (from right to left) is the particle type of the distribution (gamma/neutron), the data collection period of the data, the number of data counts, and sampling rate (counts per second) of the data collection. Images are overlayed onto a photo of the detector’s field of view taken from FIND’s mounted camera.

Before moving forwards, it is important to understand the two different coordinate systems that are used in FIND’s imaging code. First, the pixels of the image plane are each assigned a dimensionless coordinate index ranging from (0 − 24) × (0 − 19), where the origin, (0, 0), is the bottom left-most pixel (see fig. 5a). Second, the locations of each detector element and image pixels are identified by a set of three dimension Cartesian coordinates. The origin, (0 cm, 0 cm, 0 cm) is located at the center of the center scintillator in the first array (see fig. 5b). The take away from this is that a pixel in the
image plane can be identified either by a Cartesian position or by a unitless coordinate index.

![Diagram of the pixel and Cartesian coordinate systems](image)

Figure 5: Diagrams for the pixel and Cartesian coordinate systems. The origin is marked on both as well as some points of interest.

A typical image of the back-projection distribution could potentially consist of thousands of intersecting event circles. This produces images like that seem in fig. 4a. However, this is not what a user would see on the tablet interface. Instead FIND incorporates an additional imaging algorithm which is designed to make the distribution map appear sharper and more centered around the neutron source location. In the current version of FIND, this imaging algorithm utilizes a likelihood response map to achieve this effect. Fig 4b shows an example of the images produced by this algorithm.

The back-projection method has been shown to be a reliable method of locating sources of fissile neutrons. However, it does have some limitations as to the types of sources that it can identify. Back-projection works best when identifying symmetric and point sources like neutron sources. However, for more complex sources back-projection can struggle to discern the physical characteristics of the source. Two examples most relevant to this thesis are (a) radiation sources in close proximity will appear as one larger source in the image and (b) non-symmetric sources can appear far less defined in the image, potentially even appearing to be point sources. Example of these limitations are shown in fig. 6. This means that while, for example, homeland security agents may be able to use FIND to locate a rouge neutron source they would not know how many neutron sources to expect and any subtleties in the source that could help identify it would be indiscernible.

A solution to this limitation would be to implement a new imaging algorithm which can iterate on the back-projected distribution in order to resolve these more complex sources. One algorithm known to be able to preform these types of image reconstructions is the **Maximum Likelihood Expectation Maximization (MLEM)** algorithm, also referred to as the Maximum Likelihood and Expectation Maximization algorithm [12]. The objective of this thesis was to research the underlying physics of the MLEM algorithm, develop a form of the algorithm that is applicable in the case of neutron double scattering in the FIND detector, and implement the algorithm into the existing FIND imaging code. Tests were then conducted to ensure the new imaging code runs and is accurate (see Section 3.3).
2 Maximum Likelihood Expectation Maximization

2.1 Background

The MLEM algorithm is an iterative image reconstruction algorithm that is used in astronomy and medical imaging [14]. This algorithm takes an initial model for a source distribution (e.g. the back-projected distribution) and iterates on it to produce a better approximation of the true source distribution [12]. This process can be repeated as many times as needed to produce the desired image.

The principle behind the MLEM is that for every pixel in the image a new value is calculated based on the previous value, the probability that a neutron emitted from the pixel would be detected by FIND, and the likelihood of that pixel being the source of all the event circles that intersect it. This process can be summarized as

\[
\text{(for each pixel)} \quad \text{New value} = \text{Previous value} \times \frac{\text{Likelihood of being source of intersecting event circles}}{\text{Probability of neutron being detected}}.
\]

This creates a new distribution where the differences between pixel values are weighted such that, in the image, the peak(s) appear more concentrated around the neutron source(s) [13]. Fig. 7 shows a good example of what this looks like.

2.2 MLEM Formula

The MLEM algorithm has a number of special and general equations [3]. Given that the data used in FIND’s imaging code is a list of double scattering event interactions, the
Figure 7: An example of the MLEM reconstruction process for a Cs\textsuperscript{137} radiation source. (A) is the pure back-projected image, and (B), (C), and (D) are the distribution reconstructed after 2, 10, and 40 MLEM iteration [12].

The main MLEM formula used in this thesis was the List-Mode MLEM formula:

\[
\lambda_{j}^{n+1} = \frac{\lambda_{j}^{n}}{s_{j} \sum_{i \in j} t_{ijj}^{n}} \sum_{j' \in i} t_{ijj'}^{n} \lambda_{j'}^{n}
\]

(1)

where \(i\) is an index of the events/event circles, \(j\) is an index for the pixel coordinate positions, \(i \in j\) is the subset of all event circles where an event circle \(i\) intersects pixel \(j\) (see fig. 8a), \(j \in i\) is the subset of all pixels where a pixel \(j\) is intersected by event circle \(i\) (see fig 8b), \(t_{ij}\) is the probability that an event \(i\) was caused by a neutron emitted from pixel \(j\), \(s_{j}\) is the probability that a neutron emitted from pixel \(j\) is detected by FIND (sensitivity function), and \(\lambda_{j}^{n}\) is the distribution value of pixel \(j\) at iteration \(n\) [12]. The List-Mode formula is useful for FIND because it attempt to reconstruct the image from a list of measured values, rather the by parameterizing an likelihood function for some observed variable [5].

There are some features this formula that are worth elaborating on. For example, during testing of the MLEM algorithm as described in Section 3.3, the following was always observed to be true:

\[
s_{j} < \frac{t_{ijj}^{n}}{\sum_{j' \in i} t_{ijj'}^{n} \lambda_{j'}^{n}} \Rightarrow \lambda_{j}^{n} < \lambda_{j}^{(n+1)} \text{ (for all } j)\]

Figure 8: (a) Visual depiction of the set \(i \in j\). (b) Visual depiction of the set \(j \in i\).
This means that every pixel’s value in the distribution increases after each iteration of the algorithm (though pixels near the distribution peak will increase a lot more). As a result of this, it would seem that for \( n > 0 \), \( \lambda^n_j \) no longer indicates the number of event circles intersecting pixel \( j \) for iteration \( i \). Instead, \( \lambda^n_j \) becomes a unitless weighted value, which only conveys information when in the context of the image distribution. This MLEM reconstruction performed by Zhang et. al. (2014) in fig. 9 supports this idea.

Figure 9: (a) Distribution formed through the back-projection of emission detections from a dual source. (b) Distribution formed after 50 iterations of the MLEM. Note how the MLEM can resolve two sources from a seemingly equal distribution. It clear to see from this how each part of distribution has increased over the iterations. This is most apparent in dark blue background of the distribution, which range from 0-50 before at 0 iterations and 0-100 after 50 iterations [14].

Also, since the List-Mode MLEM formula is proportional to \( 1/s_j \), pixels with a low detector sensitivity (i.e. pixels near the edge of the detector’s field of view) that have a large number of event circles intersecting them, will be weighted more significantly in the next distribution than pixels with a high sensitivity (i.e. pixels near the center of the field of view) intersected by an equal number of event circles. This is presumably because a large number of overlapping event circles is less likely to be the result of random background sources at the edge of the field of view than at the center.

Equation (1) at first glance appears to be fairly simple equation to implement. It only has three independent terms, one of which, \( \lambda^n_j \), is just the initial distribution. However, \( t_{ij} \) and \( s_j \) are not simple terms to derive. The following sections will explain how the equations use for these terms were obtained and give a justification for their validity in the case of neutrons double scattering in FIND.

### 2.3 Probability (\( t_{ij} \))

Section 2.2 states that the term \( t_{ij} \) is the likelihood that a double scattering event \( i \) was caused by a neutron originating from pixel \( j \). However, it is easier to create a formula for \( t_{ij} \) if we reverse the wording of this definition. Therefore, we will instead consider \( t_{ij} \) to be the probability that a neutron originating from pixel \( j \) will be emitted in such a way that it experiences the double scattering reaction observed in event \( i \).
Before continuing, it is necessary to elaborate on what physical information event $i$ and pixel $j$ carry. As explained in Section 3, the $i$ carries the information of $(E_1, r_1, E_2, r_2, \text{TOF})$, although TOF was not used to calculate $t_{ij}$. On the other hand, the coordinate position of pixel $j$ only carries information for the Cartesian coordinate position $r_n = (x_n, y_n, z_n)$ (see fig. [10]) in the pixel (note that $z_n$ is the same for every pixel).

![Figure 10: A breakdown of the trajectories of a typical double scattering reaction.](image)

A number of assumptions were made about the emitted neutron in order to simplify the final equation for $t_{ij}$. These assumptions were

- The probability that the neutron will be emitted with the correct incident energy, $E_n$, calculated for event $i$ is 1.
- The probability that the neutron will be scattered, absorbed, or otherwise impeded when travelling through air is 0.
- All possible incident and scattered trajectories are considered to be equally likely.
- The position of a recorded scattering event for a scintillator is considered to have occurred at the center of the scintillator (i.e. $d_1 = d_{s1} = d_{s2} = \frac{b}{2}$).

With these assumptions, the formula of $t_{ij}$ was simplified to

$$t_{ij} = (P_{S_o}) (P_{d_1}) (P_{E_1}) (P_{S_c}) (P_{d_{s2}}) (P_{E_2})$$

(2)

where $(P_{S_o})$ is the probability that a neutron is emitted from pixel $j$ in the direction of the correct first layer scintillator for event $i$, $(P_{d_1})$ is the probability that the neutron scatters at the correct depth, $d_1$, in the first scintillator, $(P_{E_1})$ is the probability that the neutron will deposit the same amount of energy in the first scintillator as was recorded for the event, $(P_{S_c})$ is the probability that the neutron will then be scattered in the direction of the correct second layer scintillator, $(P_{d_{s2}})$ is the probability that the neutron doesn’t
scatter a second time in the first scintillator and does scatter at the correct depth \( d_{s2} \) in the second scintillator, and \( (P_{E_2}) \) is the probability that the neutron will deposit the same amount of energy in the second scintillator as was recorded for the event. The rest of this section will be dedicated to breaking down how each of these individual probabilities were determined to be calculated.

**Incident Trajectory Probability \((P_{S_o})\)**

The probability of a neutron being emitted towards the first scintillator \((P_{S_o})\) can be estimated by comparing the area of the scintillator the area subtended by the solid angle of all possible incident neutron trajectories at a given distance. Said another way, the probability of a hit can be calculated by comparing projectile’s spread to the size of the target. From this the following equation is obtain (note that the solid angle is \(4\pi\) as the neutron could potentially be emitted in any direction):

\[
(P_{S_o}) = \frac{h^2}{4\pi S_o^2},
\]

where, \(h\) is the length of a cubical scintillator element (5.175 cm [9]), \(S_o\) is the distance between the pixel and first scintillator \((|\overrightarrow{r_n} - \overrightarrow{r_1}|)\), and \(4\pi S_o^2\) is the area subtended by the solid angle of all incident neutron trajectories.

**First Scattering Probability \((P_{d_1})\)**

The probability of a neutron scattering at a certain depth in a material has been well explored and documented. It is therefore easy to find that the equation for \((P_{d_1})\) should be of the form

\[
(P_{d_1}) = e^{-\Sigma(E_n)d_1} \Sigma(E_n)dr,
\]

where \(d_1\) is the recorded scattering depth of \(i\)’s first scattering event (assumed to be \(\frac{h}{2}\)), \(dr\) is the thickness of the area we expect the neutron to scatter in, and \(\Sigma(E)\) is the macroscopic scattering cross-section for the stilbene scintillator material for a neutron with energy \(E\) [cm\(^{-1}\)]. See Section 2.4 for an explanation of \(\Sigma(E)\) and how its values were obtained.

**First Deposited Energy Probability \((P_{E_1})\)**

Here \(e^{-\Sigma(E_n)d_1}\) contributes the probability of the neutron reaching a depth of \(d_1\) in a material without scattering, while \(\Sigma(E_n)dr\) contributes the probability that the neutron does scatter within a thickness \(dr\) of material, which is taken to be a small value (1 cm) [4]. Putting these two expressions together gives the probability of a neutron scattering at the desired depth in the scintillator.

The probability of a neutron depositing a certain amount of energy \((P_{E_1})\) in he first scintillator is complicated to calculate exactly. However, from Zhang et. al (2014) it is claimed that “according to the kinematic equations describing the double-scattering event” this probability can be estimated by the ratio of the deposited energy to the incident energy [14]. Therefore, a valid estimation for the probability is

\[
P_{E_1} = \frac{E_1}{E_n}.
\]
Scattered Trajectory Probability ($P_{s_n}$)

The probability that a neutron is scattered towards the second scintillator ($P_{s_n}$) can be calculated in a similar manner to ($P_{s_n}$), however the possible scattered trajectories are limited by the angle $\phi$ which from fig. 2 is know to be given by $E_n$ and $E_1$. Therefore, the solid angle of all possible scattered trajectories is given by to be $2\pi(1-\cos \phi)$. Thus the probability equation is

$$P_{s_n} = \frac{h^2}{2\pi(1-\cos \phi)S_c^2},$$  \hfill (6)

where $S_c$ is the distance between the first and second scintillators ($|\vec{r}_1 - \vec{r}_2|$).

Second Scattering Probability ($P_{d_{n2}}$)

For ($P_{d_{n2}}$), the probability is obtained the same way as in $P_{d_{n1}}$, however the probability that the neutron doesn’t scatter twice in the first scintillator must allow be considered. This leads to the equation

$$P_{d_{n2}} = e^{-\Sigma(E_n-E_1)(d_{n1}+d_{n2})}\Sigma(E_n-E_1)dr,$$  \hfill (7)

where $d_{n2} = h - d_{n1} = \frac{h}{2}$ and $d_{n2}$ is the desired depth for the neutron to scatter in the second scintillator ($\frac{h}{2}$).

Second Deposited Energy Probability ($P_{E_2}$)

Finally, an estimation for ($P_{E_2}$) is obtained using the same method as in ($P_{E_1}$). This estimation is valid, as ($P_{E_2}$) assumes all previously described events have happened, so the situation is identical to the one for ($P_{E_1}$) but with a different incident energy and desired deposited energy. Thus the final probability equation is

$$P_{E_2} = \frac{E_2}{E_n - E_1}.$$  \hfill (8)

Putting everything together yields a final equation for $t_{ij}$ of

$$t_{ij} = \frac{h^2}{4\pi S_c^2} e^{-\Sigma(E_n)d_{n1}} e^{\Sigma(E_n)dr} \frac{E_2}{E_n - E_1}.$$  \hfill (9)

This is a rather long and complex equation, however there is a very powerful simplification that can be made with the main MLEM formula.

In equation (9), $t_{ij}$ depends on the indexes $i$ and $j$. If $t_{ij} = uv_{i}w_{ij}$, where $u$ is all constant terms, $v_{i}$ is all terms only depend on the index $i$, and $w_{ij}$ are terms that depend on both $i$ and $j$, then the following algebraic simplification can be applied to equation (8).

$$\lambda_{j}^{n+1} = \frac{\lambda_{j}^{n}}{s_{j}} \sum_{v_{ij}} \frac{w_{ij}v_{ij}}{\sum_{j'\epsilon i} w_{ij}v_{ij}'\lambda_{j'}^{n}} = \frac{\lambda_{j}^{n}}{s_{j}} \sum_{v_{ij}} \frac{w_{ij}}{\sum_{j'\epsilon i} w_{ij}'\lambda_{j'}^{n}}$$  \hfill (10)

so that any term in $t_{ij}$ that is constant over the summation $\sum_{j'\epsilon i}$ can be eliminated from the algorithm calculation.

From equation (8), recalling what information is carried in the $j$ index, and considering fig. 2, it is clear that only the terms $S_o$ and $\phi$ depend on $j$. Thus all those terms can be eliminated from the calculation. However, note that the summation $\sum_{j'\epsilon i}$ as described in Section 2.2 is the summation over all pixels intersected by a single event circle $i$. By
geometry, all these pixels will have the same \( \bar{\phi} \) value, and thus \( \bar{\phi} \) will be constant over the summation as well.

This leaves \( S_o \) as the only term that will not be eliminated in this simplification. Thus equation (11), when equation (9) and this simplification are incorporated, becomes

\[
\lambda_{n+1}^j = \frac{\lambda_n^j}{s_j} \sum_{i \in j} \frac{1}{S_o^2} \lambda_n^j/S_o^2.
\]  

This simplification greatly reduces the computation time for the algorithms. In one instance the computation time per iteration was reduced from 30 minutes to 30 seconds!

### 2.4 Scattering Cross Sections

The term \( \Sigma \) in equations (4) and (7) is the macroscopic cross-section (also sometimes referred to as the linear attenuation coefficient [14]). This is a parameter that describes the probability of a particle interacting in a material at a given depth [10]. Note that despite being a cross-section, \( \Sigma \) has units of \([\text{m}^{-1}]\).

Generally, \( \Sigma \) refers specifically to the total macroscopic cross-section, which is the combination of the macroscopic scattering/elastic and macroscopic absorption/nonelastic cross-section for a given material \( (\Sigma = \Sigma_s + \Sigma_a) \) [10]. These are probability parameters for scattering and absorption interactions respectively. Given that FIND involves a double scattering interaction, in this thesis, \( \Sigma \) will exclusively refer to the macroscopic scattering cross section.

The macroscopic scattering cross-section of a material is dependent on the makeup of the material and the energy \( (E) \) of the incident particle (e.g. a neutron). However, an equation for this relationship is not easily found. This is because it can differ depending on the type of incident particle, the target material, and the type of interaction [1]. This makes \( \Sigma \) unsuitable for an analytical calculation, creating the necessity for a data analysis method.

It was still difficult to find data file of \( \Sigma \) measurements for different materials. As a result, values of \( \Sigma \) needed to be derived from recorded microscopic scattering cross section measurements. The microscopic cross section \( (\sigma) \) \([\text{m}^2]\) is a probability parameter for an interaction between a particle and a single nucleus of a material [10]. The Microscopic and macroscopic cross-sections are related by the simple equation

\[
\Sigma = N \sigma,
\]  

where \( N \) is the atomic density of the target. However, this only valid for a material purely of one element. For a material with a molecular structure containing multiple elements, \( \Sigma \) can be calculated as the sum of the macroscopic cross-sections of each component element \( (\Sigma = \Sigma_1 + \Sigma_2 + \Sigma_3 + \ldots) \). Stilbene has a molecular formula of \( \text{C}_{14}\text{H}_{12} \) [2]. Therefore, the following function can be obtained

\[
\Sigma(E) = \Sigma_c(E) + \Sigma_H(E) = \frac{\rho N_A n_c}{M} \sigma_c(E) + \frac{\rho N_A n_H}{M} \sigma_h(E) = \frac{\rho N_A}{M} [14 \sigma_c(E) + 12 \sigma_h(E)],
\]  

where \( N = \frac{\rho N_A n}{M} \), \( \rho \) is the density of the material, \( N_A \) is Avogadro’s number \((6.0221409 \times 10^{23})\), \( n \) is the number of atoms per molecule, and \( M \) is the molar mass\(^1\) of the material. For stilbene \( \rho = 0.971 \text{ g/cm}^3 \) and \( M = 180.250 \text{ g/mol} \) [2].

\(^1\) McFarland (2000) states that \( M \) is a molecular mass, but data analysis suggests it is a molar mass.
From equation (13), it is possible to convert numerically evaluated values of the microscopic cross-section for carbon and hydrogen. The data sets for this calculation were obtained from Evaluated Nuclear Data Files (ENDF) for 1-H and 6-C, provided by the Los Alamos National Laboratory [1]. ENDF are data set for neutron cross-section for various material. See ”An Introduction to the ENDF Formats” by McFarland (2000) for more information on ENDF’s and how to find and extract data from them [11]. As will be explain in Section 3 a linear interpolation was conducted on the ENDF’s whenever a macroscopic cross-section was needed for a particular energy. Note that the ENDF cross-sections are recorded in barns $[1\text{barn} = 10^{-24}\text{cm}^2]$.

2.5 Sensitivity Function ($s_j$)

Zhang, et. al. (2014) claims that the sensitivity function can be calculated via the equation $s_j = \sum_{ij} t_{ij}$ (i.e. the probability of neutron detection is the sum of the likelihood of each event originating from pixel $j$). While this is presumably applicable in the case of many back-projections, testing of this method has shown that this is not applicable in general for FIND (see fig. 11). As a result a new calculation method for $s_j$ was developed for this project, based on the methods shown in Tornga (2010).

The sensitivity function ($s_j$), represents the probability of a neutron emitted from pixel $j$ being detected by FIND. To be detected by FIND a neutron must double scatter in the scintillator layers and survive a pulse shape discrimination test [8]. For the purposes of calculating ($s_j$) the pulse shape was not considered.

From this definition, $s_j$ can be thought of as the probability that a neutron emitted from pixel $j$ will double scatter in any pair of scintillators. Using this same line of logic, $t_{ij}$ from Section 2.3 can be thought of as the probability that a neutron emitted from pixel $j$ will scatter in a particular pair of scintillators ($\vec{r}_1, \vec{r}_2$) and deposit a particular amount of energy ($E_1, E_2$) in each scintillator.

Base on these two definitions, it is clear that $s_j$ can be approximated by summing values for $t_{ij}$ over every possible pairing of scintillators. From equation (9), a summation over all scintillator positions ($\vec{r}_1, \vec{r}_2$) is applied,

$$s_j = \sum_{\vec{r}_1} h^2 \frac{4\pi S_o(\vec{r}_n, \vec{r}_1)^2}{4\pi S_o(\vec{r}_1, \vec{r}_2)^2} e^{-\Sigma(E_n)dr_1} \Sigma(E_1)dr_1 e^{-\Sigma(E_n)dr_2} \Sigma(E_1)dr_2 e^{-\Sigma(E_n-E_1)(d_{1+4}+d_2)\Sigma(E_n-E_1)dr},$$

where

$$S_o(\vec{r}_n, \vec{r}_1)^2 = |\vec{r}_n - \vec{r}_1|^2,$$
$$S_o(\vec{r}_1, \vec{r}_2)^2 = |\vec{r}_1 - \vec{r}_2|^2.$$
\[
\cos[\bar{\phi}(S_o, S_c)] = \frac{S_{o,x}S_{c,x} + S_{o,y}S_{c,y} + S_{o,z}S_{c,z}}{S_o S_c},
\]

\(\sum_{r1}^{}\) and \(\sum_{r2}^{}\) are the summations over the positions all nine first and second layer scintillators, and probability \((P_{E2})\) is omitted since amount of energy deposited by a neutron does not affect whether or not it is detected. Note that \((P_{E1})\) is not omitted as \(\bar{\phi}\) is related to \(E_1\) as seen in fig. 2a.

There is a problem with this estimation which should be immediately obvious. The \(i\) index in \(t_{ij}\) carried values for the energies \(E_1, E_2,\) and \(E_n\). The sensitivity function, \(s_j\), does not have this index and thus there is no source for the energy values. A discussion of this problem could not be found during the research of this thesis, and even papers like Tornga (2010) that discuss the sensitivity function in detail do not address this issue. In light of this, an original solution was devised where an average would be calculated for the energies base on detector data. These averages would then be used in the calculation of equation (14) as a mean to obtain a sensitivity function value for a typical neutron emitted by the source. Section 3 discusses this process in greater detail.

3 Implementation

3.1 Overview of Methods

The objective of this thesis project was to successfully implement the MLEM algorithm as an imaging algorithm on top of the existing back-projection method for the FIND system. In the current version of the FIND there is a pipeline of Python 3 programs where the raw data from the detector is fed into through of series of programs which returns an image of source distribution overlayed on top of a picture taken by FIND’s mounted camera (see fig 4 in Section 1 for an example). The program where most of the work for this project was done was called GenScat.py, which is the program that handles image generation after being given the following input variable calculated by prior programs:

Input variables: \((E_n, \bar{\phi}, S_c)\).

The first thing done to implement the new MLEM algorithm into genplot.py, was to remove the previous FIND imaging algorithm (a likelihood response map), which could not resolve more complex imaging, as well as everything that other imaging technique besides the back-projection and an undercutting of all values bellow 10% (note this undercutting is the reason for the translucent white areas of the distribution seen in the FIND images of this thesis). Doing this created a blank slate where the MLEM could implemented and tested without needing to be concerned about the effects of other imaging techniques. Once this was done, the bulk of the work done for this thesis was spent on implement the equations (11) and (14) into the code.

For equation (11), a method for obtaining subsets \(i \epsilon j\) and \(j \epsilon i\) for the summations had to be created. This was achieve by creating a \(26 \times 20\) array of a empty arrays (events) and \(N\) length list of empty lists (pixels), where \(N\) is the total number of neutron events recorded by the detector. A system was then set up where whenever the code determined that a certain pixel in the imagine plane with coordinate index \([X,Y]\) (recall fig. 5a) is intersected by a event circle generated from the back projection of the \(n\)th recorded neutron event, the ordered pair \((X,Y)\) would be added to the list in pixels with index
\[ n - 1 \] (note: \( n - 1 \) is used here since in Python 0 is the first index position), and \( n - 1 \) would be added to the array in pixels with index \([X, Y]\). The idea was that after back-projection imaging was done for every event, events could be given a pixel coordinate index, \([x, Y]\), and return the list index, \([n - 1]\), for every event circle that intersected that event: \((i\epsilon j)\) (refer to fig. 8a for reference). Similarly, pixels could be given the list index for an event circle and return the index of every pixel that was intersected by the event: \((j\epsilon i)\) (refer to fig. 8b for reference). From these the summations of the main MLEM equation could be properly calculated.

In addition, it was realized that \( S_0 = |\vec{r}_n - \vec{r}_1| \) could not immediately be calculated as the scintillator positions are not part of the input variables. As a result programs earlier in the pipeline called get scat from filtered.py had to be edited so that the list of event data being given to GenScat.py would now be

Input variables: \((E_n, \bar{\varphi}, S_c, r_1, r_2)\).

This work was done by Jason Legere who worked on the FIND project.

For \( r_n \) a system was created where during GenScat.py’s initialization the physical location for each pixel with in the Cartesian space as described in Section 1 were calculated. This was done bu utilizing the following formula:

\[
\begin{align*}
    r_n = & \left[ z_n \cos \left( 90 - \frac{f_{ov_x}}{2} \right) \frac{2X - 25}{26}, z_n \cos \left( 90 - \frac{f_{ov_y}}{2} \right) \frac{2Y - 19}{20}, z_n \right], \\
\end{align*}
\]

(16)

where \( z_n \) is the distance between the image plane and the detector, \( f_{ov_x} \) and \( f_{ov_y} \) are the fields of view for FIND in the \( x \) and \( y \) directions (80.5° and 59.1° respectively), and \( X \) and \( Y \) are the pixel index numbers for the \( x \) and \( y \) directions. This equation was used to create a Python library where each pixel position was pair with its coordinate index.

For equation (14), the sensitivity function, the implementation was done as a separate function which equation (1) would call on. Since it does not depend on any event data (i.e. input variables), the positions \((r_1, r_2)\) of each scintillator had to be determined and recorded in a list which could then be used in the summations of the equation. This was done using very similar method to the method used to record \( r_n \), where a library was created and the position of each scintillator was pair with a 2D coordinate index.

As mentioned in Section 2.4, a data analysis method is needed to obtain values for \( \Sigma(E) \) in equation (14). ENDF’s from the Los Alamos National Laboratory containing microscopic neutron cross-section data at different energies for C-12 and H-1 were downloaded. GenScat.py would then read these files into lists. Whenever a calculation of equation (14) was done, a linear interpolation was conducted to obtain the microscopic cross-section values for the desired energy \( (E) \). These values were then converted from \([\text{barns}]\) to \([\text{cm}^{-1}]\), and given to equation (13) to obtain a value for \( \Sigma(E) \) in units of \([\text{cm}^{-1}]\).

This explanation does raise the question of what values to use for energy. In Section 2.5 in was briefly mentioned sensitivity function \((s_j)\) varies only on pixel position \( r_n \) so energy values \( E_n \) and \( E_1 \) should be constant in equation (14), despite being variable quantities. The ideal solution would be to calculate \( s_j \) by integrating over the probability distribution for neutron detection at various energies. However, such an integration is not simple for reasons explained in the next section, this solution was not implemented before the project’s completion.

Instead, an estimation of the true value for the sensitivity function was implemented. This estimation involved calculated an average incident energy \( E_{n,\text{avg}} \) by averaging all \( E_n \).
values in the list of event data. The energy deposited in the first scintillator \( E_1 \) was calculated by the equation

\[
E_1 = E_n \sin^2 \tilde{\phi},
\]

where \( \tilde{\phi} \) is calculated geometrically from equation \((15)\). This way a sensitivity function value for the most typical emitted neutron is calculated.

To test this approximation, a distribution map was for the sensitivity function values was created for a neutron source 3 meters from FIND. The results of this test can be seen in fig. 12. Not only is the shape of the distribution inline with what is expected, but the values are on the order on \( 10^{-7} \). Considering that the neutron source used for these test was \(^{252}\text{Cf} \), which has an emission rate of between \( 10^7 \) – \( 10^9 \) emission per second [citation needed] and FIND had an emission rate of 7.5 detections per second, this is within the right ball park of what is expected for sensitivity function values.

![Image](image.png)

Figure 12: Test of the sensitivity function. The image plane was set to be 3 meters away from the detector. Not the center maker in the image background not aligned with the center of the detector’s field of view.

### 3.2 Complications

There were a number of complications that occurred up over the course of this project, which forced the scope of this thesis to be gradually to be changed or scaled back. For example, the MLEM algorithm was not the original topic for this thesis, however, the sudden outbreak of COVID-19 in spring of 2020 and the move to online classes in March of 2020 resulted in the continuation of the original project being untenable. The implementation of the MLEM algorithm into FIND’s imaging code was chosen as the new focus of the thesis in part due to the fact that work could be done on it without access to any UNH facilities. This sudden shift in focus meant that, even with work being done over the 2020 summer break, this thesis would have a shorter period for research and work than originally planned.

Compounding this issue for time were several setbacks that occurred with the research of equation for \( t_{ij} \), \( s_j \), and \( \Sigma(E) \). In one instance, a formula for \( t_{ij} \) and \( \Sigma(E) \) was created based on the equations from Tornega (2010). It was later discovered that Tornega’s
equation’s are only applicable to Compton scattering photon and not n-p scattering neutrons, resulting in weeks of work needing to be entirely removed.

The development of equation (9) also had a major roadblock in the derivation of the probability $P_{Sc}$ (equation (6)). In Zhang et. al. (2014), it is stated that $P_{Sc}$ depends on a delta function $\delta(\phi - \phi_{scat})$, where $\phi_{scat}$ “is the angle between direction ($\vec{r}_n - \vec{r}_1$) and ($\vec{r}_1 - \vec{r}_2$)”. This delta function was integrated over a probability distribution in order to calculate $t_{ij}$, requiring a complex estimation technique to evaluate. All attempts to replicate this for the FIND imaging code resulted in distribution of the sensitivity function appearing like that seen in fig. 13, rather than a single peak in the center of the detector’s field of view. Several weeks were spent attempting to implement $P_{Sc}$ as described by Zhang et. al. (2014) before the effort was abandoned and equation (6) was developed.

Figure 13: Example of a test of the sensitivity function for the implementation of $t_{ij}$ as defined by Zhang, et. al. (2014) [13].

It should be mentioned that equation (6) was not taken from an accredited article, but rather is was derived from the logic that the equations for $(P_{So})$ and $(P_{Sc})$ should have similar structures as they are both probabilities that the neutron follows a trajectory that intersects with a target. This was supported by a reasonable distribution shape and values of the sensitivity after implementation as described in Section 3.1.

Originally, the scope this project included exploring the possibility of a back-projection technique using voxels in a 3D imaging space, rather than pixel on a 2D imaging plane, conducting live tests of FIND with the MLEM algorithm, and even implementing developing an implementation for the MLEM algorithm into the imaging code for iFIND. However, due to the reduced work period and the previously mentioned complications, all of these had to be dropped from the thesis. The most unfortunate consequence was that there was not enough time to obtain a test data-set for a complex source distribution (e.g. two nearby point sources). Thus, it remains unconfirmed if the current implementation of the MLEM algorithm in the imaging code is sufficiently able to resolve multiple distribution peaks of a complex source or if there is still more work that needs to be done.

3.3 Testing and Results

Despite not having the necessary data to test the MLEM algorithm on a complex source distribution, it was still valuable to test the algorithm on single, point-like source. The logic being that if the algorithm can not properly resolve a simple source, it will not be able to resolve a more complex source. To this end, two test were conducted on the algorithm: one where the neutron source was near the center of the detector’s field of view and one where the neutron source was in the upper right hand corner. This way it could be checked if the algorithm can resolve sources both near the center and at the
edge of the detector’s field of view. The result of this test can be seen in fig. [14] and [15].

![Figure 14: Test for the MLEM algorithm on a center neutron source (\(^{252}\text{Cf}\)).](image)

![Figure 15: Test for the MLEM algorithm on an off center neutron source (\(^{252}\text{Cf}\)).](image)

In these figures, images created by 1, 3, and 7 iterations of the MLEM algorithm are compared with the pure back-projected image and the image of FIND’s previous likelihood response map imaging algorithm. It is immediately apparent that the distributions produced by the MLEM are not as smooth as the likelihood imaging method. This is because additional imaging techniques in the future. The peaks of the distributions, however, are more condensed around the neutron source especially at higher iterations. In practice this reduction in the width of a peak could be the difference between identifying a new radiation source is in the building and identifying that the radiation source is in the second floor of a building.

As for comparison between the MLEM iterations, the distribution peaks become increasingly focused on the perceived neutron source in back-projected data. The MLEM only focuses in on this point, rather than generating one or more new peaks in different parts of the image plane. The MLEM also does not cause the peak to drift or split up into two or more distinct peaks after only a handful of iterations. The peak for the centered source does begin to separate after 7 iterations, demonstrating that too many iterations may create problems in the image distribution.

These tests also show that the MLEM algorithm requires a good initial distribution to properly reconstruct the source. Fig. [16] gives a better look at the algorithm’s test on the off-axis source. While the MLEM algorithm properly reconstruct a point-like source in the location suggested by the initial back-projected distribution, this was not true location. This indicates that reconstruction power of the algorithm has limitations. As an example, if a back-projected distribution from two nearby neutron sources looked identical to the distribution of a single source, the algorithm might resolve it as a single source. This is an important limiting factor of the MLEM algorithm that should be carefully considered if it is to be incorporated in the final version of FIND or iFIND.

Overall, this test shows that the MLEM algorithm implemented in the FIND imaging code behaves exactly as expected for single point sources. While further tests will need to be done to prove that this will be as accurate at resolving complex sources, this test at least shows that the methods and equations discussed in this thesis are capable of resolving a simple source distribution from a good enough initial set of data.
4 Conclusion

As discussed in the previous section testing of the MLEM algorithm shows that it is accurate to the initial data, and return a tighter distribution of the source location than the previous imaging techniques used. Despite this, rigorous test has demonstrated that the MLEM is not a flawless method of reconstruction. The accuracy of the reconstructed image is heavily dependent on how well the initial data reflects the true source distribution. Also, while each iteration can be calculated in less than a minute, a complex distribution may require 10 or iterations to resolve [12]. Such a lengthy calculation time may be impractical for a detector designed to be use by agents in the field.

Combining this with the fact that the algorithm has yet to be tested on the types of complex source distributions it was chosen for, it is obvious that the MLEM algorithm is not yet ready for a proper integration into the official imaging codes of FIND or iFIND. That said, the findings of this thesis suggest that the MLEM algorithm has the potential to be a powerful imaging algorithm for FIND. Therefore, it is the opinion of this researcher that the work done for this thesis be used as a foundation from which future projects could continue to develop, test, and refine the algorithm as it is currently implemented.

At some point, the current MLEM algorithm will need to be tested on a complex source distribution. The easiest type of source distribution to test would be that of two nearby point-like sources. A researcher performing this test may find that the algorithm is not yet capable of resolving multiple sources and more work need to be done on either the algorithm’s main equations (1), (9), and (14) or its implementation in the code. Equation (9) in particular should be scrutinized, as several simplifying assumptions were made to obtain it.

However, even if the algorithm is able to successfully resolve multiple sources, more tests would need to be done to get an idea of the algorithm’s power and limitations. For example, one could test how close the two sources could be before the MLEM algorithm becomes incapable of resolving them as two sources. Also, as the MLEM requires a good initial distribution, a testing should be done to determine what, on average, is the minimum number of back-projections required create an initial distribution which the algorithm can resolve accurately.

As the goal of this thesis was only to properly implement the MLEM algorithm into the imaging code, there is a lot of opportunities to future projects to improve on the algorithm’s implantation to enhance its effectiveness. Such improvements are necessary beyond simply increasing efficiency and usability, as FIND already has a working imaging algorithm in its current version. While the MLEM algorithm may work better than the
current likelihood response map method with certain types of source distribution, whether or not the MLEM algorithm can reconstruct images faster, more accurately, or with less noise will affect whether the algorithm is chosen or replace or merely supplement the current imaging techniques.

One example of a way of improve the current version of the MLEM algorithm would be to develop a method of automatically determining the number algorithm iterations that is most efficient in terms of distribution quality and computation time. Currently, the imaging code requires that the user manually input the number of iterations the algorithm should go through. This could be something that is done as part of a larger project, or it could prove complex enough to be its own project.

Once it is decided how the MLEM should be incorporated into the official FIND system, it will likely be an entirely separate project to transfer this imaging method to the newer and still under development iFIND project. iFIND works via the same detection method as FIND, so much of the work done in with this thesis would carry over to iFIND, but much care and effort would still need to be done to ensure a smooth conversation between the two detectors’ imaging codes. This would be the ultimate end goal of this line research, which could potentially contribute to the success of the iFIND project as it tries to gain attention and funding.

Acknowledgements

This thesis project was conducted over a seven month period and represents the culmination of over a decade of my education, research, and dedication. While it is a shame that outside factor, such as the 2020 quarantine, resulted in the not every initial goal of project being met, I am proud of what I have accomplished and the obstacles I have over come. I can only hope that my work will lay the groundwork for later students and aspiring physicists to build on my findings or further their own projects. One day I would love to say that the work I did for this thesis leads to a feature being incorporated in a device used to assist people in protecting the safety of others.

I would like to thank my advisor, Professor Mark McConnel, and co-advisor, Professor Karl Slifer, for their guidance on this project during this hectic and confusing time in our recent history. I would like to thank Jason Legere and Kevin M. Mello for their help in introducing me to the FIND project and assisting my work when requested. I would like to thank Peter Bloser for helping me understand his imaging code and laying the groundwork for my implementation of the MLEM algorithm. And I would like to thank the University of New Hampshire for two and a half years of valuable education and learning experiences.
References


