Question 1

We want to show that the solution for the equation

$$\hat{x} = \underset{x}{argmin}\left\{\frac{1}{2}\|x-y\|_2^2 + \frac{\lambda}{2}\|x\|_2^2\right\}$$

Is

$$\hat{x} = \frac{1}{1+\lambda}y$$

Differentiation:

$$\frac{\partial}{\partial x}\left(\frac{1}{2}\|x-y\|_2^2 + \frac{\lambda}{2}\|x\|_2^2\right) = (x-y) + \lambda x$$

Simplify and equal to zero:

$$\hat{x} - y + 2\lambda\hat{x} \overset{!}{=} 0$$
$$\hat{x}(1+\lambda) = y \Rightarrow \hat{x} = \frac{1}{1+\lambda}y$$

To show that we get a minimum, we will examine the second derivative:

$$\frac{\partial^2}{\partial x^2}\left(\frac{1}{2}\|x-y\|_2^2 + \frac{\lambda}{2}\|x\|_2^2\right) = \frac{\partial}{\partial x}(x-y+\lambda x) = 1+\lambda$$

Since $1+\lambda > 0 \; \forall \; \lambda > -1$, we conclude that for $\lambda > -1$ the solution corresponds to a minimum as desired.

Hence, the solution for the equation is:

$$\hat{x} = \frac{1}{1+\lambda}y, \qquad \lambda > -1$$

Question 2

As we can see from the plots, there isn't a significant difference between the solutions. This is because the chosen λ values are relatively close to each other, leading to only small changes in the shrinkage factor:

λ	$1/(1+\lambda)$
0.01	0.99
0.05	0.95
0.1	0.9
0.2	0.82

In addition, the solution is not sparse, as seen directly from the plots — many of the values are close to zero but not exactly zero. This behavior is expected with l_2 regularization, which tends to shrink values but not eliminate them completely. This is likely due to a known property of l_2: as the regularization parameter increases, the solution is pushed more strongly toward minimizing the overall magnitude of the coefficients, but the shrinkage is proportional to the size of each coefficient. In other words, larger coefficients are penalized more, while smaller ones are less affected. In this case, most of the noise in y consists of small values, so even with regularization, they remain relatively unchanged, resulting in a solution that is not sparse.

Question 3

$$\hat{x} = \underset{x}{argmin}\left\{\frac{1}{2}\|x - y\|_2^2 + \lambda\|x\|_1\right\}$$

As mentioned, since the variables $\hat{x}$'s are independent, we can minimize each of them separately by solving:

$$\underset{x_i}{argmin}\left\{\frac{1}{2}(x_i - y_i)^2 + \lambda|x_i|\right\}$$

Differentiation:

$$\frac{\partial}{\partial x_i}\left(\frac{1}{2}(x_i - y_i)^2 + \lambda|x_i|\right) = x_i - y_i + \lambda \cdot sign(x_i)$$

This gives the following cases:

- For $x_i > 0$

$$\hat{x} - y + \lambda = 0 \Rightarrow \hat{x} = y - \lambda$$

- For $x_i < 0$

$$\hat{x} - y - \lambda = 0 \Rightarrow \hat{x} = y + \lambda$$

At $x_i = 0$, $sign(x_i)$ is undefined, so the equation is undefined, so we have to check the behavior around zero.

Choosing optimal x:

Since $sign(x)$ is equal to (1) when $x > 0$ and equal to (-1) when $x < 0$, we need to ensure the corresponding solutions are positive or negative, respectively.

Hence:

- If $y_i > \lambda$ the solution is $x_i = y_i - \lambda$ (since $x_i > 0$)
- If $y_i < -\lambda$ the solution is $x_i = y_i + \lambda$ (since $x_i < 0$)

Now, let's consider the condition $-\lambda \leq y_i \leq \lambda$:

Let's check what happens for $x_i = 0$:

In this situation, the sub-gradient equation becomes:

$$0 - y_i + \lambda \cdot 0 = 0 \Rightarrow -y_i = 0$$

This satisfied if $y_i = 0$, which is a possibility when $|y_i| \leq \lambda$

However, the condition $|y_i| \leq \lambda$ means that y_i is small enough for this balance to occur at $x_i = 0$.

If $|y_i| \leq \lambda$, the optimal x_i can be zero because:

a. For values of y that are small enough (i.e, $|y_i| \leq \lambda$), the regularization term $\lambda|x_i|$ is strong enough that the optimal solution would be at $x_i = 0$.
b. The solution $x_i = 0$ minimize the function because any nonzero value of x would increase the cost due to the $\lambda|x_i|$, without sufficiently improving the least squares term $\frac{1}{2}(x_i - y_i)^2$

Therefore, when $|y_i| \leq \lambda$, the regularization outweighs the fit to the data, leading to the solution $x = 0$.

Thus, when generalizing from (x_i, y_i) to (x, y) the solution to the optimization problem is:

$$\hat{x} = \begin{cases} y - \lambda & , y > \lambda \\ y + \lambda & , y < -\lambda \\ 0 & , |y| \leq \lambda \end{cases}$$

Or, more compactly:

$$\hat{x} = sign(y) \cdot \max\{|y| - \lambda, 0\}$$

Question 4

When y is small compared to λ, we are looking at the case where $|y| < \lambda$, meaning the function will set the value to zero. When y is large compared to λ, we are looking at the case where $|y| > \lambda$, and the value is preserved. This is thresholding behavior, small magnitude noise (or insignificant values) are eliminated, and large values are conserved.

Question 5

As expected, and as shown in the various plots, the solution is sparse. This outcome is anticipated due to the thresholding discussed in the previous question – where small y values are pushed to zero - thereby promotes sparsity.

Question 6

The reconstructed signal is not sparse since it does not contain any zero. The signal has many non-zero values (108), this probably due to aliasing from the equispaced sampling. The equispaced undersampling leads to gaps in the frequency domain, which causes aliasing in time, meaning the sparse structure gets distorted.

Question 7

The randomly reconstructed signal has fewer near-zero values than the equispaced one, but it has less spikes. If we compare it to the original signal, it is possible to see that the spikes are in similar locations to the spikes of the original signal.

It is not possible to directly reconstruct the original signal using the inverse FFT of zero-filled data, because it assumes the missing frequencies are zero, which is almost never true.

Why does it resemble denoising? The missing data causes distortions in the result, which leads to weird patterns or artifacts in the time domain. The restored signal does not look like the original one. So just like noise, the signal looks corrupted, and our problem is to tell what the original signal is.

However, upon closer inspection, the noise is not entirely random, and it might be even structured, which might help us to reconstruct the signal.

Question 8

The under-sampling in equal spaced locations in the frequency domain leads to aliasing in the time domain (since in this case we do not hold to Nyquist condition). The aliasing creates overlapped signal, which causes large and misleading components. That cannot be undone

by soft thresholding since these components do not look like noise, and the function can't distinguish between the true signal and the aliasing.

The random under-sampling worked since the missing frequency information causes noise-like interference in time, which get spread across all time samples. Since the original signal is sparse, one can use the soft thresholding to eliminate the noise and to keep the sparse signal.

Let's examine the interference that spread across all time samples:

The random sampling of the Fourier transform can be referred to as multiplication with a binary vector, randomly chosen. The multiplication in the frequency domain is equal to convolution in the time domain. Since the vector is binary and randomly chosen, it is not smooth and not periodic. This means that the inverse transform will lead to noise, complex values and spread-out over-all time signal (When looking at Fourier transform, signal that is localized in one domain will be spread out across the other).

Now we have a noise-like interference that spreads over all time, but not large anywhere, and when the true signal is sparse the soft thresholding can suppress these interferences and recover the signal.

Question 9

Note to self, interpretation of the phase map:

δ is the undersampling ratio and ρ is the sparsity per measurement.

x-axis represents δ:

Far right = full measurements (no undersampling)

Far left = very few measurements (highly compressed)

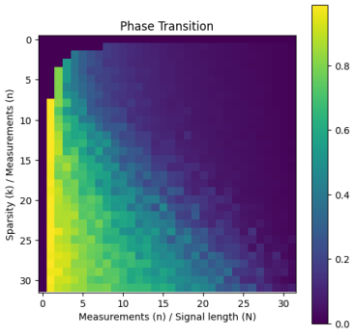
y-axis represents ρ:

Bottom = super sparse signals

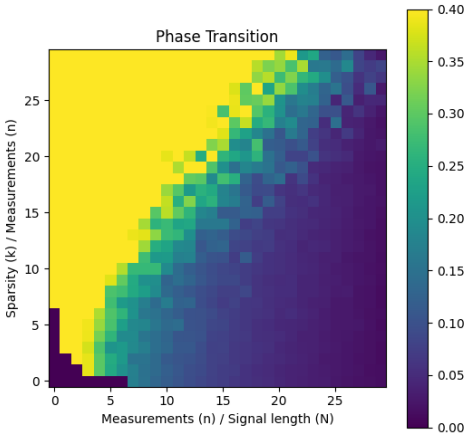
Top = dense signals (bad for Compressed Sensing)

Answer:

When using imshow(phase_transition) alone, it is possible to see phase like transitions, but not the same as in the paper. The phase transition appears reversed, with boundary effects and it's more difficult to see clear transition than in the paper:



After setting origin='lower' (which serts the [0,0] index in the lower left corner), and setting the maximum value to differentiate between colors (errors) to 0.4 we get a more similar phase transition plot:



In conclusion, it is possible to see a phase transition in the diagram. Similar to the figure from the mentioned paper (link), the processed diagram shows clear boundary between successful and unsuccessful recovery regions.

However, the transition in the homework diagram is not as sharp or precise as in the paper. This result is expected since the homework simulation used fewer trials and has lower resolution (32×32 grid instead of a finer one).

These differences (between paper and homework) result in more 'direct' transition in the paper case, which makes sense given the lower statistical confidence and the averaging method used. Nevertheless, the overall shape and behavior still align with what we would expect.

Question 10.1 - Plot the masked wavelet coefficients. What has been thresholded?

While setting f to 0.2, we are keeping the top 20% of the largest coefficients (magnitude wise) and setting the rest to zero.

In this scenario, the wavelet-mother is defining such that it will detect local changes (or transitions in the signal). Since this image does not hold many changes, or in other words – the image is smooth, the inner product between the wavelet and big part of the image is small, which yields small coefficients.

These small coefficients will probably represent noise in the image, and not it's main objective – the edges of the brain tissues. According to this understanding, we can tell that the noise has been "thresholded".

Question 10.2 - What, in your opinion, is the sparsity level of the image? Provide a reconstruction and difference image to support your argument.

Firstly, for a naked unprofessional eye, there is not a big difference between the reconstructed images, even with only 2.5% of the coefficients and the difference image looks like blurred image and not like a real image (except for the 2.5% image, where we can see starts of edges). This tells us that most of the "important data" is a small part of the whole image, thus meaning the sparsity level is high.

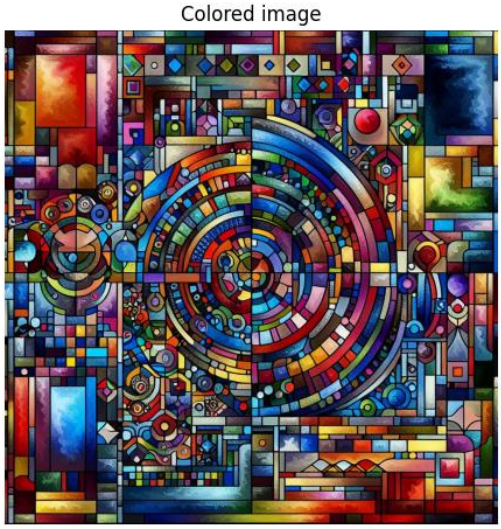
Secondly, one can look at the amount of non-zero coefficients and consider the amount of the zero coefficients. When looking at both 20% and 2.5% thresholds, we get these results:

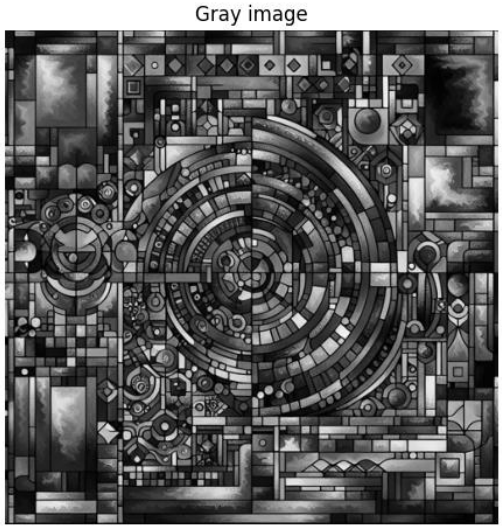
	20% threshold	2.5% threshold

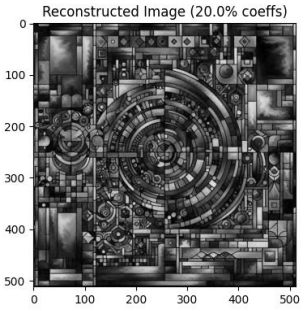
Image size	252144	252144
Sparsity level [#non-zero]	54428	6553

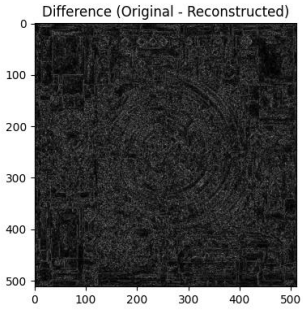
Even though there is a big difference in the sparsity level, there is not a big visible difference between the reconstructed images, which reinforces the argument that the image is sparse.

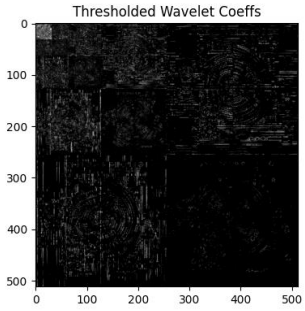
Thirdly, let's explore a different image, more detailed and with more colors image, which created using co-pilot:

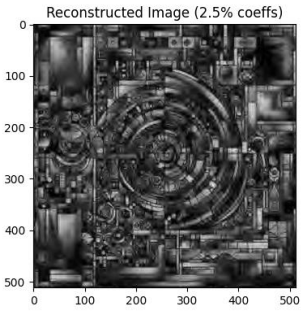


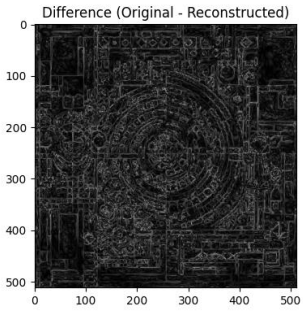


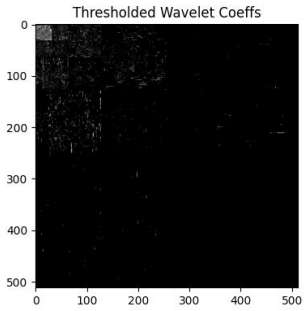












More images are in the notebook.

As we can see, there is a big difference between the 2.5% and the 20% image, when the 2.5% image is blurrier and the reconstruction cannot restore the small details/rapid changes in the image. This tells us that the sparsity level was significantly lower for this image, as expected, since colored or high-texture images tend to spread energy across more wavelet coefficients.

This comparison confirms that the MRI image is naturally sparse in the wavelet domain.

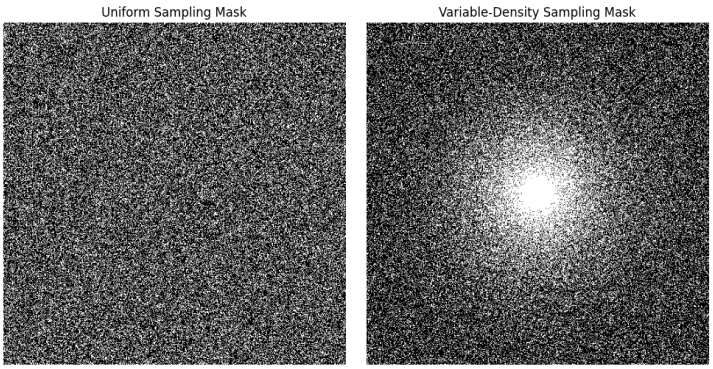
Question 11

White noise looks like random speckles with different intensities and no discernible pattern.

For the uniform mask, the difference image and the reconstructed image looks "cloudy". This suggests that the noise (or in that case, the aliasing) is not white random noise, since the artifacts has kind of a "swirly" pattern. There are small changes which are visible in the original image and are not visible in the reconstructed one. This tells us that the changes has been going through aliasing, and now the reconstruction is damaged.

For the variable mask, the difference image does not hold large-scale structures, and it looks more like random-residual errors. This situation is more like white noise. The variable-mask-reconstructed-image is sharper than the uniform-mask-reconstructed-image which tells us that the variable-reconstruction is better.

Let's try to understand why this happens by viewing the masks:



Reminder: value 1 is represented by the color white, while the value 0 is represented by the color black.

From these masks we can see that the variable-density sampling puts more samples in the low-frequency region. The missing data in high-frequency regions probably looks more noise-like when reconstructed. So, using this mask allows us to capture the dominant low frequency components, and the aliasing that results from the undersampling tends to spread out.

Reconstruction from Random Sampled k-Space Data

First, we want to create a soft threshold function which can deal with complex data. To do so we will observe the original function.

$$\hat{x} = sign(y) \cdot \max\{abs(y) - t, 0\}$$

$$sign(y) =_{assume\ y\neq 0} = \frac{y}{|y|} = \frac{|y| \cdot e^{j\angle y}}{|y|} = e^{j\angle y}$$

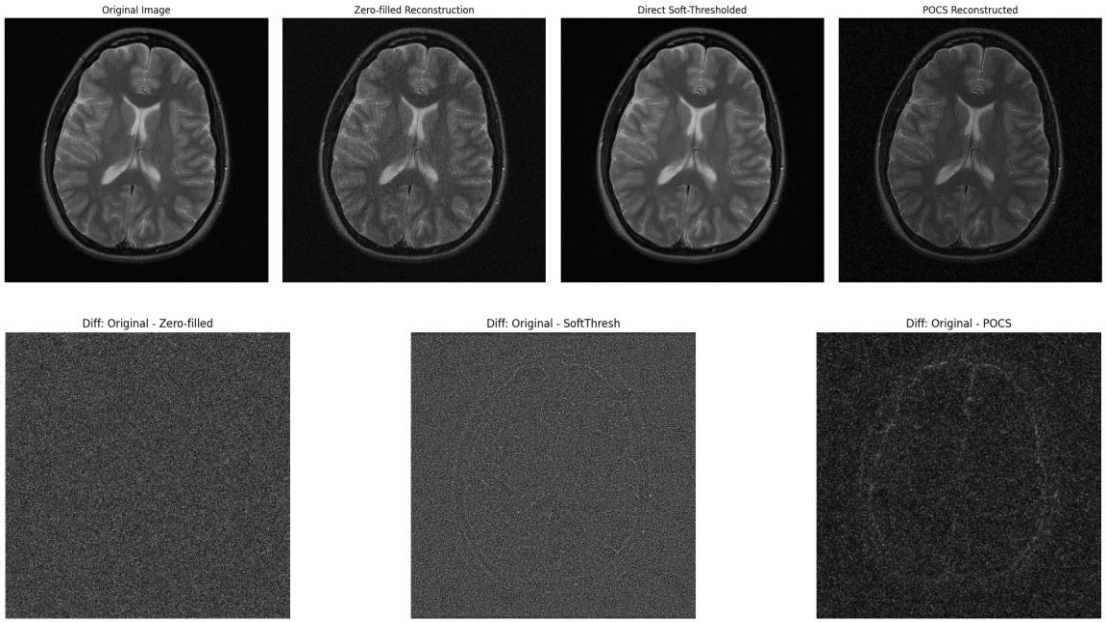
If $y = 0$ than we choose $sign(0) = 1$

$e^{j\angle y}|_{y=0} = e^0 = 1$ align with these settings.

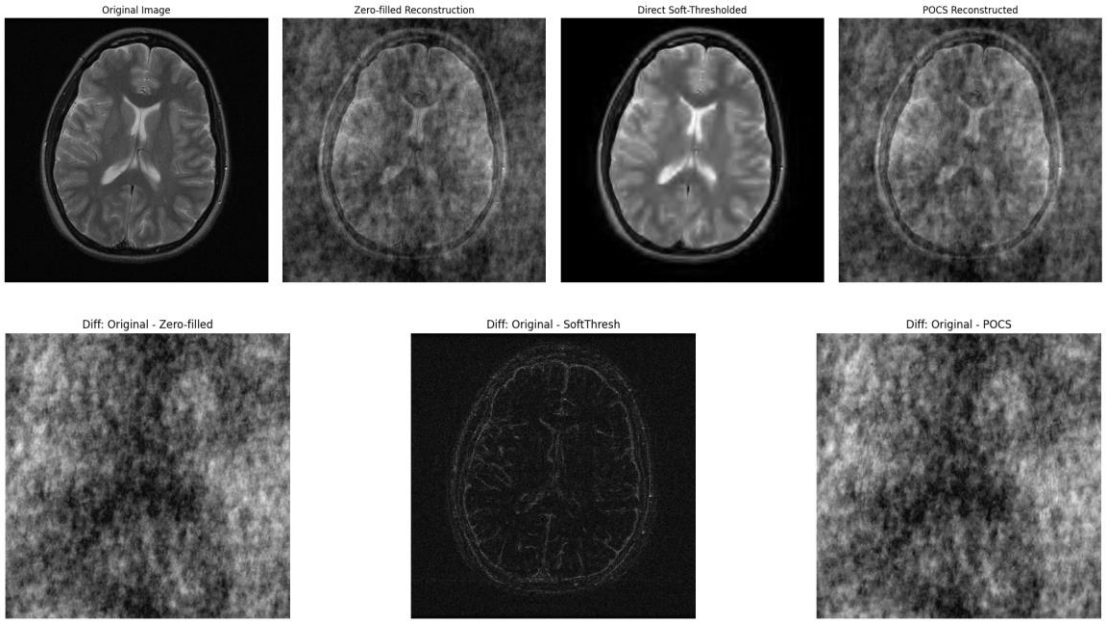
Thus, we will calculate $\hat{x}$ with:

$$\hat{x} = \underbrace{e^{j\angle y}}_{phase} \cdot \underbrace{\max\{abs(y) - t, 0)}_{magnitude\ threshold}$$

After managing to create this function, we wrote the pocs_2d function. Results:
For the variable density data:



For the uniform density data:



For variable density, the zero-filled reconstruction produces very good results, likely because the mask captures most of the low frequencies. The direct soft threshold may blur some fine details, while the POCS reconstruction looks clear and is very close to the original image. Determining whether POCS or zero-filled is better would require professional evaluation.

For uniform density, the best results (minimal “cloudy” noise) come from direct soft thresholding. This may be because uniform sampling often leads to coherent aliasing patterns that soft thresholding (in the wavelet domain) can suppress. In contrast, POCS and zero-filled reconstructions yield similar outcomes (as we understand, uniform sampling does not offer enough redundancy for the sparsity constraints to provide a significant advantage).