

Simulating and Reconstructing X-Ray CT Projections

Reece Garthwaite

Abstract

I have chosen to simulate x-ray transmission through digital phantoms, using energy dependent attenuation from NIST data, to generate sinograms that I then reconstruct using simple back projection.

1 Introduction

Computed tomography (CT) scans are a form of medical imagery that uses x-rays to create images of the body. CT scans reconstruct cross sectional images of the body from multiple projections of x-ray scans at a range of angles. This project aims to deepen my understanding of medical imaging techniques through the simulation and reconstruction of CT projections.

This work particularly focuses on incorporating energy-dependent attenuation coefficients, derived from NIST data, to accurately model materials like bone and soft tissue, thereby allowing for the investigation of phenomena such as beam hardening.

2 Methods

2.1 Background Physics

2.1.1 X-ray generation

X-rays are produced when high kinetic energy electrons are accelerated towards a positive anode in a x-ray tube. Tungsten is a common choice for the anode (due to its high melting point and high atomic number). Electrons come close to nuclei of the target, causing a deceleration and change in direction, converting kinetic energy into a spectrum of electromagnetic radiation. Incident electrons can ionise the material, removing an electron from the anode. As the electron orbit vacancy gets filled by a orbital shell electron in a further out shell a photon is emitted. As orbital energies and their differences are unique in atoms, this leads to what we call "characteristic radiation". [1]

Figure 1 [2] shows a diagram of a x-ray tube . Which helps to visualise the origin of the x-ray spectrum as electrons are emitted via thermionic emission

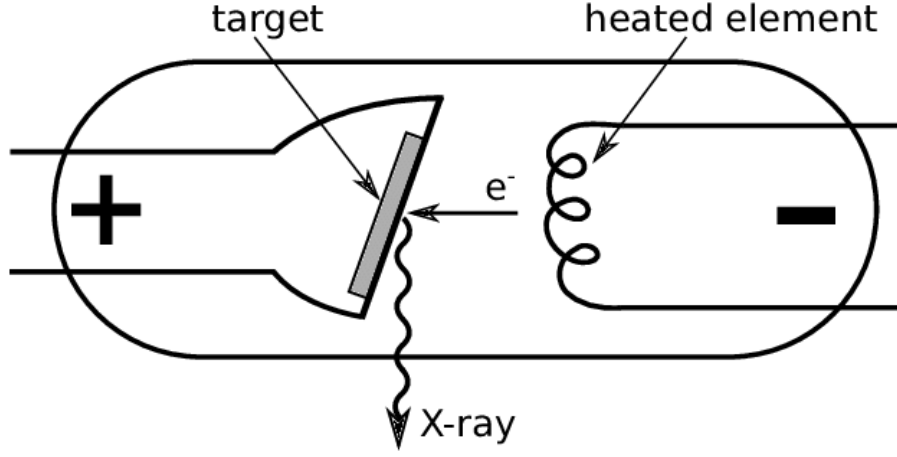


Figure 1: Schematic of a X-ray tube

and then accelerated toward a positively charged anode, where their interactions with the target produce both bremsstrahlung and characteristic radiation.

2.1.2 Attenuation of X-rays

As x-rays pass through matter, they are attenuated according to the Beer-Lambert law:

$$I(E) = I_0 \cdot e^{-\mu(E)x} \quad (1)$$

Where:

- I = transmitted intensity of radiation after passing through the material
- I_0 : Initial intensity
- μ : Linear attenuation coefficient
- x : Thickness of material

I have included μ to be energy dependent as this is what is observed in experiments, and allows us to look for an interesting phenomenon known as beam hardening. I calculate μ using:

$$\mu(E) = \left(\frac{\mu}{\rho} \right) \cdot \rho \quad (2)$$

Where $\left(\frac{\mu}{\rho} \right)$ is the mass attenuation coefficient from NIST [3]

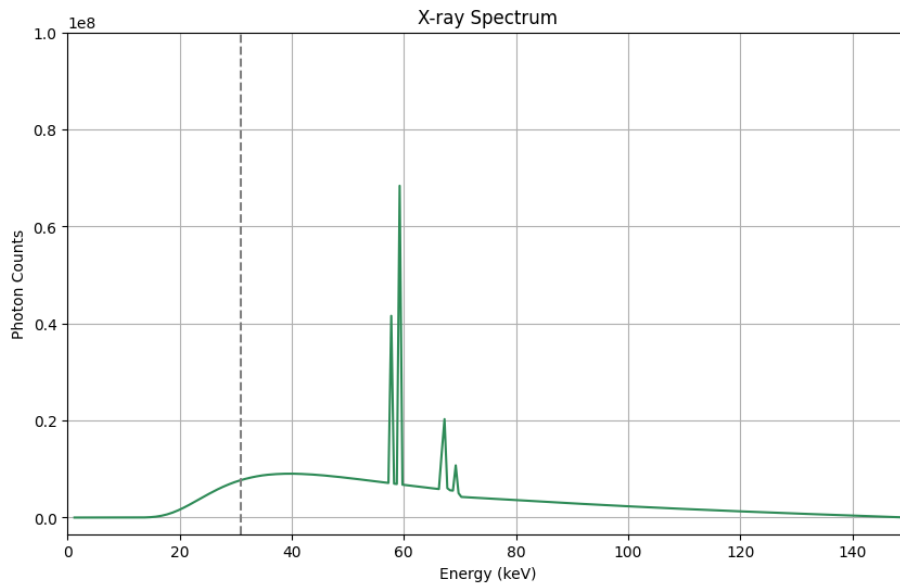


Figure 2: X-ray Spectrum from a x-ray tube with a Tungsten target

2.2 Investigating Beam Hardening

Beam hardening occurs during x-ray attenuation as higher energy (hard) X-rays penetrate dense materials more effectively than the low energy (soft) X-rays. This has the effect of shifting the peak of the bremsstrahlung radiation towards higher energies. Beam hardening cannot occur for monochromatic X-rays, as the average energy will not change, so we can predict that the characteristic radiation will not harden. I used *spekpy*, a dedicated x-ray spectrum simulation toolkit, and *matplotlib* to generate a typical spectra from a Tungsten x-ray tube and plot for energies upto 150 keV. I used a 2mm Aluminium filter to remove low energy, non-diagnostic x-rays. We can use *spekpy*'s built in filter function for this but will also write our own filtering code shortly.

In Figure 2 we can see the sharp characteristic peaks that correspond to Tungsten. Writing code for filtering using NIST data and interpolation to fit the number of data points required.,

```
import pandas as pd
from scipy.interpolate import interp1d

# Load csv
bone_data = pd.read_csv("bone.csv", names=["Energy_MeV", "Mu_Rho"],
                        header=None)#

# Convert energy to KeV
bone_data["Energy_KeV"] = bone_data["Energy_MeV"] * 1000

# Average out duplicates by grouping
```

```

bone_data = bone_data.groupby("Energy_KeV", as_index=False).mean()

# Convert to linear attenuation ( $\mu = \mu/\rho * \rho$ )
bone_density = 1.85 # g/cm3
bone_data["Mu"] = bone_data["Mu_Rho"] * bone_density # [cm-1]
mu = bone_data["Mu"].to_numpy()

# Create interpolation function for attenuation coefficients
mu_interp = interp1d(bone_data["Energy_KeV"], bone_data["Mu"],
                     bounds_error=False, fill_value="
                     extrapolate")

mu_at_spectrum_energies = mu_interp(energies)

thickness_cm = 0.5
I0 = s.get_spk()

I_filtered = I0 * np.exp(-mu_at_spectrum_energies * thickness_cm)

```

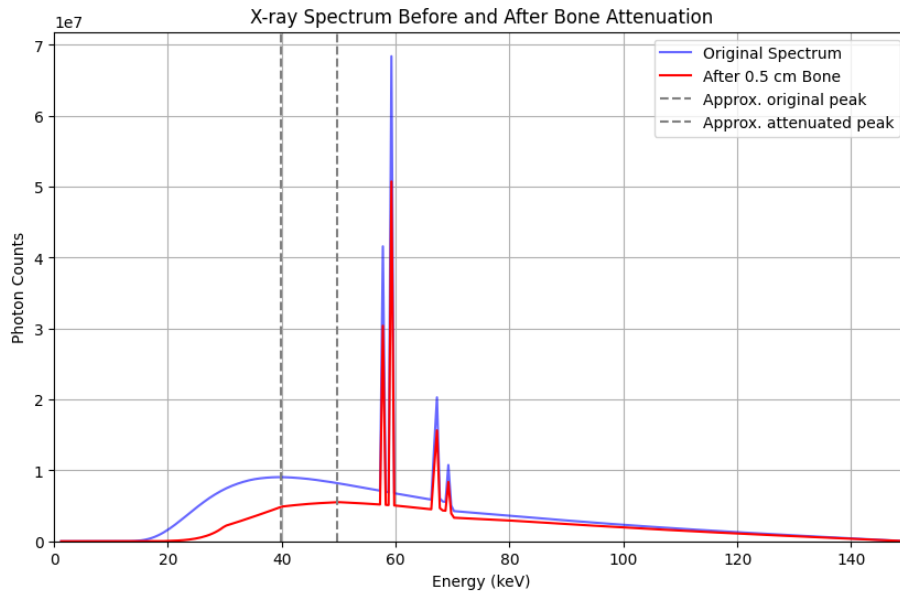


Figure 3: Comparing X-ray spectra before and after passing through 0.5cm of bone

Figure 3 shows beam hardening on the X-ray spectrum. The lower-energy (soft) x-rays are preferentially attenuated, resulting in a spectrum that is skewed toward higher energies. Lower-energy photons are absorbed more readily than high-energy ones. As a result, the transmitted spectrum becomes "hardened," with its average energy increasing. We also don't see any hardening for the monochromatic peaks, as expected.

Beam hardening is significant in image reconstruction, as it may result in

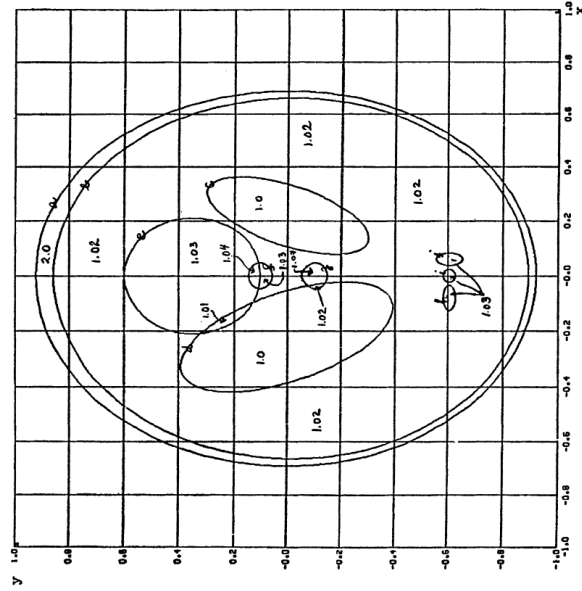


Fig. 1 The centers, axes, tilting angles, and gray levels for each ellipse are given in Table 1. In an attempt to be consistent with known facts about the human head, the skull in the figure is about twice as dense as the interior tissues, the ventricles are twice as dense as the interior tissues, and the optic chiasm (water) are least dense (1.0). While gray matter has density 1.02 and fills the interior of the head except for tumors (1.03) at h, i, j, g, and f. Note that in the regions (s-g-d) and e-f the density is 1.04, in d-f, it is 1.01, and in g-f it is 1.02. In comparing dimensions with a human skull with approximate axes dimensions of 15 cm., 11 cm., and 15 cm., the small circular tumor at the bottom is 3.5 cm. in diameter. Three dimensional plots of $f(x,y)$ are given in Fig. 2.

Figure 4: Shepp-Logan head phantom image

characteristic artefacts. CT beam hardening artefacts are known as: streaking artefacts, where dark bands appear between dense structures, and cupping artefacts, where the center of a homogeneous object appears less attenuating than the periphery [4].

2.3 Phantoms and Sinograms

2.3.1 Generating Phantoms

In medical scans, phantoms are used as stand-ins for human tissues to ensure that systems and methods are working correctly. One common such phantom is the Shepp-Logan phantom [5]. The Shepp-Logan phantom is a standard benchmark for CT reconstruction algorithms due to its complex structure that still models real world conditions.

Using Jan Hrach's python code, [6], I can recreate the Shepp-Logan phantom with constant attenuation coefficients.

Another phantom I created was a simple circle of bone surrounded by a larger circle of soft tissue. This was an easy way to test using the energy dependent attenuation coefficients. The *circlemask* function is used to define circular regions on the phantom, which can then be assigned with properties for different materials. Using a phantom with simplified geometry allowed for clearer observation of attenuation characteristics.

```
def circlemask(radii, centre_x, centre_y, width, height):
```

```

    x, y = np.ogrid[:width, :height]
    mask = (x-centre_x)**2 + (y-centre_y)**2 <= radii**2
    return mask

width, height = 180,180
s = sp.Spek(kvp=150,th=12)
energies = s.get_k()
# Load each material from a csv file
mu_bone = load_mu_data(r"code\bone.csv", density=1.85,
                      energy_range_KeV=energies)
mu_tissue = load_mu_data(r"code\soft_tissue.csv", density=1.06,
                        energy_range_KeV=energies)
E=len(energies)

phantom = np.zeros((height, width, E))
bone_mask = circlemask(10,width//2,height//2,width,height)
thigh_mask=circlemask(40,width//2,height//2,width,height)

# Background
phantom[:, :, :] = 0 # broadcast over all pixels

# Thigh region
phantom[thigh_mask, :] = mu_tissue

# Bone region
phantom[bone_mask, :] = mu_bone

```

Using matplotlib to display these phantoms:

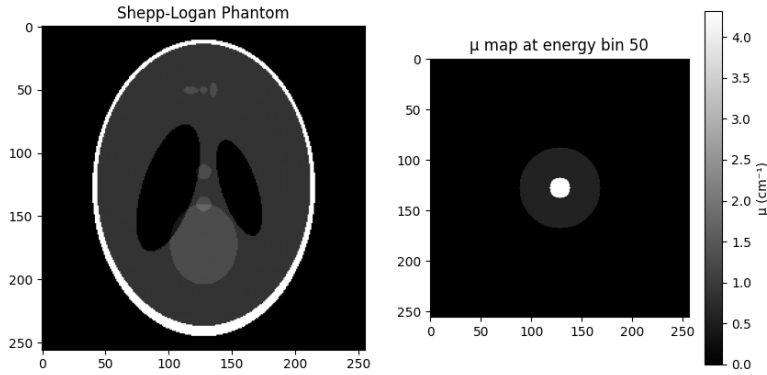


Figure 5: Left: Shepp-Logan phantom. Right: Circular phantom with central bone inclusion (μ map at energy bin 50).

2.3.2 Sinograms

A sinogram is a 2D representation of the raw projection data acquired in computed tomography (CT). It organizes the attenuation measurements from dif-

ferent angular views around an object, where each row corresponds to a 1D projection taken at a specific angle. Before we can reconstruct the cross-sectional image of the scanned object, the initial x-ray attenuation measurements are compiled into this sinogram format.

To understand image reconstruction, we need to define projections. A projection in CT scans represents the total attenuation of x-rays along a specific path of the object, mathematically this is represented as a line integral. At an angle θ , the parallel beam geometry represents a new coordinate system (r, s) in which the intensity profile $I_\theta(r)$ is measured. Other beam geometries such as fan or cone geometries require different methodology.

$$\begin{pmatrix} r \\ s \end{pmatrix} = R \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Conversely,

$$\begin{pmatrix} x \\ y \end{pmatrix} = R^T \begin{pmatrix} r \\ s \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} r \\ s \end{pmatrix}$$

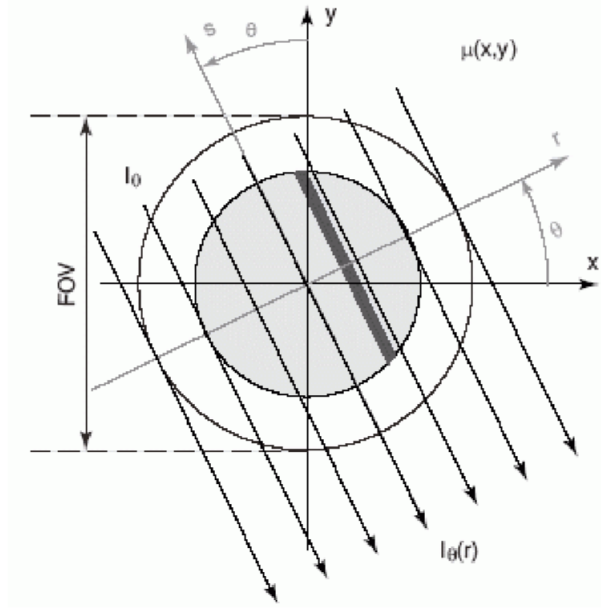


Figure 6: The object is defined in the fixed Cartesian coordinate system (x, y) . For a projection at angle θ , a new coordinate system (r, s) is defined, where r is the perpendicular distance from the origin to the x-ray ray, and s is the coordinate along the ray path

Using parallel beam geometry and a fixed angle θ , the measured intensity at position r is the integrated mass attenuation along $L_{(r, \theta)}$ is given by the

Beer-Lambert law:

$$I_{\theta}(r) = I_0 \cdot \exp\left(- \int_{L(r,\theta)} \mu(x,y) ds\right) \quad (3)$$

$$= I_0 \cdot \exp\left(- \int_{L(r,\theta)} \mu(r \cos(\theta) - s \sin(\theta), r \sin(\theta) + s \cos(\theta)) ds\right) \quad (4)$$

Each Intensity profile $I_{\theta}(r)$ is transformed into an attenuation profile $p_{\theta}(r)$, by taking the negative natural logarithm of the ratio of transmitted intensity to incident intensity:

$$p_{\theta}(r) = -\ln \frac{I_{\theta}(r)}{I_0} = - \int_{L(r,\theta)} \mu(r \cos(\theta) - s \sin(\theta), r \sin(\theta) + s \cos(\theta)) ds \quad (5)$$

This attenuation profile is what our detector measures at each angle. Going back to the circle phantom example, we can write some code to simulate this:

```
# Sum attenuation coefficients along the ray path (top to bottom)
# for each (width, energy).
# 'mu_total' is 2D (width, energy), representing total attenuation
# per ray per energy.
mu_total = phantom.sum(axis=0)

I = I0 * np.exp(-mu_total * pixel_thickness_cm)

# Sum transmitted intensities across all energies for each detector
# position.
# 'incident_intensity' is 1D (width,), representing the final
# polychromatic projection profile.
incident_intensity = I.sum(axis=1)

# Visualize detector readout (log form)
detector_readout = -np.log(incident_intensity + 1e-6) # add epsilon
# to avoid log(0)
```

Plotting our result we get:

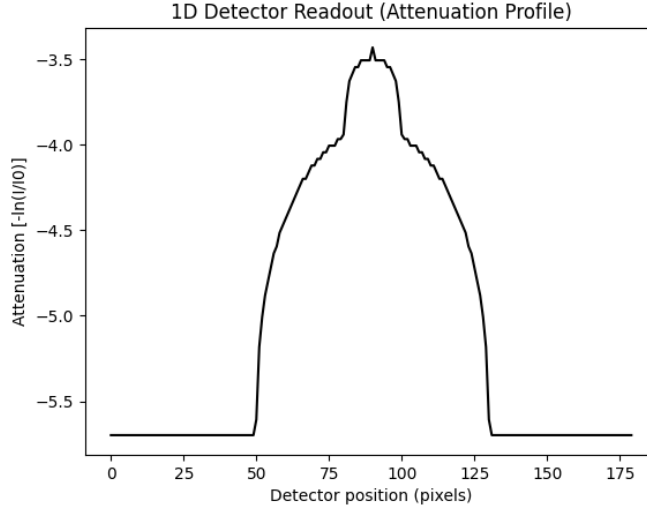


Figure 7: 1D detector readout for our circular phantom, we see the increase in attenuation in the middle as we go through the bone.

If we arrange projections from all angles (only $(0, \pi)$ as $(\pi, 2\pi)$ are redundant for parallel beams) side by side as a 2D image, we get a sinogram. We can view this 2D dataset $p(r, \theta)$ for our circular phantom in Figure 8, this sinogram was generated using `scipys ndimage.rotate` to simulate the projections. Another, more interesting sinogram can be found from our more complex Shepp-Logan phantom, which we can see in Figure 9. We can notice sinusoidal curves throughout the image, motivating the name sinogram.

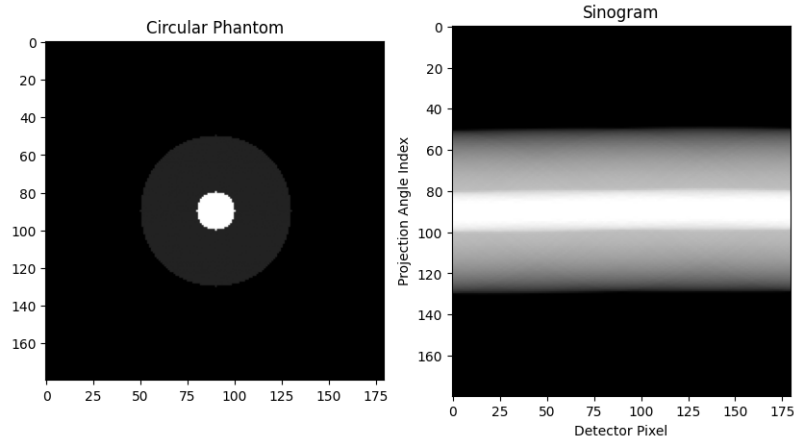


Figure 8: Sinogram for our circular phantom, note how it is constant in θ due to the rotational symmetry of the circle.

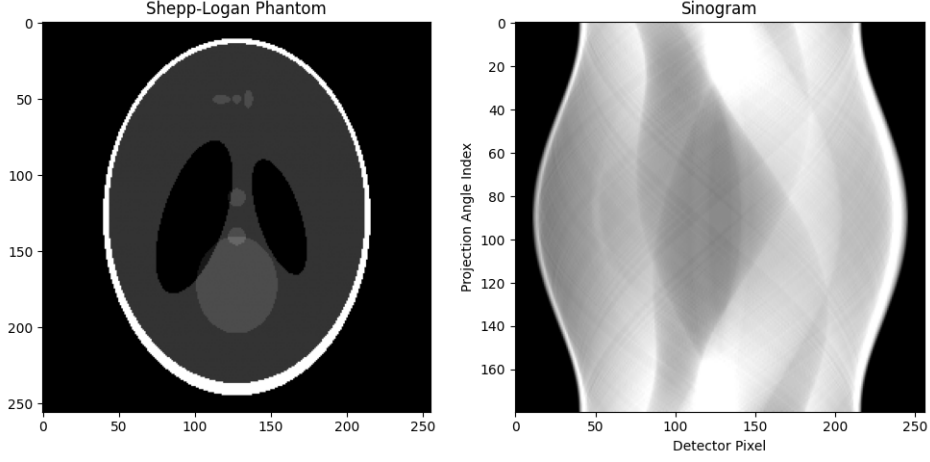


Figure 9: Sinogram for the Shepp-Logan phantom.

The transformation of a function $\mu(x, y)$ into $p(r, \theta)$, is known as the Radon transform defined as:

$$p(r, \theta) = R\{\mu(x, y)\} \quad (6)$$

$$= \int_{-\infty}^{\infty} \mu(r \cos \theta - s \sin \theta, r \sin \theta + s \cos \theta) ds \quad (7)$$

We can therefore phrase the problem of CT reconstruction as finding an inversion formula for the Radon transform [7].

2.4 Inverse Radon Transform and Back projection

References

- [1] D. Tafti and C. V. Maani, “Physics, x-ray production,” 2023. Accessed: 2025-06-06.
- [2] J. Mason, *Quantitative cone-beam computed tomography reconstruction for radiotherapy planning*. PhD thesis, 11 2018.
- [3] “X-Ray Mass Attenuation Coefficients — nist.gov.” <https://www.nist.gov/pml/x-ray-mass-attenuation-coefficients>. [Accessed 09-06-2025] at <https://www.nist.gov/pml/x-ray-mass-attenuation-coefficients>.
- [4] A. Murphy, “Beam hardening — radiology reference article — radiopaedia.org,” Oct 2016.
- [5] L. A. Shepp and B. F. Logan, “The fourier reconstruction of a head section,” *IEEE Transactions on Nuclear Science*, vol. 21, p. 2143, Jun 1974.

- [6] J. Hrachovec, “phantom.py,” n.d. Accessed: 2025-06-09 at <https://jenda.hrach.eu/f2/cat-py/phantom.py>.
- [7] J. Beatty, “The radon transform and the mathematics of medical imaging,” Master’s thesis, Colby College, 2012.