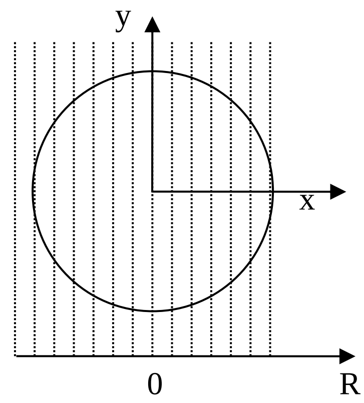
Computed Tomography Project

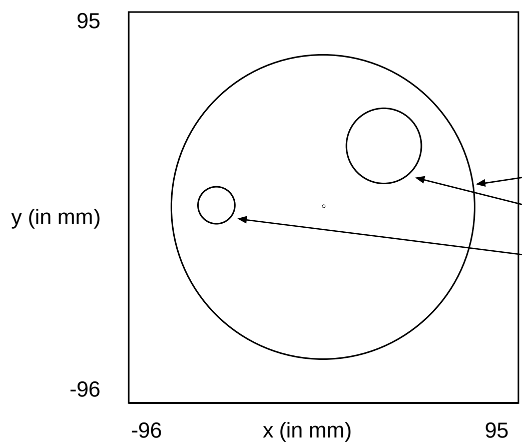
Official due date: Tuesday, Dec. 5, 2023
Due date without penalty: Friday, Dec. 8, 2023

In this project, you will implement and compare different methods of reconstructing an image from a set of projections. The methods to be implemented are: filtered-backprojection (FBP) and direct Fourier interpolation (FI). We will first generate a sinogram for the set of circular objects described below – knowing the object will help debug your algorithm. Later you will apply one of the reconstruction methods to variations on the data as well as a mystery object.

We will assume a parallel ray imaging geometry with the $\theta = 0$ projection is shown here →



The reconstructed image should be 192 by 192 pixels with a pixel size of 1 by 1 mm (field of view is 192 mm). The number of radial samples is 192, with $\Delta R = 1$ mm. The radial samples are placed as shown to the right (for a subset of the 192 lines). Note that the radial samples are at R = [-N/2:N/2-1]*ΔR. The angular sampling covers almost π radians, i.e., $\theta_k = \pi k/N_\theta$, where k=0,1,2,…,(N_θ-1) and N_θ is the number of angular samples. Note the last angle is not at exactly π radians. The object is shown here:



object center (mm)	radius (mm)	atten coeff (mm-1)
0,0	75	0.1
30,30	18	0.2
-53,0	9	0.3

Administrative items:

- You may work alone or with a partner for this project (any partner must be different from those you worked with in other projects). Please put your partner's name on your solutions.
- The partnering is not intended to be a "divide and conquer" approach – that is, you may not split up the problem and then share solutions. I expect you to do your own work, but when you get stuck, you may ask your partner or the instructor for help. Think of this as an exam, but one in which on may ask for help from one other person.
- Getting help from anyone other than the instructor or your partner or copying solutions (even from your partner) will be considered a violation of the engineering honor code.
- Each person must hand in their own solution.
- Prepare a report describing the tasks below. Hand in answers to questions below, all requested plots/graphs/images and copies of your final ".m" files. For each question/task please write 2 or

3 sentences about what you did, what equation you implemented, and about the appearance of the resultant images, and explain why you think causes such appearance, etc. Say something interesting!

- Title and label the axes on all plots and images.
- Files to get you started can be found on Canvas.
- If you are having problems, please come to the office hours and/or e-mail me the questions.

Questions/Tasks:

1. Determine the required number of angular samples (N_θ). Let N_θ be a multiple of 4 (parts 9, 10).
2. Synthesize and display a N_θ by N_r sinogram for the object described above using analytical formulae for the path length through a circular object. This is already implemented in the Matlab template. Create an image of the sinogram. Describe features of the sinogram as they relate to the original object.
3. Implement a simple backprojection algorithm and reconstruct the image without filtering. One easy way to do this in Matlab is to backproject at $\theta = 0$ to a temporary image by replicating the project for all rows and then rotating that image to the appropriate angle of backprojection. Superimpose these temporary images to generate the final image.
4. Now filter each projection and plot the projection at $\theta = 0$ before and after filtering. Please verify that the projection is real valued (there may be very small components in the imaginary part, say $<10^{-6}$, which you should set to zero by using only the real part.
5. Generate the FBP image by back projecting the filtered sinogram. Set any negative pixels to zero and generate an image. Please describe any artifacts that you see in the images.
6. Use Matlab's `griddata` or `interp2` functions to generate a 192x192 Fourier array for the object. Take the inverse Fourier transform to generate the image for the FI method. Please describe any artifacts that you see in the images.
7. We will now do FI with more finely sampled data. Zero pad the sinogram (to 384 in the R dimension) before taking Fourier transform. The Fourier data will now be 2x more finely sampled (in kr). Similarly, make the target Fourier grid more finely sampled (now 384x384) and take the inverse FT. Lastly, extract the central 192x192 of the reconstructed image. Compare to part 6.
8. Plot the profile through the image along the $y = 0$ line for all three methods (FBP, FI, oversampled FI). You may need to normalize to put on the same scale. Comment on any inaccuracies or artifacts.
9. Subsample the original sinogram by a factor of 4 ($N_\theta/4$ angular samples) and reconstruct the image. Reconstruct using both approaches and comment on any inaccuracies and artifacts.
10. Now create a sinogram using only the first ¼ of the projections. Reconstruct using one of the approaches and comment on any inaccuracies and artifacts.
11. Load the mystery sinogram ("mys22.mat") and examine it. Reconstruct the object by any method and generate the image. Note this data set has a different number of projection angles, 360 angular locations from 0 to 179.5 degrees. Can you guess the image from the sinogram?

Tips:

1. Recall calculation of sinogram

$$g_\theta(R) = \int \mu(x,y)\delta(x\cos\theta + y\sin\theta - R)dxdy$$

   Since there are three objects in Figure 1, we can simplify the projection as:

$$g_\theta(R) = \sum_{i=1}^{3} \mu_i \times (length\ of\ \ segment)_i$$

   μ_i is "amp" in the ct_template.m. This is already implemented in the code.

2. You will have to call the backprojection reconstruction many times, so you may to create a function that implements backprojections from a sinogram. To do your backprojections, you may want to use the imrot3 function.

3. Note imrot3 function – rotates and image **counterclockwise**. For rotating π/3 clockwise, should use `a = imrot3(a, -pi/3, 'bilinear');`
   and the rotation is around the (N/2+1,N/2+1) point. There is an `imrotate` function in Matlab, but it rotates around a different point than what we are specifying here and expects angles in degrees.

4. Take a good look at your sinogram and determine if it was obtained by rotating R clockwise or counterclockwise. Either way is acceptable, but it will determine the rotation direction to reconstruct the backprojection image.

5. Note `fftshift` works differently for vectors and matrices.

6. Filtering for FBP - Can do it either in the Fourier domain or spatial domain or any filter you want to you. Doing it in the Fourier domain is generally easier, but beware of circular convolution.

7. For FI, `interp2` or `griddata` can be used. It might be a good idea to look at the 2D Fourier data before `ift2` to verify that the Fourier data are complete and look ok.