Progress Update on Iterative Reconstruction of Neutron Tomographic Images, 2019

Paul Hausladen
Matthew Blackston
Jens Gregor

September 26, 2019

OAK RIDGE NATIONAL LABORATORY

MANAGED BY UT-BATTELLE FOR THE US DEPARTMENT OF ENERGY

Isotope and Fuel Cycle Technology Division

PROGRESS UPDATE ON ITERATIVE RECONSTRUCTION OF NEUTRON TOMOGRAPHIC IMAGES, 2019

Paul Hausladen
Matthew Blackston
Jens Gregor

September 26, 2019

CONTENTS

FIGURES

TABLES

ACRONYMS

2D	two-dimensional
3D	three-dimensional
ART	algebraic reconstruction technique
DU	depleted uranium
FY	fiscal year
HDPE	high-density polyethylene
MLEM	maximum likelihood expectation maximization
ORNL	Oak Ridge National Laboratory
PET	positron emission tomography
PSF	point spread function
SIRT	simultaneous iterative reconstruction technique
SPECT	single photon emission tomography
TV	total variation
WLS	weighted least squares

ACKNOWLEDGMENTS

SCOPE

The fiscal year (FY) 2019 update of this report satisfies the technical deliverable "Simultaneous reconstruction of simultaneous observables" for project OR16-ML-3DTomography-PD3Jb, "3D Tomography and Image Processing Using Fast Neutrons." At its inception, this project had two overall goals. The first of these goals was to extend associated-particle fast neutron transmission and, particularly, induced-reaction tomographic imaging algorithms to three dimensions. The second of these goals was to automatically segment the resultant tomographic images into constituent parts and then extract information about the parts, such as the class of shape and potentially shape parameters.

This report addresses the component of the project concerned with three-dimensional (3D) image reconstruction and represents the cumulative progress to date. In particular, it addresses the ability to decompose neutrons arriving at times consistent with transmission into transmission and small-angle scattering. Because incremental progress in each year builds on the previous year's progress and end-of-year status, progress is most usefully recorded in a cumulative document where incremental progress is reported on a year-by-year basis. Consequently, this report constitutes an update of the last 3 years' reports [1, 2, 14]. The present year's progress is detailed in Section 7.

1. INTRODUCTION

Most readers are familiar with the notion, if not the details, of computed tomography from medical physics. In computed tomography, projection data from a number of angles are used to reconstruct cross-sectional images of patients for diagnostic purposes. The crucial elements of this process are that measured data divide the object (patient) into lines of response, and multiple views through the object are mathematically combined to estimate an image. Computed tomography, and particularly x-ray computed tomography, has been commonplace in medicine for decades, but emission tomography, such as positron emission tomography (PET) and single photon emission computed tomography (SPECT), also sees wide diagnostic use. For emission tomography, iterative reconstruction is often used to obtain the best possible images from data whose quality may be limited by a number of factors, including counting statistics, detector resolution limitations, and the number of projections.

Although x-ray computed tomography, PET, and SPECT have existed for some time, Oak Ridge National Laboratory (ORNL) recently developed a suite of novel fast neutron induced-reaction (induced fission and elastic scatter) imaging techniques to characterize objects of interest [3, 4]. At the outset of the present project, transmission and induced-reaction image reconstruction codes used for this purpose existed as a set of Matlab scripts. These codes are based on the maximum likelihood expectation maximization (MLEM) formalism [5] and were developed to demonstrate proof of concept. The scripts are functional, but they have a variety of drawbacks. Foremost among these is the limitation that the scripts can only reconstruct two-dimensional (2D) images, and they are sufficiently slow that extension to three dimensions would make them unusable in practice. This limitation is contrary to the inherently 3D nature of induced-reaction images whose data have contributions from points in the reconstructed plane to detectors that are out of plane. In addition, the scripts cannot correctly weight errors and are not readily suited to the addition of penalties, constraints, or other priors to select for desirable image properties.

The intent of the present work is to combine modern parallel iterative reconstruction codes with novel associated-particle fast-neutron induced-reaction (induced fission and elastic scatter) and transmission imaging techniques recently developed by ORNL. In this way, the increased speed of reconstruction will make routine use of 3D reconstruction of fast neutron images practical. In particular, an increase of reconstruction speed of approximately 2 orders of magnitude will make 3D reconstructions approximately

as time-consuming as the present 2D reconstructions. The necessary increases in speed will be realized through the use of

- compiled (not interpreted) code,
- sparse matrices (to avoid performing operations on a large number of zero-value elements),
- multi-threaded architecture to take advantage of multicore computers, and
- ordered subsets to increase convergence speed and enable use of fewer iterations.

Ordered subsets enable updating the guessed solution based on a fraction of the data and generally speed convergence by the number of such subsets. Moreover, once the iterative reconstruction machinery is in place, it can be used, along with modifications of the update equation, to accommodate other algorithms that correctly weight errors for data that are background subtracted or that select for favorable solutions by the use of regularization, penalties, or constraints.

2. ASSOCIATED-PARTICLE IMAGING

In the associated-particle method, fast (14 MeV) neutrons are made in a deuterium–tritium neutron generator using the $d + t \rightarrow \alpha + n$ reaction. Because the alpha particle and neutron are emitted simultaneously in (nearly) opposite directions, detection of the time and position of the alpha particle determines the time and direction of the associated neutron. A fast timing, position-sensitive alpha-particle detector inside the deuterium–tritium neutron generator records the time and position of alpha particle detections, as shown in Figure 1. Likewise, a position-sensitive fast neutron detector records the time and position of neutrons. In the neutron data, transmitted, scattered, and induced neutrons are identified by their arrival times, positions, and number.

Transmission imaging is performed when no interactions occur, and the transmitted neutron is detected on the other side of the object. The transmitted neutrons decrease with thickness of the object like $e^{-\mu x}$. A transmission radiograph gives an image of μx along each path through the object from the source to the detector pixel, and the radiograph is extracted from measured data by taking the logarithm of the ratio of the number of neutrons with and without the object present. In instances where multiple views through an object can be combined, tomographic reconstruction of the transmission data provides the value of the attenuation coefficient, μ, for each pixel in a slice through the object. The interpretation of the value of the attenuation coefficient is the probability per unit length of the neutron interacting. Fast neutron images have contrast because different materials have different values of μ, which originates from different total nuclear cross sections or different number densities of nuclei. Neutron transmission imaging is similar to x-ray transmission imaging in principle, but neutrons are much better at penetrating high atomic number materials than even the highest energy x-rays and they maintain contrast for low atomic number materials, such as hydrogenous materials.
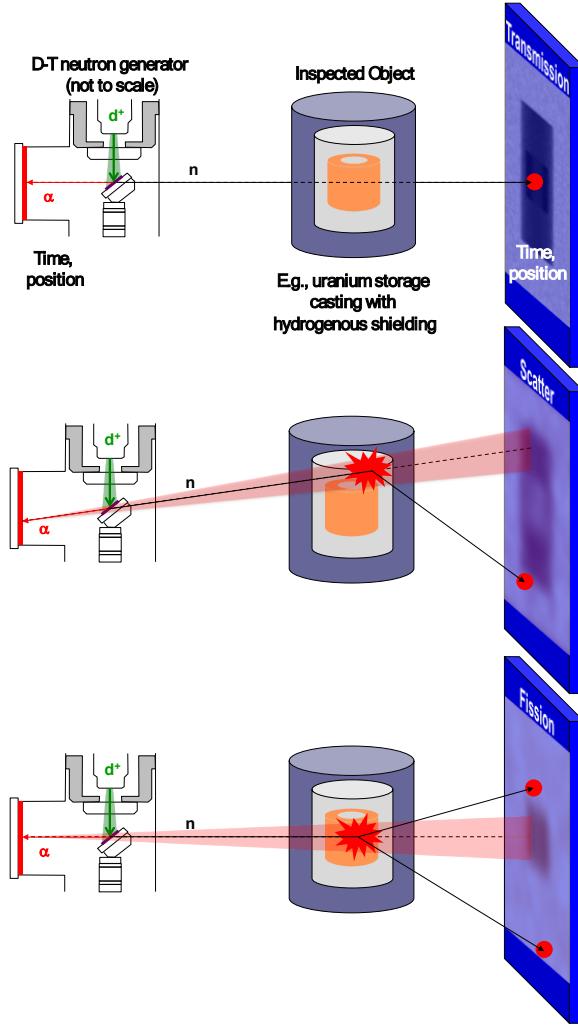
Figure 1. Schematic diagram of tagged 14 MeV neutrons used for (a) transmission imaging, (b) elastic scatter imaging, and (c) induced-fission imaging.

Elastic-scatter imaging is performed when a single elastic scatter on hydrogen takes place, and the scattered-but-transmitted neutron is detected at an appropriate angle and time on the other side of the object. The population of elastic single scatters varies with the thickness of the object approximately as μ_s $x \cdot e^{-\mu x}$, where μ_s is the attenuation coefficient due to elastic scatter on hydrogen. Like transmission imaging, the interpretation of the value μ_s is the probability per unit length that a neutron elastically scatters from hydrogen. An elastic-scatter radiograph produces an image of the number of detected scatter counts for each emitted alpha along each direction from the alpha detector through the object. In instances where multiple views through an object can be combined, tomographic reconstruction of the transmission data gives the value of the attenuation coefficient, μ_s, for each pixel in a slice through the object. This imaging technique differs from transmission imaging in that each point along each ray from the source toward the detectors has a different efficiency for contributing a count to the measured scatter data. As a result, this efficiency must be explicitly calculated and include the attenuation from the source to the image pixel, the average attenuation from the image pixel to the detector (weighted by the solid angle of each detector pixel), the probability of scatter at a particular angle, the relative efficiency at the scattered

3

energy, and the probability of detection at the appropriate angle-time gate for the timing and direction resolution.

Induced-fission imaging is performed by counting neutrons at arrival times consistent with fission-energy neutrons. Like scatter imaging, fission neutrons are created with probability per unit length $\mu_f \overline{v}_{chain}$, where μ_f is the attenuation coefficient for fission (probability of fission per unit length) and $\overline{v}_{chain}$ is the average number of neutrons emitted in the subsequent fission chain. This property depends on the constituent materials as well as the overall geometry. Likewise, fission neutron pairs are created with probability per unit length $\mu_f \overline{v(v-1)}_{chain}$, where $\overline{v(v-1)}_{chain}$ is the average number of neutron pairs emitted in the fission chain. Induced fission images can be constructed based on induced neutrons or induced neutron pairs. Induced-fission imaging is similar to elastic-scatter imaging in that each point along each ray from the source toward the detectors has a different efficiency for contributing a count to the measured induced neutron data. Again, this efficiency must be explicitly calculated. For neutron singles, it includes the attenuation from the source to the image pixel, the average attenuation from the image pixel to the detector (weighted by the solid angle of each detector pixel), and the detection efficiency for fission-spectrum neutrons to fall within the bounds of the fission time gate. For neutron pairs, an additional calculation is performed to see whether the relative position and time of the two neutrons satisfies a cut to eliminate interdetector scattering.

3.   ITERATIVE RECONSTRUCTION

Suppose an image **x** with pixel values x_j and the corresponding measured projection data (or sinogram) **y** with values y_i are connected by a system matrix A such that $Ax = y$, or equivalently, $y_i = \sum_j a_{ij} x_j$. Then, image reconstruction is the process of solving for x given y and A. Generally speaking, A will be ill conditioned, making the explicit solution a poor choice because it will compound errors in y. There are a number of methods for solving the problem iteratively of which several are implemented. However, all require the ability to "forward project" a guessed image to get a trial sinogram.

Forward projection of a trial image to get a calculated sinogram is illustrated in Figure 2, where the values of one row of the system matrix A correspond to a ray that originates at the source and terminates on a particular detector pixel. In the case of transmission imaging, forward projection corresponds to summing the attenuation values along that ray.

(a) (b) (c) $\sum_j a_{35,j} x_j = y_{35}$ (d)

System matrix $\mathbf{A} = [a_{ij}]$ Image $\mathbf{x} = [x_j]$ One projection   Sinogram $\mathbf{y} = [y_i]$
row 35 $a_{35,j}$
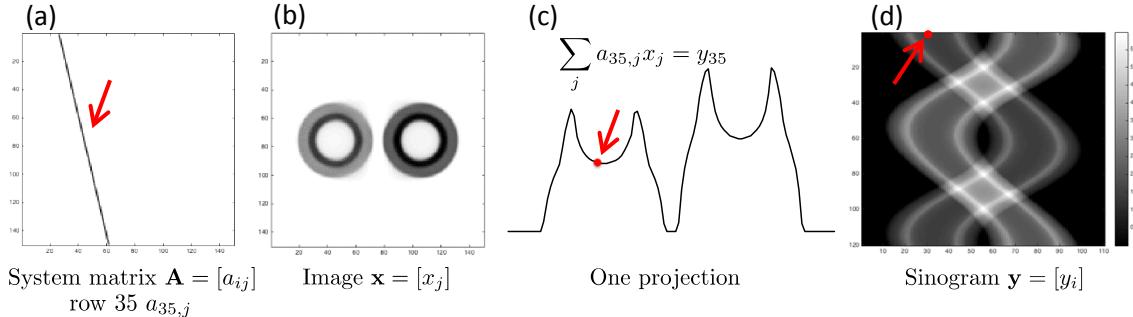
Figure 2. Forward projection to get a trial sinogram. For instance, the values of one row of the system matrix sums the attenuation values along a path through the object that terminates on a detector pixel.

For transmission imaging, the system matrix can be fairly simple, but for induced-reaction imaging, it can be quite complex. However, most of the complexity is encoded in an efficiency ε_{ij}, the components of which were enumerated in the previous section. In addition, the ray connecting the source and detectors is

4

spread. This is shown pictorially in Figure 3, where the smeared ray is shown in (a), and the efficiency is shown on a logarithmic color scale in (b).

(a)   (b)   (c)

Coefficients l_{ij} Efficiency ε_{ij} Image $\mathbf{x} = [x_j]$

System matrix $\mathbf{A} = [a_{ij}] = [l_{ij}\varepsilon_{ij}]$

Projection $\mathbf{Ax} = \sum_j a_{ij}x_j = \sum_j l_{ij}\varepsilon_{ij}x_j$
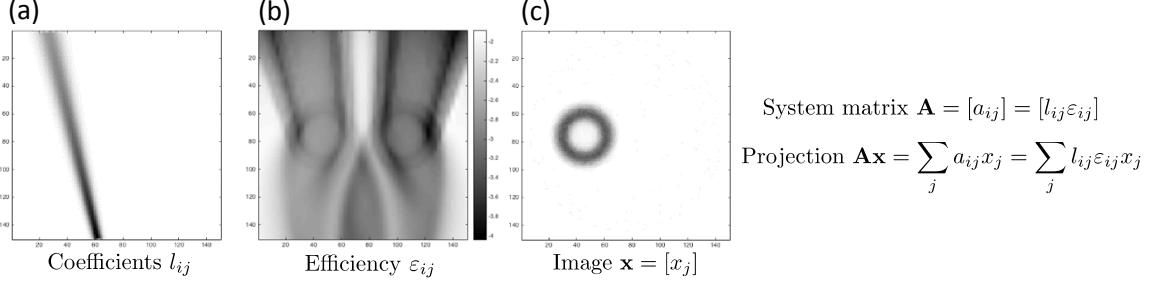
Figure 3. Forward projection for induced-reaction imaging which includes (a) a spread ray between the source and detector and (b) a complicated efficiency map that encodes the induced-reaction physics. The resulting reconstructed induced-reaction image is shown in (c).

The Matlab scripts for performing image reconstruction were based on the maximum likelihood expectation maximization algorithm. That algorithm is derived from formally maximizing the log likelihoods, assuming the y values to be Poisson distributed about the solution Ax. This maximization results in an update equation for the $k + 1^{st}$ image values $x_j^{(k+1)}$, which are multiplicative.

$$x_j^{(k+1)} = x_j^{(k)} \frac{1}{\sum_i a_{ij}} \sum_i \frac{y_i}{\sum_h a_{ih}x_h^{(k)}} a_{ij}$$

In words, the kth trial image is forward projected to get a trial sinogram. The ratio of the measured-to-calculated sinograms is then back projected to update the image. The derivation of the update equation is only valid if the y_i are Poisson distributed. Unfortunately, the transmission and induced-reaction values are not Poisson distributed. In the former case, the values of the transmission sinogram are not counts but the logarithm of a ratio of counts. In the latter case, the induced counts are background subtracted. Nonetheless, we have applied MLEM as a general multiplicative algorithm for image reconstruction.

To handle errors in the data consistently, a least squares formalism is more appropriate. In this case, image reconstruction corresponds to finding an image, x, that minimizes the Euclidian norm

$$\frac{1}{2}\|Ax - y\|^2.$$

This minimization results in an additive update equation:

$$x_j^{(k+1)} = x_j^{(k)} + \alpha \frac{1}{\sum_i a_{ij}} \sum_i \left(y_i - \sum_h a_{ih}x_h^{(k)}\right) a_{ij}$$

This is a gradient descent method known as the simultaneous iterative reconstruction technique (SIRT). Restated, the kth trial image is forward projected to produce a trial sinogram. The difference between the measured and calculated sinograms, scaled by a value α chosen to accelerate convergence, is then back projected to update the image. In matrix notation, this can be written as

$$x^{(k+1)} = x^{(k)} + \alpha A'\left(y - Ax^{(k)}\right).$$

Here, A' denotes the transpose of A. A more general form of this update equation enables penalizing undesirable solutions using a matrix Q, a weighting matrix W (a diagonal matrix that contains the data variances), and a preconditioning matrix $\tilde{C}$ that ensures convergence.

$$x^{(k+1)} = (I - \alpha\beta\tilde{C}Q'Q)x^{(k)} + \alpha\tilde{C}A'W(y - Ax^{(k)}) \qquad (1)$$

The parameter β is a user-defined parameter that controls the degree to which minimization of the data fidelity term drives the computation versus the desire to regularize the solution. The preconditioning matrix $\tilde{C}$ has the following values:

$$\tilde{C} \equiv \mathrm{diag}\left\{\frac{1}{\sum_i w_i\left(\sum_h a_{ih}\right)a_{ij}}\right\}.$$

4. YEAR 1 PROGRESS

The original plan for this project stipulated development of a fully 3D transmission imaging code in the first year. However, after initiating work, it was considered more prudent to implement both transmission imaging and induced-reaction imaging in 2D before adaptation to 3D. This course of action enabled any aspects of the induced-reaction reconstruction that were not consistent with the implementation used for transmission imaging to be identified.

At the end of FY 2016, a 2D iterative reconstruction code for transmission, hydrogen scatter, and induced fission (singles and doubles) was implemented as compiled C code. This code shared a common framework for forward projection, back projection, and calculating efficiencies that was readily extensible to 3D. Updated equations were implemented based on the MLEM algorithm as well as SIRT [6], a gradient descent method for solving least squares problems. Using the SIRT method enabled weighting by appropriate errors. The 2D code reduced reconstruction times for transmission images by more than 2 orders of magnitude. For induced reaction images, reconstruction times are dominated by the time to calculate the induced-count efficiencies over the reconstruction volume. The duration of these calculations was also reduced by approximately 2 orders of magnitude.

The SIRT algorithm can be used with or without weighting. In the first year, weighted reconstruction was implemented using weight values calculated from the data, assuming the variance to be equal to the expected counts (i.e., to be proportional to the negative exponential of the projection value). This weighting scheme is only appropriate for transmission imaging that does not include subtractions. In future years, weighted reconstruction will be implemented using user-supplied variances. In this way, the user-supplied values can include the increase in variance from background subtraction.

In the first year, the following values of the regularization Q were implemented:

1. no regularization, or $Q = 0$;

2. minimum norm regularization, or $Q = I$ and $\|Qx\|^2 = \sum_j x_j^2$; and

3. finite differences regularization, or $\|Qx\|^2 = \sum_j \sum_{k \in N_j} (x_j - x_k)^2$, where N_j denotes the set of lexicographical predecessor neighbors (i.e., neighboring image pixels).

In FY 2016, the forward projector for the induced-reaction reconstruction codes did not include a point spread function (PSF), or smearing on the ray from the source to detectors. That is, it uses the ray shown in Figure 2(a) rather than that shown in Figure 3(a). Implementation of the PSF was left for future years.

As implemented, the speed performance of the compiled 2D reconstruction code increased more than 2 orders of magnitude over the Matlab script. As a measure of reconstruction time, the example sinogram shown in Figure 2(d) was reconstructed using the Matlab script and the compiled C code. The sinogram consisted of 110 rays by 120 projections. For the reconstruction, the sinogram was oversampled so that 440 rays sampled the reconstruction volume.

Table 1. Speed improvement of the new compiled code

	Matlab script, MLEM, 40 iterations	C code, MLEM, 40 iterations, 1 core	C code, MLEM, 40 iterations, 4 cores	C code, MLEM, 10 iterations, 4 ordered subsets, 4 cores
Time (s)	286	13.6	4.2	1.1
Improvement	N/A	21.0	68.4	267

In addition to a factor of ~20 from efficient, compiled code, there are additional factors roughly proportional to the number of cores and the number of ordered subsets. In total, this test indicates that future 3D reconstructions will take about the same amount of time that 2D reconstructions previously took.

Comparison images derived using the (a) Matlab MLEM reconstruction code for 40 iterations and the (b) compiled C code MLEM reconstruction for 10 iterations and 4 ordered subsets are shown in Figure 4. Note that the conversion between image pixels and centimeters is slightly different in each case, but visual inspection indicates that the result is essentially identical.
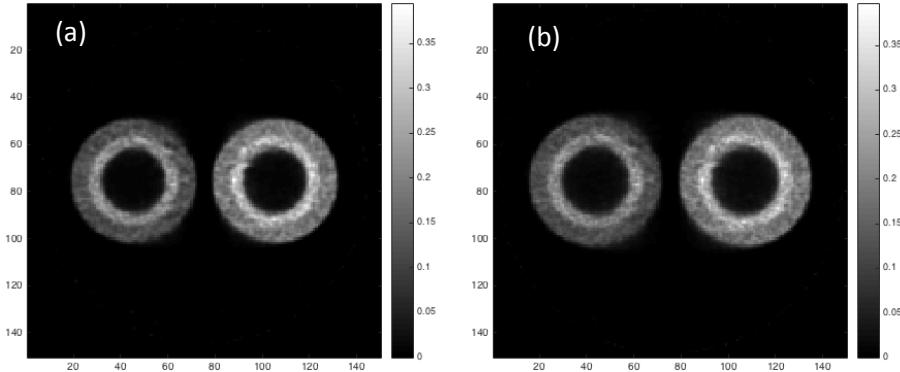
Figure 4. Visual comparison of outputs of the (a) Matlab MLEM reconstruction script and (b) compiled C code MLEM reconstruction.

5. YEAR 2 PROGRESS

In FY 2017, a number of improvements to the reconstruction software were implemented. These improvements include

- extension of the reconstruction code to 3D,

- the ability to specify the variance of each data point (given that the data points are background-subtracted differences or log ratios),

- implementation of a PSF in the forward and back projection operators for induced-reaction imaging,

- implementation of a total variation constraint to achieve smoother images while preserving sharp edges [7], and

- the code for supporting multithreading was modified to allow compilation and execution on MacOS as well as GNU Linux multicore computers.

Elaboration on the improvements as well as some example images are provided below. Note that variance weighting and total variation (TV)-constrained reconstruction are only available using weighted least squares minimization.

5.1 3D IMAGING

In FY 2017, the reconstruction code was modified to support 3D imaging, including the ability to incorporate multiple tomographic scans taken at different heights. In principle, this is a simple matter of extending x and y and accordingly adding more rows and columns to A. That is, x changes from being an $N \times 1$ column vector where $N = N_x N_y$, corresponding to the number of voxels in the 2D image, to a column vector where $N = N_x N_y N_z$, corresponding to the number of voxels in the 3D image. Likewise, y changes from being an $M \times 1$ column vector where $M = N_\theta N_Y$, corresponding to the combined number of one-dimensional (1D) detector pixels for all object rotations (indexed by Y and θ, respectively), to one where $M = N_S N_\theta N_Y N_Z$, corresponding to the combined number of 2D detector pixels (indexed by Y and Z) for all scans (indexed by S) and object rotations (indexed by θ). In both cases, A is the resulting $M \times N$ matrix that connects the two spaces.

In the details of implementation, switching from 2D to 3D required large portions of the code be rewritten to properly account for the change in the underlying system geometry. For example, matrix A is computed on-the-fly on a row-by-row basis by interpolating along the modeled ray. The change from 2D to 3D imaging meant a change from 2D to 3D interpolation. More generally, computations and support data structures had to be updated throughout the code to account for the change in the underlying system geometry.

Tomographic measurements were performed on a variety of target objects at multiple heights, and the new code was used to reconstruct 3D images. Example projection data and reconstructed images are shown for a target consisting of a depleted uranium (DU) annular casting and a hollow lead cube supporting a high-density polyethylene (HDPE) block. A photograph of the target is shown in Figure 5(a). In this image, the HDPE block is askew. The measurement consisted of 36 projections at three heights separated by 7.5 cm steps that spanned the object. Figures 5(b)–(d) show the 0° transmission projections for each height, and (e) shows a composite projection obtained by stitching the three height measurements together. The stitching was accomplished by averaging pixels that overlap.
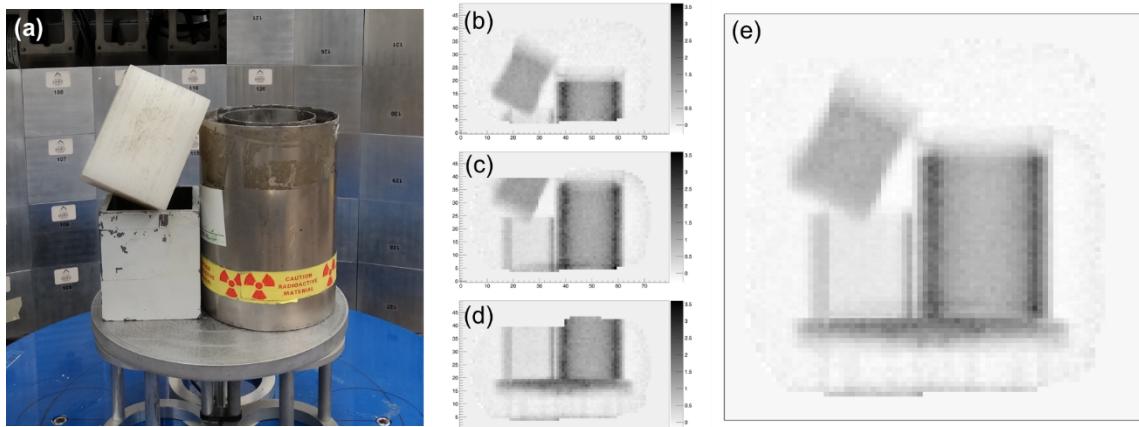
Figure 5. (a) Photograph of a target consisting of a DU casting, hollow lead cube, and a HDPE block. (b, c, d) 0° projection radiographs at the (b) highest, (c) middle, and (d) lowest heights.

(e) Composite 0° radiograph made by averaging the individual overlapping radiographs from the different heights.

Likewise, composite projections can be created for the induced reaction images. These projections are shown in Figure 6 for (b) induced neutron doubles, (c) induced neutron singles, and (d) hydrogen elastic scattering. The corresponding images with the induced-reaction image overlaid on the transmission image are shown in Figure 6(f)–(h). In these image overlays, the hydrogenous and fissionable materials are readily identifiable. Note that the data shown in Figures 5 and 6 is projection data. The projection data (as individual projections at each height, not the composite projections) is the input to the 3D iterative reconstruction code.
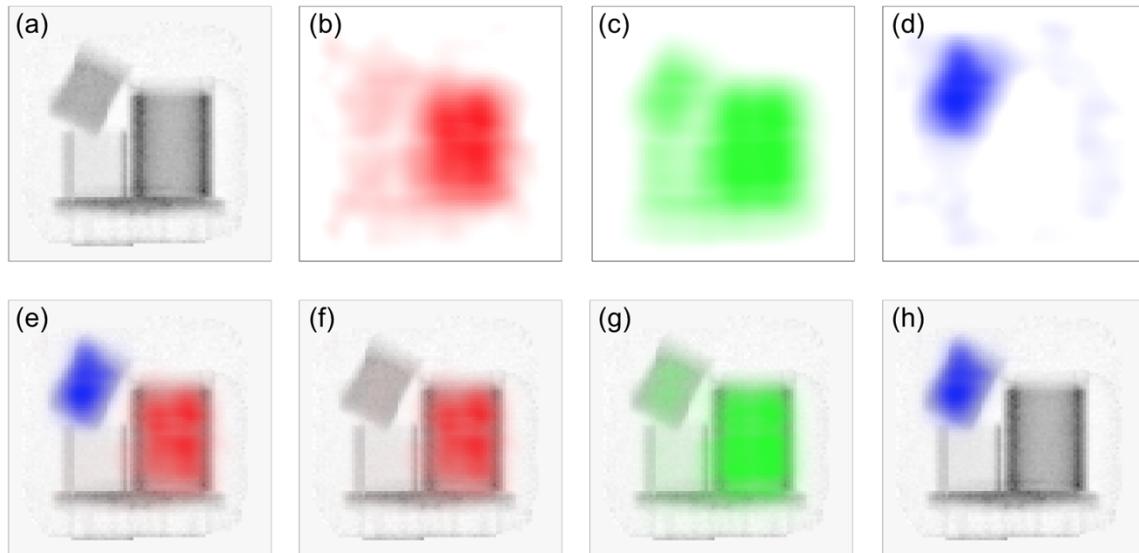
Figure 6. Composite 0° projection radiographs produced by combining radiographs from all three heights. (a) Transmission radiograph, (b) induced doubles radiograph, (c) induced singles radiograph, (d) hydrogen scatter radiograph, (e) hydrogen scatter (blue) and induced doubles (red) overlaid on transmission radiograph, (f) induced doubles (red) overlaid on transmission radiograph, (g) induced singles (green) overlaid on transmission radiograph, and (h) hydrogen scatter (blue) overlaid on transmission radiograph.

The output of the reconstruction code is a stack of 2D slices that spans the volume that can be reconstructed; these results can be viewed in a variety of ways. Figure 7 shows slices through the 3D reconstruction at two heights, analogous to what was typically shown from a 2D reconstruction.
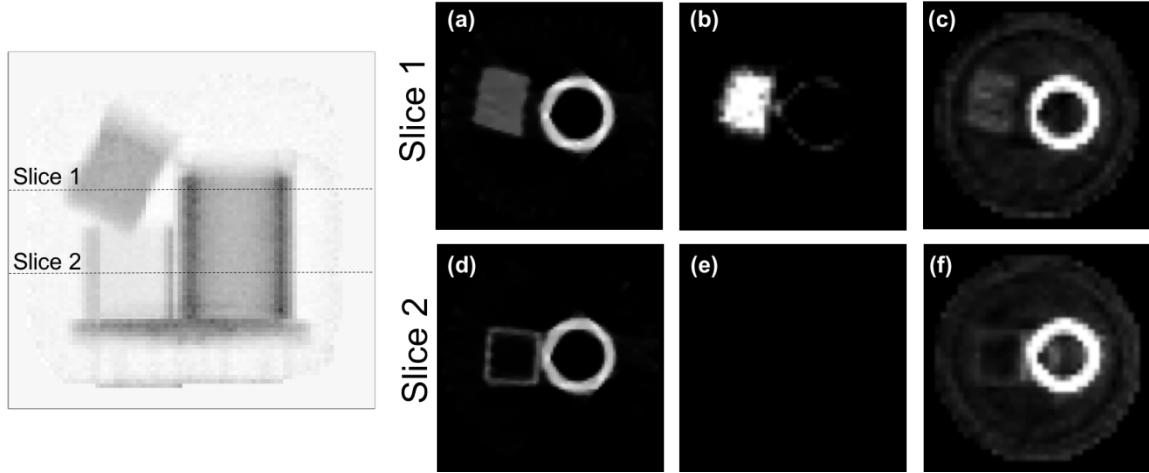
Figure 7. (*Left*) Composite 0° projection transmission radiograph with approximate slice locations labeled. Slices through the 3D reconstructions for (a) transmission, (b) hydrogen scatter, and (c) induced doubles for slice 1. Slices through the 3D reconstructions for (d) transmission, (e) hydrogen scatter, and (f) induced doubles for slice 2.

Another way to view the data is in 3D volumetric form, as shown in Figure 8. Here, the transmission reconstruction nicely reproduces the target object, and the hydrogen scatter and induced-doubles reconstructions highlight the volumes associated with the respective types of interactions.
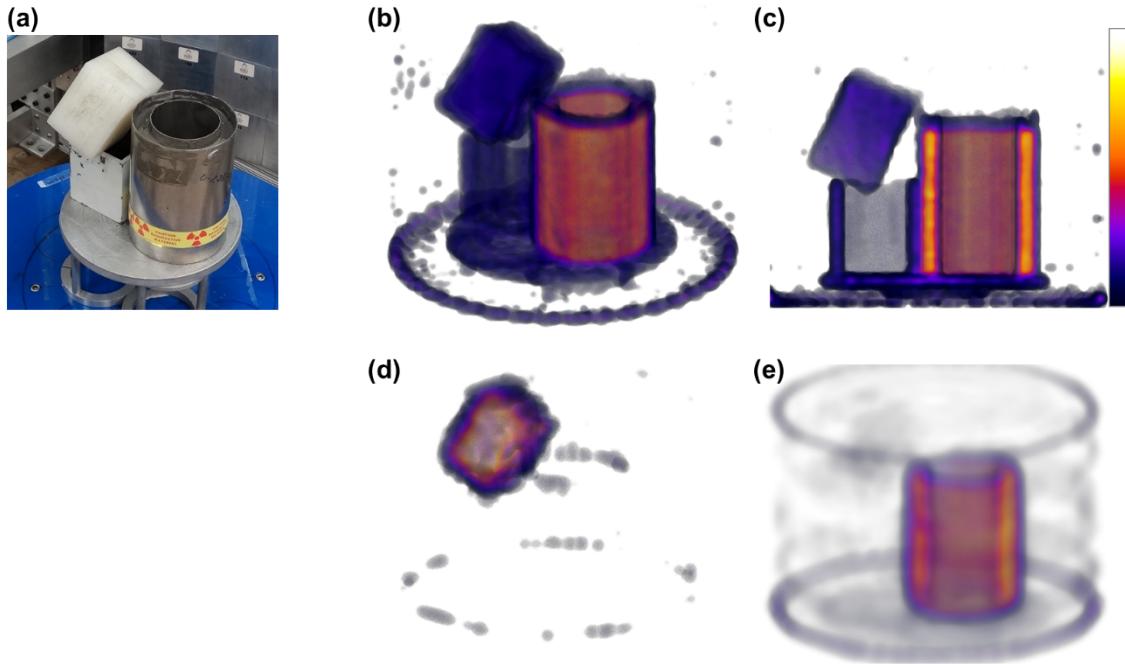
Figure 8. (a) Photograph of the target object for comparison. (b) 3D rendering of the transmission reconstruction with color scaling using the value of the reconstructed transmission attenuation coefficient. (c) Cutaway of the transmission reconstruction revealing the insides of the DU casting, lead cube, and HDPE block. (d) 3D rendering of the hydrogen scatter reconstruction with color scaling using the reconstructed value of the attenuation coefficient for hydrogen scatter. (e) 3D rendering of the induced doubles reconstruction with color scaling using the reconstructed value.

In addition, the time to perform the reconstructions was evaluated as a function of the number of compute cores used in the calculation, and the results of this evaluation are shown in Figure 9. Note that for the first few cores, additional cores reduce the time needed to both initialize and run the reconstruction in proportion to the number of cores. However, the amount of speedup saturates at approximately 16 cores. This saturation indicates that there is sequential overhead in the code, most likely in the portion of the code performing the TV computation, which requires sorting.
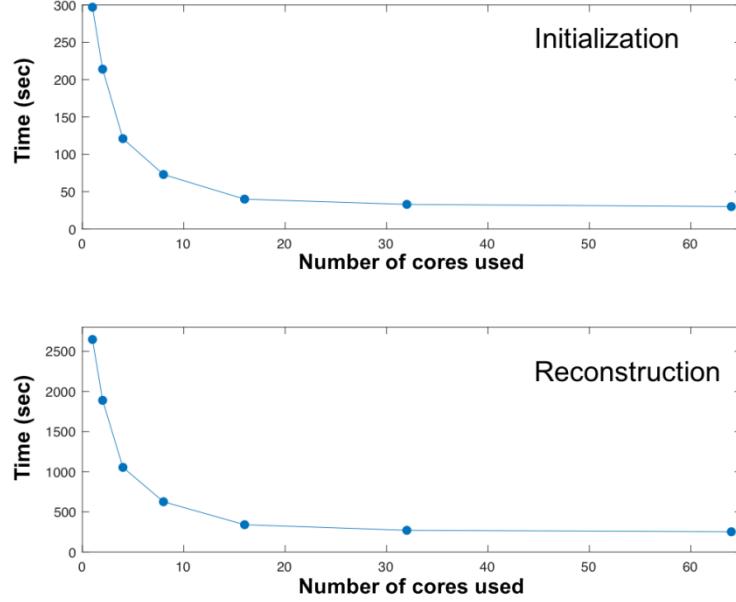
Figure 9. Calculation time versus number of cores used for transmission (*top*) initialization and (*bottom*) reconstruction.

5.2   VARIANCE WEIGHTING

Matrix $W = [w_i]$ from Equation (1) denotes a diagonal weight matrix. The code was modified to allow W to take the following forms: no weighting ($W = I$), data-driven weighting, and variance-based weighting. Data-driven weighting is appropriate for transmission data and refers to allowing forward projected values to deviate from heavily attenuated measurements at a lower cost than the less attenuated measurements. That is, the weights are estimated to be $w_i = \exp\left(\frac{-y_i}{\max(y_i)}\right)$. Variance-based weighting refers to reading w_i values from a user-provided file that represent the mean squared uncertainty associated with each data measurement. This mode also supports binary weight values that indicate which detector pixels to include in the model of the system geometry. For example, a detector block that has been taken off-line can be flagged as such. Similarly, the irregularly shaped coincidence cone for the alpha particles can be indicated. The code was written to handle the combination of these two modes as well.

5.3   INDUCED-REACTION PSF BLURRING

The initial directions of neutrons that are associated with induced reactions have an associated uncertainty. The code was modified to take this uncertainty into account by incorporating a spatially invariant smoothing kernel based on a heavy-tailed Gaussian PSF. Mathematically, the forward projection system model was changed from A to HA, where the matrix H denotes a block-circulant matrix

implementing the smoothing kernel. For computational efficiency, the application of $\boldsymbol{H}$ and adjoint operator $\boldsymbol{H^T}$ was implemented using 2D fast Fourier transforms.

5.4   SCATTER IMAGING AND PHI-TIME SMOOTHING

Scatter imaging uses scatter angle-time, or phi-time, cut masks to identify elastic scatters from hydrogen. The reconstruction code needs to calculate the efficiency of these cut masks. The original Matlab script for 2D imaging performed this computation inefficiently by repeatedly applying a smoothing kernel to the phi-time efficiency maps computed for the many pixel-projection pairs. Although the corresponding C code was fast enough for 2D imaging, it was prohibitively slow for 3D imaging. We determined that it is mathematically equivalent to smooth the binary phi-time cut masks, which only needs to be done once. Consequently, the time needed for phi-time smoothing is now negligible.

5.5   TOTAL VARIATION CONSTRAINT

To reconstruct smoother images and preserve sharp edges consistent with machined parts, the mathematical imaging model was extended to consist of a weighted least squares (WLS) minimization problem that is subjected to a TV constraint.

$$\boldsymbol{x}^* = \arg\min_{0 \le x} \frac{1}{2}\|\boldsymbol{Ax} - \boldsymbol{y}\|_W^2 \text{s.t. TV}(\boldsymbol{x}) \le \epsilon$$

As before, the matrix $\boldsymbol{A} = \left[a_{ij}\right]$ incorporates knowledge about the system geometry and physics associated with the induced reactions, and the vectors $\boldsymbol{x} = [x_j]$ and $\boldsymbol{y} = [y_i]$ represent the image being reconstructed and the acquired projection data, respectively. Each equation in the linear system models describes what happens to neutrons as they travel along a ray from the deuterium–tritium source to the detector array for a given rotation of the object being imaged. TV refers to the L1 norm of the L2 norms of the image gradients, which is

$$TV(x) = \sum_i \|\nabla_i x\| \,.$$

The effect of constraining the TV value of the image is sparsification of the gradient magnitude image. This sparsification leads to solutions that exhibit smooth regions with well-defined edges.

The TV constraint was implemented using a relaxed, incremental proximal gradient scheme that consists of a two-step iteration. First, the WLS term is minimized using a proximal mapping that keeps the image somewhat close to the one produced in the previous iteration. Second, the TV constraint is satisfied by mapping the image to the closest point on the surface of an L1-ball. This mapping is carried out using the Chambolle–Pock algorithm for solving convex optimization problems [8].

Using a less mathematical description, the WLS computation initially converges to an image that has relatively sharp edges, even if it appears grainy or otherwise noisy. Then, the closest image that satisfies the TV constraint is selected. The process is then repeated. In each iteration, the WLS computation is prevented from deviating too far from the image produced by the TV constraint and vice versa. After a small number of these two-step iterations, a solution emerges with a relatively small WLS residual error, and the TV value is close to or smaller than the given constraint.

Exemplar reconstructions with and without the TV constraint are shown in Figure 10. Note, with the TV constraint areas common to a single material are significantly smoother and better reflect a homogeneous attenuation coefficient within the material.
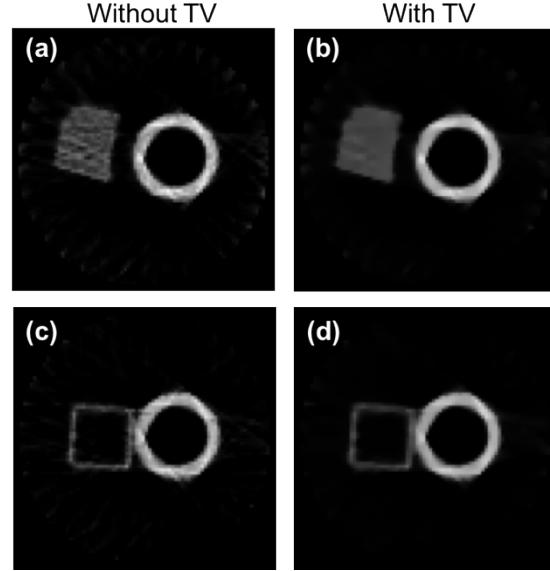
Figure 10. Slices from 3D transmission reconstruction (*left*) without and (*right*) with TV constraint. Slices at a height with the HDPE block and DU casting are shown in (a) and (b), and slices at a height with the hollow lead cube and DU casting are shown in (c) and (d).

6.   YEAR 3 PROGRESS

By the start of FY 2018, the reconstruction software was fully operational for performing 3D reconstructions of transmission and induced-reaction projection data. Therefore, most of the work during this year focused on improving individual components of the reconstruction to correct errors, improving performance, or adding more realism to the response model used. Year 3 efforts included

- improving geometrical source and detector models,

- implementing directionally variant PSF in the forward and back projection operators for induced-reaction imaging,

- adding the option to blur image voxel values for cases when small voxel sizes are chosen,

- incorporating user-defined variances to weight the projection data in the reconstruction,

- adding the ability to correct for misalignment of the target rotation axis from the center of the imager,

- adding the option to automatically calculate the maximum TV value to be used in the TV-constrained reconstructions,

- adding the option to specify the size of the reconstruction volume of interest,

- improving parallelization implementation,

- validating the reconstruction software, and

- developing a 3D visualization of the results and the ability to perform 3D overlays of induced-reaction reconstructions on transmission.

The subsections below describe the changes in more detail.

6.1   SOURCE AND DETECTOR MODELING

Improvements were made to the geometrical models of the source and detectors to make them more realistic. For the neutron detectors, the pixel array was originally modeled as an arc of pixels with gaps between groups of 10 pixels to represent gaps between detectors. The actual detectors present a square face to the source with gaps between the plastic scintillator pixel blocks. Figure 11 shows a drawing of the block detectors as they are configured for the measurements results shown in this report. The piecewise linear configuration and gaps between blocks can be seen in this image.
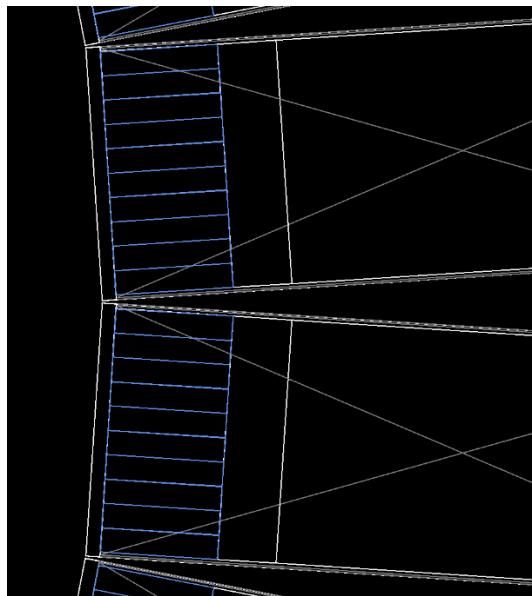
Figure 11. Detector geometry with gaps between detectors shown. This figure shows the location of the components of the neutron block detector. Of particular interest is the placement of the plastic scintillator pixel blocks shown in blue. The piecewise linear nature of their construction and the gaps between pixels is visible.

The reconstruction software now allows the detectors to be modeled as curved, as before, or as piecewise linear, which is a more accurate reflection of how the imager is constructed. The detector blocks can now be placed on an arc including user-defined gaps between the pixel block and outer wall of the detectors and the gaps between the outer walls between adjacent blocks. The code was refactored to allow for not only this and the curved detector option, but also for any future detector model. This was achieved by separating the definition of the physical characteristics, such as detector pixel coordinates and the directions normal to their face, from the code that uses them.

The source model was also updated to include a finite diameter aperture (circular opening) to more accurately approximate the spot geometry where the deuterium–tritium reaction takes place within the neutron generator. This allows the forward model to capture the line of response from the target spot to a given neutron pixel, which is actually made up of multiple different, but closely related, lines of response through the target. The number of rays cast from the source to the detector pixels can be set by the user to be 1, 5, or 9 rays. In all cases, the center ray is modeled by a line connecting the center of the source with the center of a detector pixel. When more than one ray is used, the remaining rays are uniformly distributed in circle at 0.25 times the aperture diameter. These rays are randomly connected with uniformly distributed raster scan points at the detector pixel.

6.2   PSF IMPLEMENTATION

The modeling of the PSF of the imager response was updated to be either defined by a function or in a user-supplied file. The former uses a spatially invariant Gaussian convolution kernel (see Section 5.3). The latter reads user-defined kernel parameters from a file, which allows the smoothing to be spatially variant. This PSF can be used by any of the imaging modalities. Figure 12 shows the variation in alpha cone widths of the PSF across the face of the neuron detector array. A fit to the measured widths is also shown. The fit values are used as input to the reconstruction code when file input is specified.
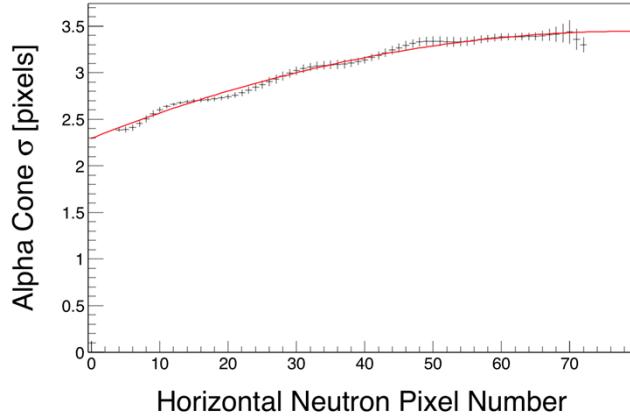
Figure 12. Alpha cone horizontal width as a function of horizontal neutron pixel number. The initial neutron cone directed toward a given neutron pixel has a horizontal width that varies across the face of the detector array. The Gaussian sigma of the cone is plotted in black in units of detector pixels, and a fit through the data is shown in red.

6.3   IMAGE BLURRING

Just as the detector data can be blurred by a PSF kernel following forward projection (**HA**), the image data can also be blurred before forward projection (**AH**). This is desirable when the requested number of image voxels is large, giving voxel volumes that are small, which may mean that the number of rays to detector pixels through some voxels is small (if not zero). For cases of few rays through a voxel, this blurring can be used in combination with, or as an alternative to, upsampling the projection data. The image blurring kernel is a Gaussian distribution with a user-defined full-width at half-maximum.

6.4   VARIANCE WEIGHTED RECONSTRUCTION

The weighted least squares algorithm for reconstructing the image was rewritten at the end of FY 2017 to allow user-defined variance weighted measurements (see Section 5.2). The input can either be a single projection-sized array that specifies identical variances for each projection or one array that specifies unique variances for each projection. The alternative options for handling the variances are outlined in Section 5.2.

So far, uncertainty propagation has only been implemented for transmission imaging in the analysis. The code has been tested with uncertainties passed to the reconstruction. The induced-reaction analysis does not yet have uncertainty propagation, so testing will occur during the next FY.

6.5   AXIS OF ROTATION CORRECTION

The turntable target objects are placed and rotated on is not rigidly fixed to the rest of the imager gantry, leading to possible alignment errors when an imaging measurement is set up. Previously, the code assumed that the axis of rotation of the turntable was aligned with the center of the detector array. The code has been updated to allow a correction to the lateral offset of the detector assembly relative to the axis of rotation to be specified, using either a user-defined offset value or a value calculated from the projection data themselves.

Two automatic calculations of this offset were attempted. The first used two projections spaced 180° apart, with the second projection flipped horizontally to align with the first projection. A mutual information algorithm [9] was then applied to the projections, but this method failed for reconstructions that requested upsampling of the detector array. A second method was implemented based on cross correlation. In this method, all the projection data were summed (P), and summation was then compared with its mirror image (P'). Since the correction was only needed for the horizontal dimension, all the data in the vertical dimension were collapsed onto the horizontal axis. Then, the pixel offset, k_{max}, was found that gave the maximum cross correlation

$$\sum_i P_{i+k} P'_i.$$

The offset is then given by $\frac{1}{2} k_{max} \, dY$, where dY is the size of a detector pixel in the horizontal direction. To get the offset correction in absolute units, the magnification of the imager must be included.

6.6   AUTOMATIC TV CONSTRAINT HANDLING

The code was modified to support automatic control of the TV constraint parameter. (See Section 5.5 for a discussion of the TV constraint.) Previously, a determination of the target value for the TV constraint required performing a reconstruction, calculating the final TV in the reconstruction, and then running the reconstruction again with a maximum TV value based on the initial reconstruction. Consequently, the reconstruction could not be run unattended. This process is now performed automatically in the code if desired. With this option, an unconstrained reconstruction is first computed. The resulting image is then smoothed with a Gaussian function with user-defined full-width at half-maximum, and the TV value is calculated for the smoothed copy of the image and used as the target for the subsequent TV constrained reconstruction. Alternatively, the ability to specify the desired TV cap explicitly is also still available. In addition to automatic TV constraint handling, the code was updated to allow simultaneous definition of a quadratic regularizer and the TV constraint.

6.7 RECONSTRUCTION REGION HANDLING

An imager with a 2D fan beam produces data that can be reconstructed with pixels contained within a circular disk. For 3D cone beam data, the valid reconstruction region consists of voxels contained within a cylinder with pointed ends. The code was accordingly modified to restrict which voxels are considered. Since voxels at the edge and, in particular, the ends of the reconstruction region are only partially intersected by the detector rays, care was taken to eliminate those voxels from the computation. This helps to reduce problems that can be encountered when small values are being divided by even smaller values.

The refinement of the reconstruction region definition just described allowed for the implementation of an additional option to define a radius for the reconstruction region cylinder. This option is used in cases where the extent of the target object is known, and the reconstruction region can be limited without loss of information. A limited reconstruction region results in faster completion times for reconstructions.

In addition, if any lateral offset exists for the axis of rotation, the radius of the reconstruction cylinder must be reduced, an operation the code performs automatically.

6.8 PARALLELIZATION

The parallelization of the code was improved during FY 2018. Sequential bottlenecks were identified in both the SIRT reconstruction code and the PSF blurring code and were fixed.

The code that handles the L1-ball projection for the TV constraint was updated to use a linear-time, randomized partitioning algorithm [10]. The original implementation sorted the gradient data first, and then multiple binary heaps were used to detect and eliminate the low TV contributors. The new implementation merely partitions the data until the desired cutoff has been found that eliminates all low TV contributors.

6.9 DETECTOR EFFICIENCY VARIATION

The ability to specify varying detector efficiencies for each block detector in the imaging array was added for induced-reaction reconstructions. For transmission imaging, differences in efficiency from detector to detector get canceled out because all count data is normalized by void counts when calculating the attenuation for a given detector pixel. For induced-reaction reconstructions, the efficiency calculations require the absolute detection efficiency as a function of neutron energy to be specified for each detector. Previously, all detectors in the array were assumed to be uniformly efficient. This new option allows an efficiency distribution to be passed to the code for each detector in the array, if desired. This option has been implemented but not yet tested.

6.10 VALIDATION EFFORTS

The reconstruction code consists of approximately 10,000 lines of highly sophisticated C programming statements. Many options have been created to handle the different imaging modalities and reconstruction configurations. Validation efforts have begun and are ongoing, and several data sets have been generated from simulations of the imaging measurements where the ground truth is perfectly known. This effort has already resulted in the identification of several bugs that have been fixed, but work is ongoing to complete the validation.

6.11 3D VISUALIZATION

With the development of a means of reconstructing 3D imaging data, a means of visualizing the 3D images was needed. The visualization tool used was ImageJ [11], an image processing package widely used in the medical community. ImageJ is open source, cross platform, and scriptable, and it contains a wide range of plugins for image operations. For this project, ImageJ was used as the embedded core of the of the Fiji [12] distribution package.

Scripts were developed and run within Fiji to visualize individual reconstructions and overlay the induced-reaction reconstructions on transmission reconstructions. For the latter, the transmission reconstruction is displayed in gray scale, and induced-fission doubles and hydrogen scatter reconstructions are displayed in red and blue, respectively. The scripts performed thresholding operations to eliminate noise and performed a merge operation to combine the transmission and induced-reaction images. Both operations are standard in Fiji. Example visualizations are shown in Figure 13 for a target consisting of a DU annulus, a neutron imaging detector, and a partially filled water bottle inside a steel tee pipe. In these visualizations, the fissionable DU annulus and the hydrogenous plastic scintillator and water are clearly highlighted within the reconstructed transmission image.
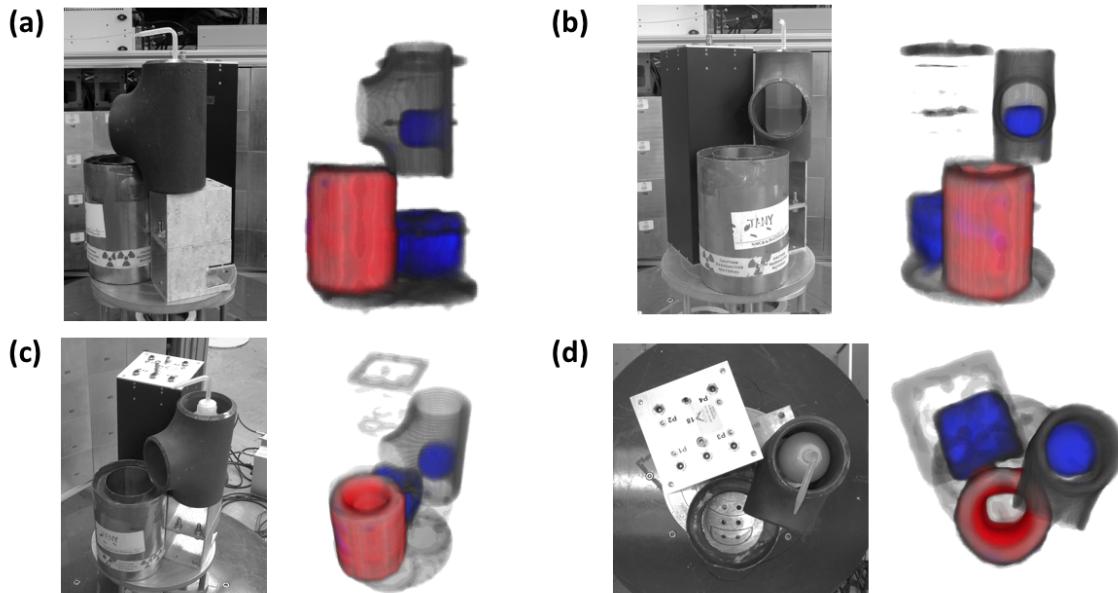
Figure 13. Photographs and images for a target consisting of a DU annulus, a neutron imaging detector, and a partially filled water bottle inside a steel tee pipe. Photograph and reconstructions are shown at several viewing angles. (a) and (b) show the target from the perspective of the deuterium–tritium generator for two different target rotations, (c) shows the target from an oblique overhead angle, and (d) shows the target from directly overhead. The reconstructions for transmission, induced-fission doubles, and hydrogen scatter are shown in gray scale, red, and blue, respectively.

7. YEAR 4 PROGRESS

In FY 2019, a number of improvements to the reconstruction software were implemented. These improvements include

- updating handling of the boundaries of the the reconstruction region,

- improving geometrical modeling of the detector,

- improving the ability to correct for misalignment of the target rotation axis from the center of the imager,

- adding ART-WLS as a reconstruction algorithm option,

- validating the transmission reconstruction code,

- delivering a conference presentation on reconstruction framework, and

- performing initial development of a joint-estimation of transmission and small-angle scattering.

The subsections below describe the work in more detail.

7.1   RECONSTRUCTION REGION HANDLING

Owing to the way that the coincidence cones for multiple projections overlap, the reconstruction region is restricted to a cylindrical volume that is tapered at the ends. Previously, this region was defined by a single scan. The code was modified to axially extend the central portion of the region to account for the overlapping regions seen for multiscan reconstructions. The code was also modified to ensure consistency between the coincidence cones and the geometrically derived reconstruction region.

When imaging large items, neutrons may intersect parts of the object that extend outside the reconstruction region, resulting in rays that indicate a higher degree of transmission, scatter, or fission activity than what can be accounted for by a system model that does not extend to those regions. Any apparent attenuation that cannot be explained by interior voxels becomes spuriously attributed to the peripheral voxels of the reconstruction region. To avoid this, an option was added to the code to automatically scale the projection data in proportion to the ratio of the modeled ray intersection length of the reconstruction region relative to the intersection length associated with a purely cylindrical region. As illustrated in Figure 14, this scaling reduces the peripheral buildup that otherwise would occur.
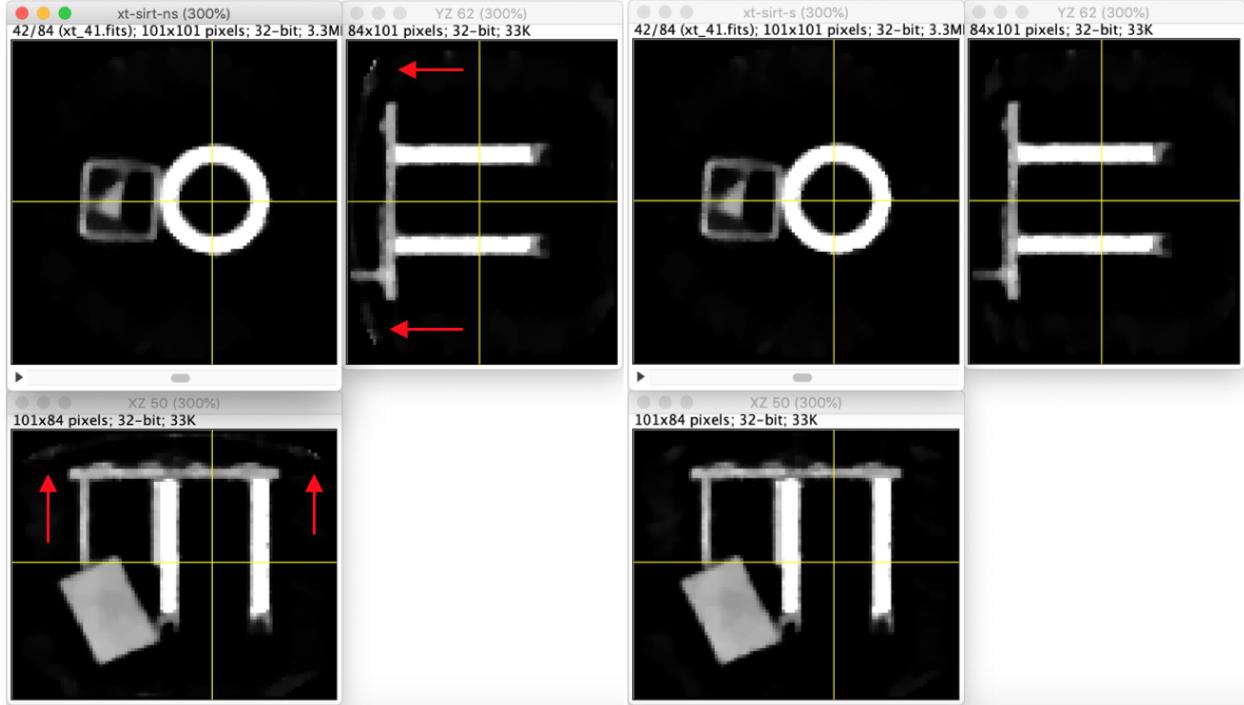
Figure 14. Effects of projection-data scaling. Projection-data scaling accounts for attenuation that takes place outside of the reconstruction region, thereby reducing buildup otherwise seen for peripheral voxels: (*left*) no scaling vs (*right*) scaling. Notice activity near bottom of scan (max gray scale reduced from 0.30 to 0.10 for illustration purposes).

7.2 DETECTOR MODELING

Improvements were made to the geometrical detector model to overcome imaging artifacts that after extensive investigation were found to be related to the dead zones created by the physical gaps that exist between adjacent detector blocks. To compensate, the user can now uniformly resample the projection data onto an idealized circular detector arc. Resampling is based on 2D interpolation of physical (angle, height) coordinate pairs. Detector-pixel unit vectors are calculated from the resulting interpolated locations.

The minimum amount of resampling interjects one additional data point in each dimension to ensure that a data point exists through the axis of rotation of the reconstruction region, thereby mapping a physical $N \times M$ detector onto an idealized $(N+1) \times (M+1)$ detector. To allow reconstruction of higher resolution images than supported by the projection data, resampling can be performed based on user-defined scaling factors, S_N and S_M. In that case, the physical $N \times M$ detector is mapped onto an idealized $(S_N \times N+1) \times (S_M \times M+1)$ detector. As Figure 15 demonstrates, resampling substantially improves image quality. Bright pixels that previously occurred near the center of several images have been eliminated. Comparison with Figure 10 also shows that the cylinder now appears truly round as opposed to the somewhat flattened look produced by the older geometrical detector model.

(a)	(b)	(c)
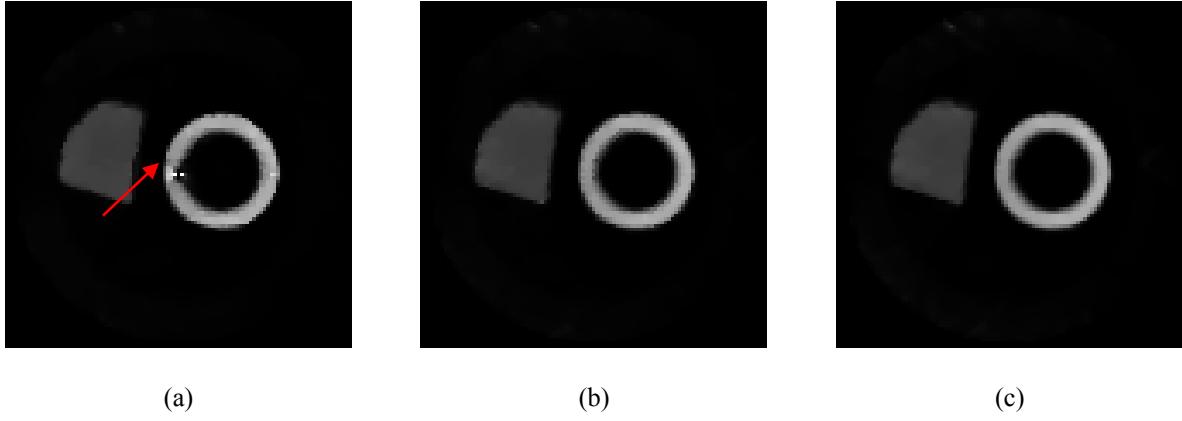
Figure 15. Slices from 3D transmission reconstruction based on raw and resampled projection data. (a) Reconstruction without resampling (SN = 0) produces artifically bright pixels (approx. 2× true value) surrounded by a dark background. (b) Reconstruction based on minimal resampling (SN = 1) eliminated the artifact. (c) Reconstruction based on more intense resampling (SN = 3) leads to a more circular appearance of the object.

In addition to the circular arc and piecewise linear configurations introduced in year 3, the user now also has the option to import detector-pixel coordinates and unit vectors directly (e.g., from models). This data can also be resampled as described above.

7.3   AXIS OF ROTATION CORRECTION

The automatic axis-of-rotation correction computation was updated to use the resampled detector model. The user can specify that a specific lateral detector offset be used for the transmission reconstruction or request that the code calculates it using either the default resampling factor of ×10 or a user-defined factor. To ensure alignment, elasic scatter and induced-fission reconstructions were performed using the same lateral offset used in the transmission reconstruction.

7.4   ART-WLS RECONSTRUCTION ALGORITHM

In preparation for a publication that describes our algorithmic work based on SIRT, the ART (algebraic reconstruction technique) algorithm was implemented for comparison and added as an option. ART is described in [13]. The ART update for the WLS problem can be expressed as

$$\forall i: x^{(k_{i+1})} = x^{(k_i)} - \frac{A_i^T x^{(k_i)} - y_i}{|A_i|^2 + (w_i \beta_k)} A_i,$$

where the iteration is over individual rays (index i) and A_i denotes the corresponding row in the system matrix. Although this update equation represents a closed-form solution to the first step of the relaxed, incremental proximal gradient scheme for the WLS-imaging mentioned in Section 5.5, it does not support the regularization mentioned in Section 4 and cannot accommodate the PSF blurring described in Section 5.3 or the image blurring described in Section 6.3. Since its update is ray-based, ART is also more susceptible to noise and other outlier data than SIRT, which uses an average update computed for all rays across many projections. In terms of implementation, ART requires an extra step to compute the L2 norm for row-sum A_i. Specifically, the a_{ij} interpolation coefficients need to be sorted and summed on a per voxel basis before reconstruction. ART is parallelized in a manner similar to SIRT.

Figure 16 provides a visual comparison of SIRT and ART. None of the mentioned shortcomings manifest themselves; the reconstructions appear similar in quality. Figure 17 provides a plot of the WLS values produced by the two algorithms before applying the TV constraint. SIRT reduces the WLS value, albeit non-monotonically, whereas ART fails to produce reductions in its large WLS value after a few iterations. This indicates that ART repeatedly produces high-valued outlier voxel values. Figure 18 illustrates the progression of the corresponding TV values before and after application of the TV constraint. As expected, ART and SIRT converge to the same TV constraint value, but while SIRT is seen to converge to a predictable narrow range of values, ART continually drives the TV value to a large value in the next iteration. This again indicates that ART repeatedly produces a noisier image. The circles on the SIRT plots mark a reduction in the number of ordered subsets. Initially the above mentioned average update is computed on a per projection basis. After a number of iterations, the update is computed using pairs of projections, then groups of four projections, until finally all projections are considered together. The effect of this reduction in the number of ordered subsets is to slow convergence down as the algorithm gets closer and closer to the solution.
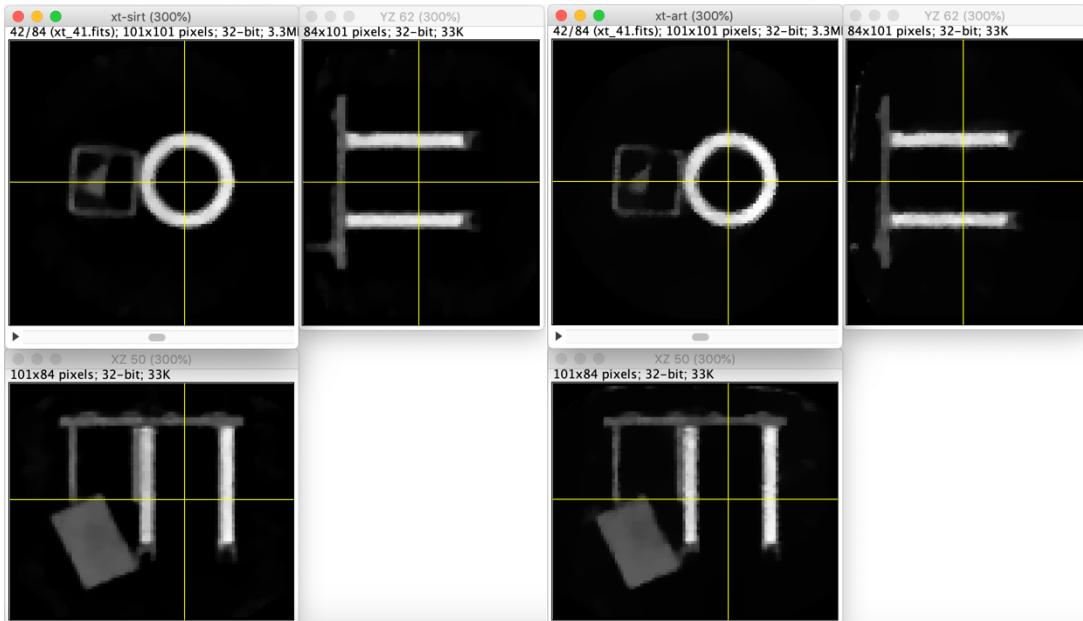
Figure 16. SIRT vs ART comparison for transmission reconstruction. SIRT (left) and ART (right) produce visually comparable images. Close-up inspection of the 3D XY, XZ, and YZ cross-sections reveals subtle, but noticable differences.
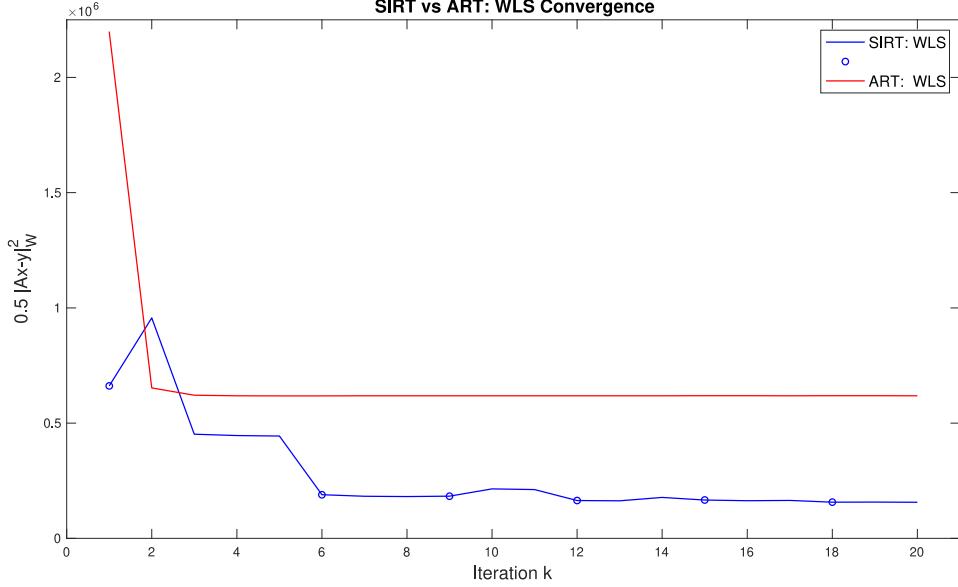
Figure 17. SIRT vs ART comparison for transmission reconstruction. SIRT (blue) reduces the WLS value, albeit non-monotonically, whereas ART (red) stalls out at a large value after a few iterations.
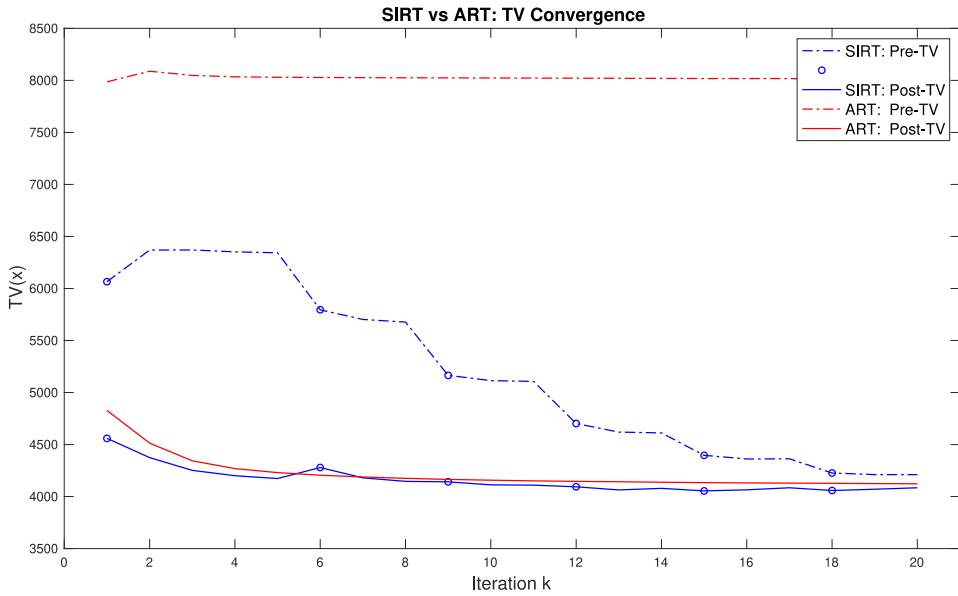
Figure 18. SIRT vs ART comparison for transmission reconstruction. SIRT (blue) converges to the TV constraint value and remains close to it while ART (red) continually pulls away in the next iteration.

7.5   VALIDATION EFFORTS

Validation was completed for the transmission code. For the validation effors, simulated data sets were generated using a perfectly known ground truth. This effort resulted in the identification of several minor bugs that were fixed, after which transmission reconstructions were found to be in agreement with the ground-truth data. Validation remains to be performed on the induced reaction reconstructions, an effort expected to be completed at a future time.

7.6   CONFERENCE PRESENTATION

The conference "Modern Challenges in Imaging" was held from August 5 to 9, 2019, at Tufts University (Boston, MA) in celebration of the 40th anniversary of Alan Cormack's 1979 Nobel Prize in physiology and medicine. J. Gregor delivered a presentation entitled "Unified Reconstruction Framework for Multi-Modality Neutron Imaging" at an associated minisymposium on security imaging. The focus of the presentation was on the ability to reconstruct transmission, elastic scatter, and induced-fission images using the same basic WLS solver by incorporating the different physics in the system model as described earlier in this report. The authors plan to write a journal article based on this presentation.

7.7   JOINT ESTIMATION OF TRANSMISSION AND SMALL-ANGLE SCATTER

The primary systematic error in generating images with quantitative voxel values originates from the inability to accurately determine the amount of small-angle scattering that is mistakenly attributed to transmission. A significant number of neutrons scatter in the target at very small angles and with very little energy loss, resulting in their detection at times and locations that are impossible to distinguish from transmission events. Present data analysis performs an approximate correction, but the simple model used to estimate the small-angle scattering background is an inadequate approximation of the actual contribution. During the present year, a reconstruction approach that solves for transmission and small-angle scattering simultaneously was initiated. A description of the current status of the effort follows.

To support development of a forward model of transmission and elastic scattering, simulations were performed that allowed the scattering and transmission components to be known a priori. This provided a direct means of evaluating the performance of the forward model. For these simuations, a single row of detector pixels was simulated by an arced surface that recorded the crossing of neutrons (see Figure 19). In this way, the effects of scattering in the target object were isolated, and the effects of scattering within and between detectors were ignored. The scattering response of the detectors will be included at a later date, perhaps by use of a point-spread function applied to forward-projected data.

To allow neutrons to scatter into the 1D row of detectors, the deuterium–tritium neutrons were thrown into the standard 2D cones associated with a single row of alpha pixels (see Figure 19, right). Most of the scattered neutrons observed by the detectors originate from neutrons initially directed outside of the detector plane, but scattering in the target redirects them toward the detectors. Simulated transmission and scattering data were generated for a hollow triangular prism made of pure iron, with outer side lengths of 15 cm, wall thicknesses of approximately 2.2 cm, and a height of 15 cm.
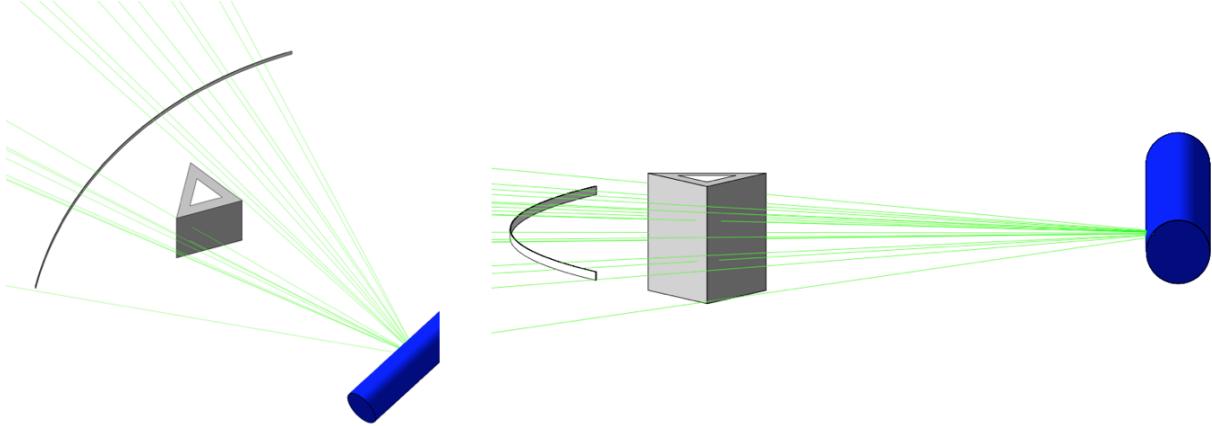
Figure 19. Detector and target geometry modeled for joint estimation development. (*Left*) an overhead view of the iron triangular prism and single detector row. (*Right*) Side view of the target and detector, illustrating the modeled neutron cone's vertical extent.

Before describing the details of the reconstruction, the nomenclature for the basic variables used in the following discussion will be defined. There are two different classes of variables to be considered: (1) variables that reference neutron detector pixels and (2) variables that reference alpha pixels. For a given projection, $\vec{r}_j$ denotes the vector from the source to the center of the jth neutron-detector pixel, d_j, as depicted on the left side of Figure 20. Likewise, $\vec{\rho}_i$ denotes the vector through the center of the coincidence cone for a given alpha-detector pixel, α_i, as is depicted on the right side of Figure 20. Note that the alpha direction vectors will not, in general, coincide with neutron detector pixels but will typically point between pixels. The precise direction of the alpha pixels is determined from fits to the coincidence distributions observed in the "target-out" void data. The forward projections described later are performed along the $\vec{r}_j$ vectors and use the content of a voxelized volume to determine the observed quantities (Figure 20).
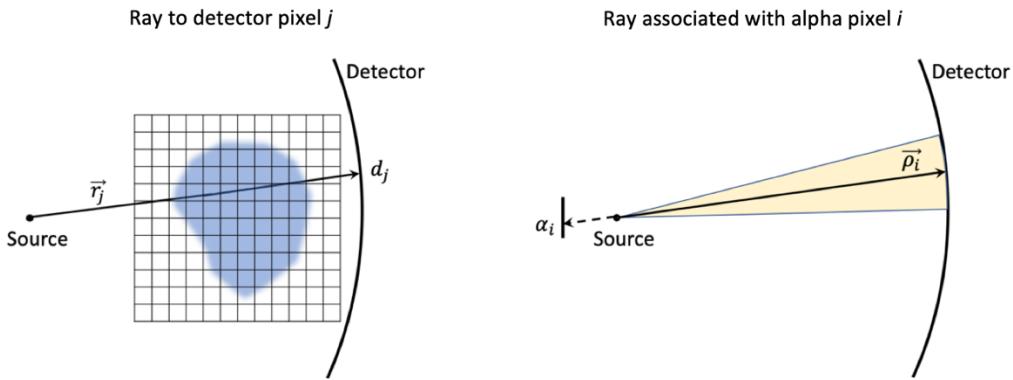
Figure 20. Vector definitions for the joint estimation problem. (*Left*) Vector from source to detector pixel center. (*Right*) Vector of the center of the alpha pixel cone from the source to where it intersects the detector array.

The actual initial direction of an individual deuterium–tritium neutron is known only to the size of its coincidence cone for the alpha pixel with which it is coincident. Therefore, to calculate the angular deviation of a particlar neutron detector event from its initial direction, the central cone direction $\vec{\rho}_i$ is

used. The angular deviation, $\Delta\phi_{i,j}$, for a neutron detected in d_j and coincident with alpha pixel α_i is defined according to the relation

$$\cos\Delta\phi_{i,j} = \hat{\rho}_i \cdot \hat{r}_j,$$

and the number of counts in that pixel is denoted $N_{i,j}$.

7.7.1   Reconstruction observables

Joint estimation attempts to fit two measured observables simultaneously. In the present work, these quantities are the total observed counts in the transmission coincidence window and the total observed angular variance for the detected locations of those counts relative to their initial directions. To calculate variance, the basic projection data require an added dimension to represent the coincident alpha pixel (i.e., counts per neutron pixel coincident with a given alpha pixel, $N_{i,j}$). In addition, unlike the previous reconstruction approach, the transmitted counts are not scatter-subtracted, and thus contain both transmitted and scattered counts. From this basic projection data, the two observables that are used for comparison to the forward model are calculated.

For simplicity, the basic projection data are collapsed down to a single dimension, indexed only by neutron pixel number. For the total observed coincident counts, this is easily accomplished by adding the hit distributions for every alpha pixel together. For the total angular variance for those counts, a two-step process is required. First, the variances by alpha pixel, σ_i^2, are calculated from hit distributions on the neutron detector as

$$\sigma_i{}^2 = \frac{\sum_j N_{i,j}(\Delta\phi_{i,j})^2}{\sum_j N_{i,j}},$$

where the sums are over neutron pixels. In a second step, the alpha-pixel-based variances are converted to neutron-pixel-based variances, s^2, by taking the weighted average of all the alpha pixels that contribute to a given neutron pixel. Figure 21 shows an example in which three alpha pixel coincident cones overlap for the neutron pixel outlined by the box. Each of the alpha pixels have a calculated variance, σ_i^2, that is averaged for that neutron pixel according to the weights associated with the strength of the overlap of the given alpha pixel as measured from the void, indicated by the points in the figure. For this example, the variance assigned to the highlighted neutron pixel will be most strongly influenced by the variance observed for the alpha pixel highlighted in green and least strongly influenced by the one highlighted in red.
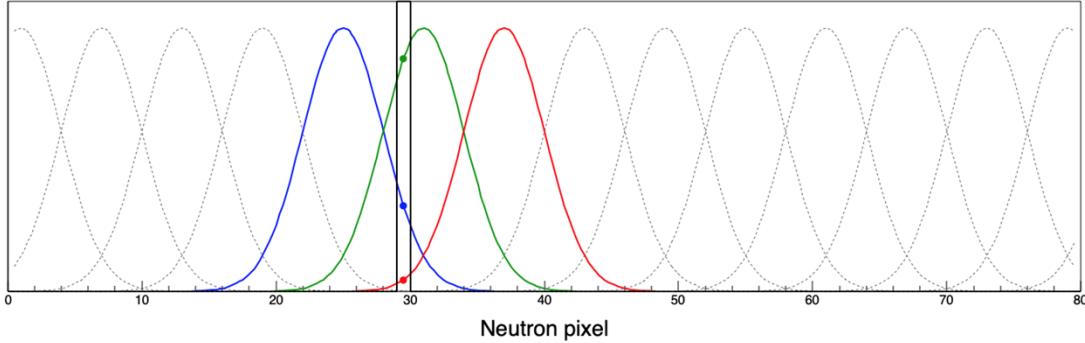
Figure 21. Illustration describing the overlap of multiple alpha pixel coincidence cones for a given neutron pixel. The distributions highlighted in colors are the alpha pixels that contribute to the neutron pixel highlighted in the box. The weights used in a weighted average of alpha pixel variances are indicated by the points.

The result of these operations on the simulated data is two sinograms, one representing the number of counts in the transmission window and the other representing the variance observed along those rays. Each is given per neutron pixel, per projection. Figure 22 shows the sinograms obtained from the simulated data of the iron triangular prism. It also shows the effect of scatter for these observables, which contributes up to about 30% of the total observed counts and variance.
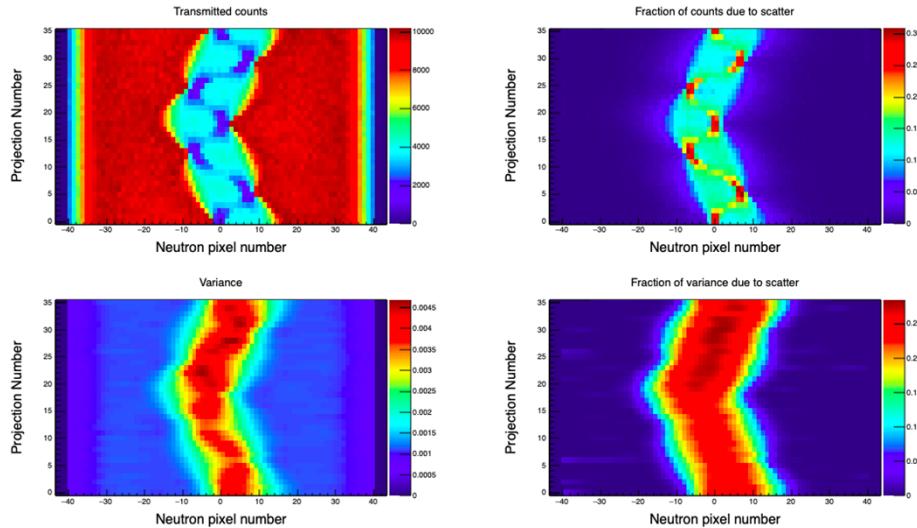
Figure 22. Sinograms used in joint estimation. (*Top left*) Sinogram of transmitted sinogram counts, (*top right*) fraction of transmitted counts due to scatter, (*bottom left*) sinogram of variance, and (*bottom right*) fraction of variance due to scatter.

7.7.2   Image-voxel quantities

For standard transmission reconstructions, the image voxels are ascribed attenuation coefficient values, μ, which represent the probability of interaction per unit length. For the joint estimation, an additional quantity that represents the angular variance due to elastic scattering is needed. This physical variance, to be denoted Σ^2, is a material-dependent quantity that describes the average angular width of the distribution of scattered neutrons based on the differential cross section for elastic scatter. It will be distinguished from the observed variance, s^2, discussed previously, which has a strong dependence on the geometry of the measurement as illustrated in Figure 23. Scattered events nearer to the detectors have less

28

distance over which to diverge from the initial direction before interacting with the detector, but the detector will subtend more solid angle. These effects are modeled as part of the forward projection.
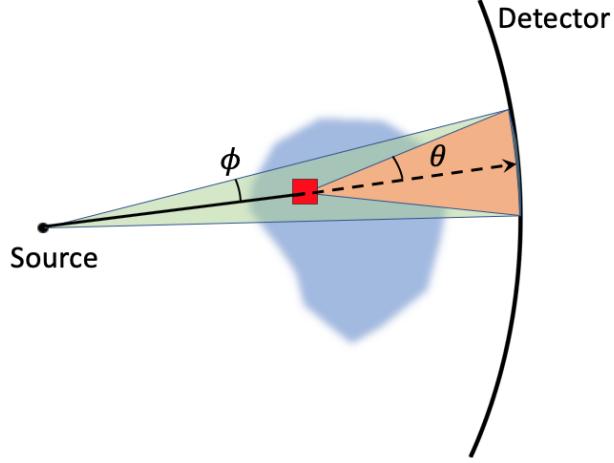
Figure 23. Illustration distinguishing the physical and observed variances for a single image voxel. The observed variance, s^2, due to scatter from the red image voxel is based on the angular deviations with angles (ϕ) measured with the source as the vertex. The physical variance, Σ^2, is a material-dependent and geometry-independent quantity that depends on the shape of the differential cross section with angles (θ) measured relative to the scatter vertex.

7.7.3   Forward projection

The forward projection uses the current guess for image voxel quantities (μ and Σ^2) and produces the corresponding guesses for the reconstruction observables (observed counts and observed variance). The forward projection is performed along the vectors $\vec{r}_j$, including the vectors directed out of the detector plane. The number of transmitted counts is calculated in the typical way, by summing the products of μ_k values with the path lengths, Δx_k, through each voxel:

$$(\mu L)_j = \sum_{k \in \vec{r}_j} \mu_k \Delta x_k$$

The total $(\mu L)_j$ is then converted to counts via the exponential function. The counts are then added with appropriate weights to obtain the estimated transmitted counts per neutron pixel per alpha pixel ($N'^{\text{trans}}_{i,j}$) using the weight values from the void as illustrated previously in Figure 21. The contribution from scattered counts ($N'^{\text{scat}}_{i,j}$) must be added before comparing against the $N_{i,j}$ values from the data.

The forward projection for calculating the number and distribution of scattered counts is more complex. The total probability of a scattered event occurring and being detected is the product of a series of individual probabilities representing different aspects of the physics of the interaction of fast neutrons in an extended object, each of which will now be described. An overall diagram with these factors labeled is shown in Figure 24.
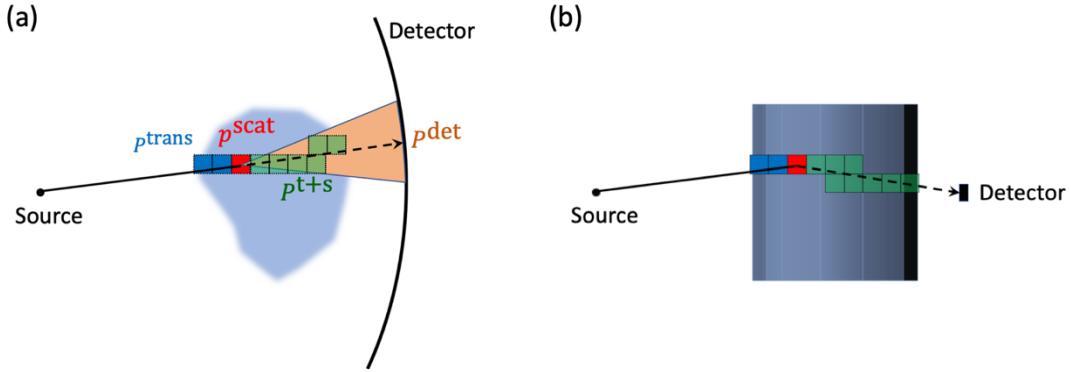
Figure 24. Illustration labeling the efficiency factors used in the calculation of elastic scatter from a single image voxel. In this example, the neutrons transmit through the blue voxels, some fraction scatter in the red voxel, and then must traverse the green voxels before reaching the detector. (a) An overhead view shows labels with the probability factors used labeled. The factors are described in the text. (b) A side view illustrates the three-dimensional nature of the problem, indicating the fact that out of plane initial neutrons can scatter into the detector.

The forward projector steps along the sequence of voxels that $\vec{r_j}$ intersects. In the discussion that follows, probabilities per voxel will be indicated with a lowercase letter p, and aggragated probabilities will use an uppercase letter P. For scatter originating from a particular voxel, i (highlighted in red in Figure 24), the first factor that must be calculated is the probability of 14 MeV neutrons transmitting to that point through the previous voxels. With individual voxel transmission probabilities denoted by p_k^{trans}, the total probability of reaching the scatter site in voxel i, P_i^{trans}, is given by

$$P_i^{\text{trans}} = \prod_{k \in \vec{r_j}; k < i} p_k^{\text{trans}} = \prod_{k \in \vec{r_j}; k < i} e^{-\mu_k \Delta x_k}.$$

The probability of elastically scattering in the voxel, i, is estimated using the calculated probability of an interaction and assuming a fixed, material-independent fraction of these are elastic scatters. The total interactions in the voxel can be calculated as $1 - e^{-\mu_i \Delta x_i}$ (i.e., the probability of *not* transmitting). The elastic scatter fraction, f_{el}, is set at 0.35 for this work, and the scatter probability in this voxel can be written as

$$p_i^{scat} = (1 - e^{-\mu_i \Delta x_i}) f_{el} = (1 - p_i^{\text{trans}}) f_{el}.$$

The neutrons that scatter in voxel i must then traverse the voxels after the one where the scattering took place, where they may be attenuated or rescattered. As a simplification, the cone of scattered neutrons is assumed to be attenuated according to the attenuation coefficients for the voxels that continue along the initial neutron direction vector, rather than considering all the neighboring voxels into which they spread. Given this assumption, the neutrons are only considered to be removed if they undergo an interaction other than elastic scattering, so the probability of surviving in an individual voxel, p_k^{t+s}, is given by the summing the probabilities for transmission and elastic scattering in these voxels.

$$p_k^{t+s} = p_k^{\text{trans}} + p_k^{scat} = e^{-\mu_k \Delta x_k} + (1 - e^{-\mu_k \Delta x_k}) f_{el}$$

The individual probabilities for all the subsequent voxels must be multiplied to give a total probability of scattered neutron transmission, P_i^{t+s}.

$$P_i^{t+s} = \prod_{k \in \vec{r}_j; k > i} p_k^{t+s}$$

The final factor for determining the number of scattered neutrons detected within the transmission window is the probability that they are detected, which is a function of the shape of the differential cross section for a given physical variance in the scattering voxel, Σ_i^2, the solid angle subtended by the detector array from the scatter location, and the detector efficiency, ϵ.

$$P_i^{\text{det}} = f\left(\frac{d\sigma}{d\Omega}, d\Omega, \epsilon\right)$$

The differential cross section is modeled as a Gausssian distribution in the polar angle, θ, with a sigma given by Σ_i that eventually gets converted to a distribution in observed angle, ϕ. In principle, there is also an isotropic component to the final distribution of scatters, but this is not included at present and will be added at a later time. The solid angle is easily calculated from each image voxel to each detector pixel, and the detection efficiency is presently set to 1 to match the simulated ideal detector.

Once these probabilities are calculated, the number of detected scattered neutons along $\vec{r}_j$ that originate in voxel i can be calculated as

$$N'^{\text{scat}}_{i,r_j} = N_{0,\vec{r}_j} P_i^{\text{trans}} p_i^{scat} P_i^{t+s} P_i^{\text{det}},$$

where $N_{0,\vec{r}_j}$ is the number of neutrons initially directed along $\vec{r}_j$.

One complication in the calculation of P_i^{det} is that the observed distribution of counts at the detector will be wider than the distribution specified by the physical variance due to the interactions of the scattered neutrons in subsequent voxels. This is illustrated in Figure 25, where the intial cone in the drawing gradually grows as it interacts with subsequent voxels.
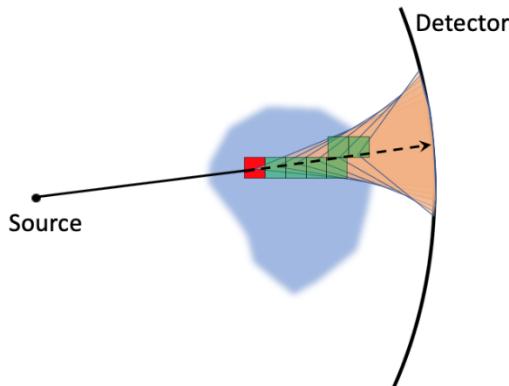

Figure 25. Drawing illustrating the widening of the cone of elastic scattered neutrons from a single image voxel when interacting with subsequent voxels.

In the forward projection, this widening is modeled to get a more accurate estimate of N'^{scat}_{i,r_j} and to know the distribution of neutrons for use in comparing the observed variance in the data. The variance of the scattered cone of neutrons after initial scatter is first converted from physical variance Σ_i^2 to the equivalent observed variance $\left(s_i^0\right)^2$ at the detector. Recall that this conversion is geometry dependent (Figure 23). The widening is performed by incrementing the observed variance for each voxel beyond the scatter site by the fraction of the scattered neutrons that rescatter in each subsequent voxel times the observed variance for that voxel, giving a final observed variance due to scatter from this voxel, s_i^2, of

$$s_i^2 = \left(s_i^0\right)^2 + \sum_{k \,\in\, \vec{r}_j \,;\, k > i} \frac{p_k^{scat}}{p_k^{\text{t}+\text{s}}} s_k^2 .$$

As with the calculation of the number of scatters that transmit through the remaining material, the physical variances from subsequent voxels are simplified by estimating the voxels that lie along the continuation of the initial neutron direction vector.

Once the number and observed angular variance of the scattered neutrons is calculated, precomputed detector hit distributions, $h(s_k^2)$, are used to increment the number of scattered counts observed in every detector pixel due to every voxel intersected by $\vec{r}_j$:

$$\sum_{k \,\in\, \vec{r}_j} N'^{\text{scat}}_{k,r_j} h(s_k^2).$$

The aggregate scattered neutron hit distribution is then distributed to the alpha pixels that contribute to the intended neutron pixel using the weight values from the void as illustrated previously in Figure 21, resulting in an estimated number of scattered counts per neutron pixel per alpha pixel, $N'^{\text{scat}}_{i,j}$. The total estimated hit distributions are given from the sum of the transmitted and scattered counts:

$$N'_{i,j} = N'^{\text{scat}}_{i,j} + N'^{\text{trans}}_{i,j},$$

at which point the hit distributions are in the same form as the basic measured projection data, $N_{i,j}$. The operations described above to generate the measured observable sinograms are used on these count distributions for use in the iterative solver.

Note, the efficiency factors calculated during the forward projection to determine the observed variances depend on the variables being reconstructed, μ and Σ^2, and because these values will change with every iterative update, the efficiencies will also change. With the complicated nature of the forward projection, an equally complicated back projection would be needed to extract update factors for the physical variance values. To avoid this and simplify the back projection, the calculations performed during the forward projection are used to determine the effective efficiency factors that linearize the back projection. In other words, the variance calculation expressed as

$$s_j^2 = \left(s_j^2\right)^{\text{trans}} + \sum_{k \,\in\, \vec{r}_j} \varepsilon_{k,j} \Sigma_k^2,$$

where $\varepsilon_{k,j}$ captures the response and physical factors described above. To accommodate this simpler back projection approach, the $\varepsilon_{k,j}$ values are calculated and stored as the forward projection is being performed, and these values are updated every time the forward projection is recalculated.

Because the data being reconstructed are from radiation transport simulations, the idealized values for the image voxels are known, as seen in Figure 26. These images were used to generate forward-projected data and compared to the full Geant4 simulation to evaluate the model being used in the forward projection. Figure 27 shows the forward-projected results in red for a single alpha pixel for transmission, elastic scatter, and the total (transmission plus scatter), with Geant4 results shown in black. The shape of the count distributions agree well with the Geant4 simulation, but the magnitude is slightly off. For this alpha pixel, the forward-projected transmission counts are approximately 25% less than the Geant4 calculation, but the number of scatter counts agrees to about 3%. The origin of the discrepancy in transmission is under investigation, but it may be due to the coarse voxelization of the reconstruction area, which generates errors when forward projecting through voxels that are only partially occupied by iron.
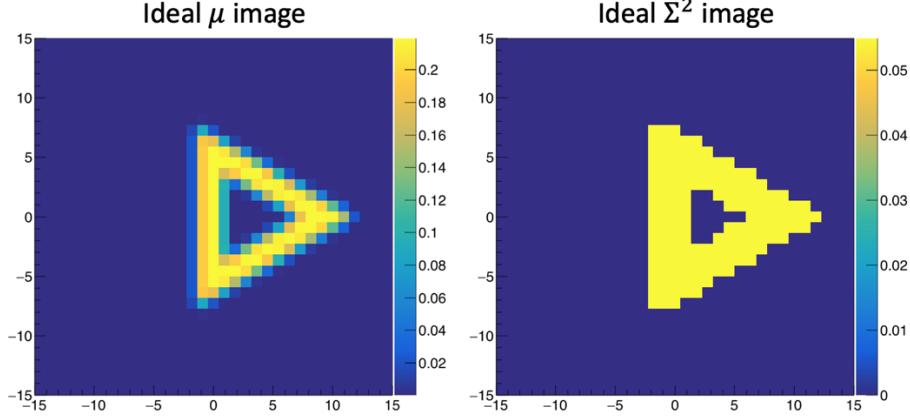

Figure 26. Idealized image values for attenuation coefficient and angular variance for the simulated geometry.
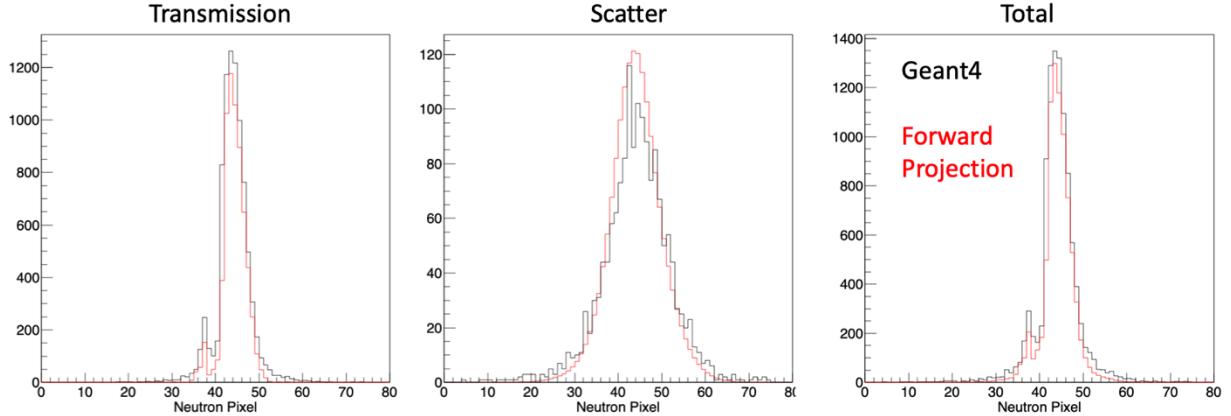
Figure 27. Comparisons of forward-projected data with Geant4 simulated data. Forward projected data are in red and GEANT-simulated data are in black for (left) transmission, (center) elastic scattering, and (right) total counts for a single alpha pixel.

7.7.4   Reconstruction results

Using the forward projection and back projection approaches just described, the SIRT algorithm was implemented to reconstruct simulated data. To keep the Σ^2 values from taking on arbitrary values when there is no scattering material present in a voxel, the voxel value was set to zero when the μ in that voxel

had a value below 0.01 cm^{-1}. In addition, the Σ^2 values were required to remain with a range of physically meaningful values. The largest angular widths exist for the lightest elements, and the Σ^2 value for hydrogen is approximately 32°, so a maximum allowed value of 35° was used. The starting values for μ were set to be zero, and the starting values for Σ^2 were set to be the middle of the allowed range. The Σ^2 values were held fixed for a prescribed number of iterations at the beginning of the reconstruction to keep them from immediately taking on extreme values after a few iterations. Once the μ values converge to values near their final values, the Σ^2 values were then allowed to vary along with the μ values.

As a first test, the idealized images were used to generate synthetic data via the forward projection. Then, this synthetic data was used as input to the iterative reconstruction to see whether the reconstructed object matches the ground truth. The results of this test are shown in Figure 28, where there were 125 total iterations and the variance was held fixed for the first 50 iterations. In this test, the final μ and Σ^2 values match the true values well, which provides confidence that the algorithm is implemented correctly.

When data from a full Geant4 simulation are used as the projection data, the results are less successful. Example results of this test are shown in Figure 29, where there were 125 total iterations and the variance was held fixed for the first 50 iterations. In this test, the final Σ^2 values go to extreme values as soon as they are allowed to vary. The suspected cause of this behavior is the absence of an isotropic term in the modeled differential cross section for scattering, but verifying this hypothesis requires more investigation.
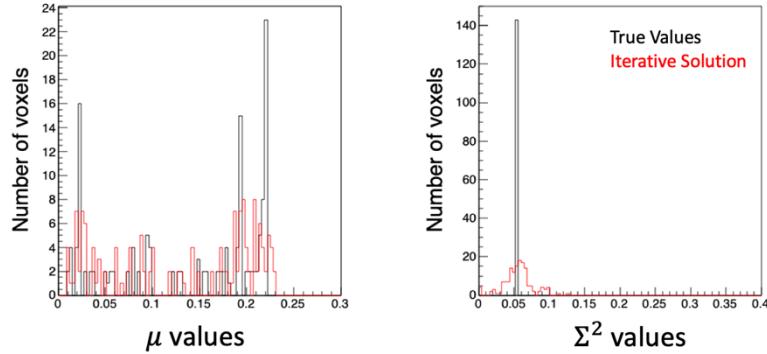
**Figure 28. Comparisons of reconstructed image values for forward-projected data used as the measured data. (Black) true and (red) reconstructed values of (left) attenuation coefficient and (right) scattering variance for reconstruction of forward-projected data.**
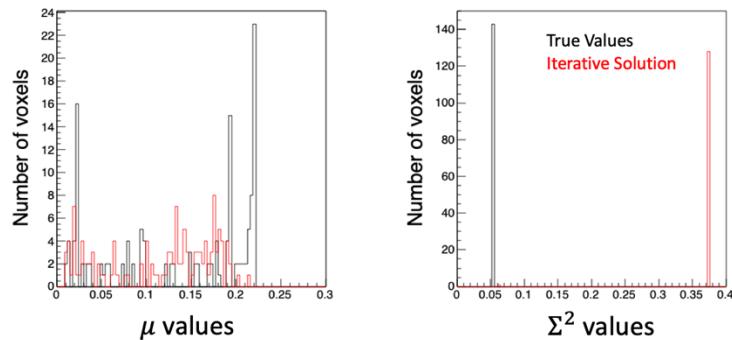
Figure 29. Comparisons of reconstructed image values for Geant4-simulated data used as the measured data. (Black) true and (red)

**reconstructed values of (left) attenuation coefficient and (right)
scattering variance for reconstruction of forward-projected data.**

8.   SUMMARY AND FUTURE PLANS

During this project, a fully 3D reconstruction framework was successfully developed for producing transmission and induced-reaction images using projection data from fast-neutron imaging systems that employ a wide cone beam rather than a spiral scan. The framework employs a total variation constraint to reconstruct smoother images and preserve sharp edges, consistent with machined parts. In addition, the code is fully parallelized, allowing reconstructions to be performed significantly faster compared to the iterative codes used prior to this work.

During the last year of this effort, refinement and validation efforts resulted in improvements to reconstructed image quality, especially near the boundaries of the coincidence cone and near the axis of rotation of the target. In addition, promising first steps were taken in the development of the joint estimation of transmission and small-angle scattering, where it was demonstrated that the physics of the interaction can be adequately modeled with the addition of a single quantity to each image voxel, the angular variance that characterizes the shape of the elastic scattering cross section.

There are two primary tasks for future work related to the development of the 3D reconstruction framework described in this report.

- Validation of the induced-reaction reconstructions needs to be completed, with particular attention paid to the magnitude of the image values. These values have been observed to be significantly different than expected and the cause needs to be identified.

- Development of the joint-estimation algorithm for transmission and small angle scattering needs to be completed. In particular, successful reconstruction of data produced by full Geant4 simulation and also by measurements have yet to be demonstrated. Modification to the forward projection is expected to be necessary to include an isotropic term.

A journal article summarizing the reconstruction framework developed as part of this work is in preparation. This work will serve as the starting framework for the reconstruction of other physics observables in API systems, such as inelastic gamma-ray reconstructions.

REFERENCE DOCUMENTS

[1]   P. A. Hausladen and J. Gregor, "Progress Update on Iterative Reconstruction of Neutron Tomographic Images," Report No. ORNL/LTR-2016/544, Oak Ridge National Laboratory, Oak Ridge, TN, September 2016.

[2]   P. A. Hausladen, M. A. Blackston, and J. Gregor, "Progress Update on Iterative Reconstruction of Neutron Tomographic Images, 2017, " Report No. ORNL/SPR-2017/490, Oak Ridge National Laboratory, Oak Ridge, TN, September 2017.

[3]   P. A. Hausladen, et al., "Induced-fission Imaging of Nuclear Material," 51st Annual Meeting of the Institute of Nuclear Materials Management, Baltimore, MD, July 2010.

[4]   M. A. Blackston and P. A. Hausladen, "Fast-neutron Elastic-scatter Imaging for Material Characterization," Proceedings of the 2015 IEEE Nuclear Science Symposium and Medical Imaging Conference, San Diego, CA, November 3, 2015.

[5] D. S. Lalush and M. N. Wernick, "Chapter 21: Iterative Image Reconstruction" *Emission Tomography, the Fundamentals of PET and SPECT*, Ed. M. N. Wernick and J. N. Aarsvold. Academic Press, 2004. 443–72.

[6] J. Gregor and J. A. Fessler, "Comparison of SIRT and SQS for Regularized Weighted Least Squares Image Reconstruction," *IEEE Transactions on Computational Imaging*, vol. 1, no. 1 (2015) 44–55.

[7] J. Gregor, P. Bingham, and L. F. Arrowood, "Total Variation Constrained Weighted Least Squares Using SIRT and Proximal Mappings," 4th International Conference on Image Formation in X-ray Computed Tomography, Bamberg, Germany, July 2016.

[8] A. Chambolle and T. Pock, "A First-order Primal-dual Algorithm for Convex Problems with Applications to Imaging," *Journal of Mathematical Imaging and Vision*, vol. 41, no. 1 (2011) 120–45.

[9] X. Lu, et al. "Mutual Information-based Multimodal Image Registration Using a Novel Joint Histogram Estimation," *Comput. Med. Imaging Graph*, vol. 32, no. 3 (2008) 202–9.

[10] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient Projections onto the L1-ball for Learning in High Dimensions," Proceedings 25th Intl. Conf. Machine Learning, Helsinki, Finland, 2008.

[11] C. T. Rueden, J. Schindelin, M. C. Hiner, et al., "ImageJ2: ImageJ for the Next Generation of Scientific Image Data," *BMC Bioinformatics*, vol. 18, no. 529 (2017).

[12] J. Schindelin, I. Arganda-Carreras, E. Frise, et al. "Fiji: An Open-source Platform for Biological-image Analysis," *Nature Methods* (2012) 676–82.

[13] S. Rose, M. Andersen, E. Sidky, and X. Pan, "Noise Properties of CT Images Reconstructed by use of Constrained Total-Variation Data-Discrepancy Minimization," *Medical Physics*, vol. 42 (2015) 2690–98.

[14] P. A. Hausladen, M. A. Blackston, and J. Gregor, "Progress Update on Iterative Reconstruction of Neutron Tomographic Images, 2018, " Report No. ORNL/SPR-2018/1029, Oak Ridge National Laboratory, Oak Ridge, TN, September 2018.