EOSC 454 / 556B Assignment 3: Straight Ray Tomography

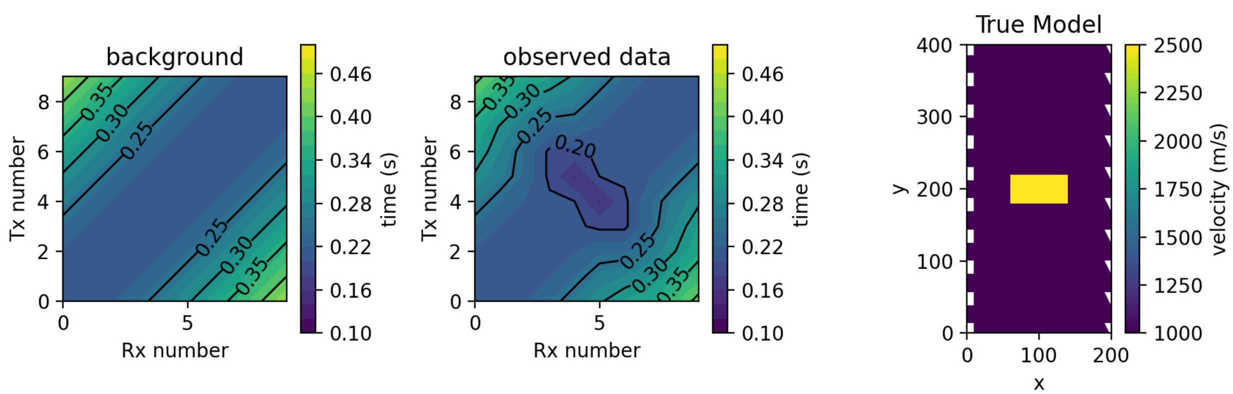
DUE:

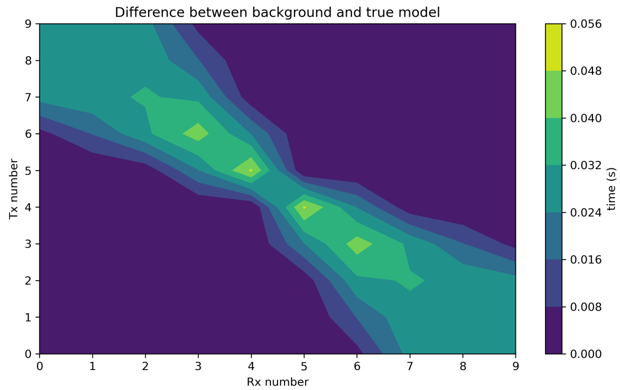
Name: Parth Pokar
Student #: 43949999

Note: The code used to generate all figures is in the Jupyter notebook attached along with this assignment and also at this Github repo: `https://github.com/pokarparth/EOSC_556_Applied_Geophysics.git`. Snippets of the code from this notebook are included here for some questions that ask for code.

Q1. In this part of the question, you will set up a forward simulation and explore how changes in the model impact the observed data. Start by setting the background velocity to 1000 m/s.

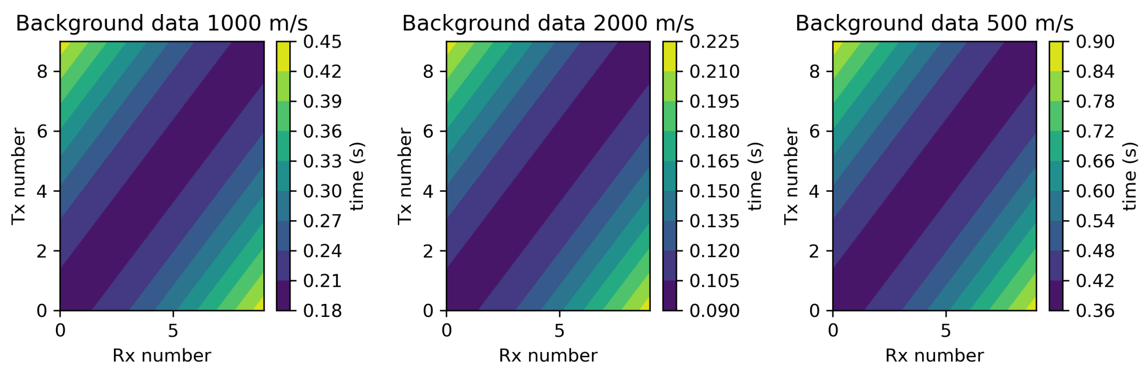
a. Add a block near the center of the survey region with a different velocity. The velocity can be smaller or larger (use a difference of at least a factor of 2). Plot the true model, the simulated data in a homogeneous backgroud, the data with the block and the difference between the data with and without the block. Don't add any noise yet.





b. First, lets convince ourselves that the forward simulation is working properly. By looking at the data for the uniform 1000 m/s background, can you demonstrate that the forward simulation is working properly? You might want to test a couple of different uniform backgrounds to convince yourself.

Given a background velocity of 1000 m/s we expect the straight-ray travel-time to be 0.2s. Similarly, for background velocity of 2000 m/s and 500 m/s we expect the travel-time to be 0.1s and 0.4s, respectively. We can test this by simulating the data for these three background velocities and plotting the data as shown below.

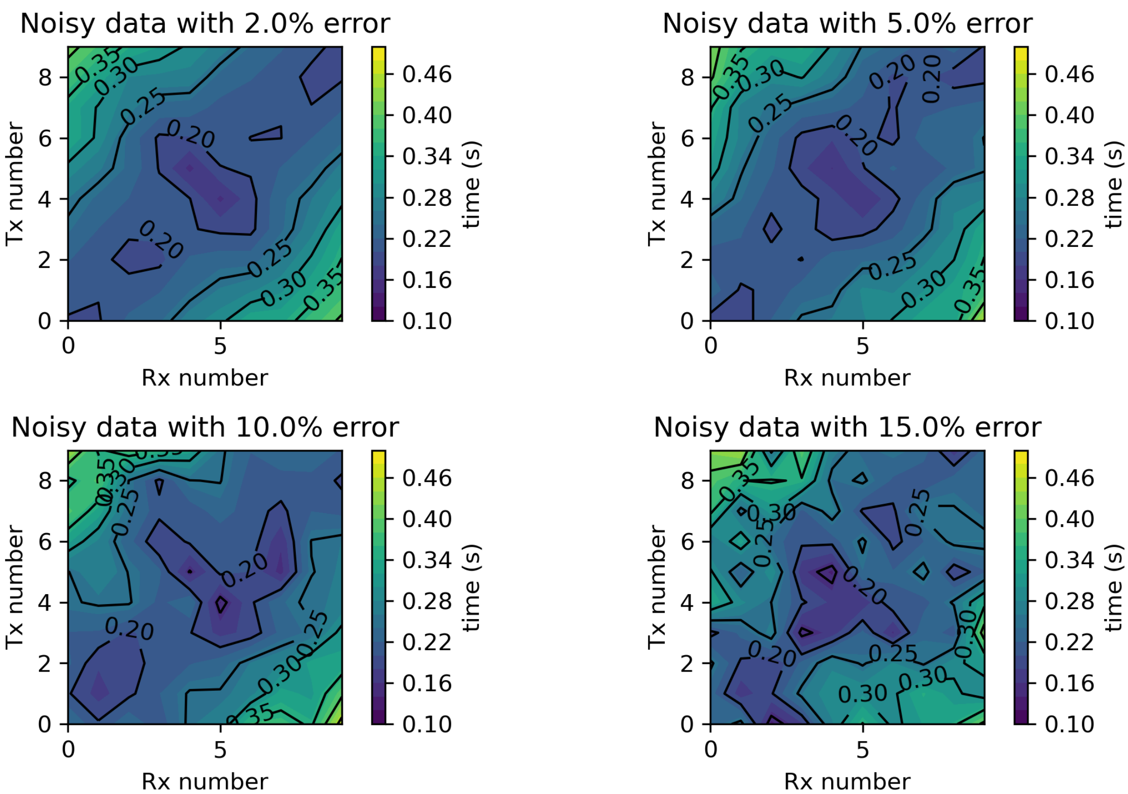


c. What is the maximum difference between the model with and without the block? What is that difference as a percentage?

The maximum difference between the true and background model is 0.0491 s or 11.95% as a percentage.

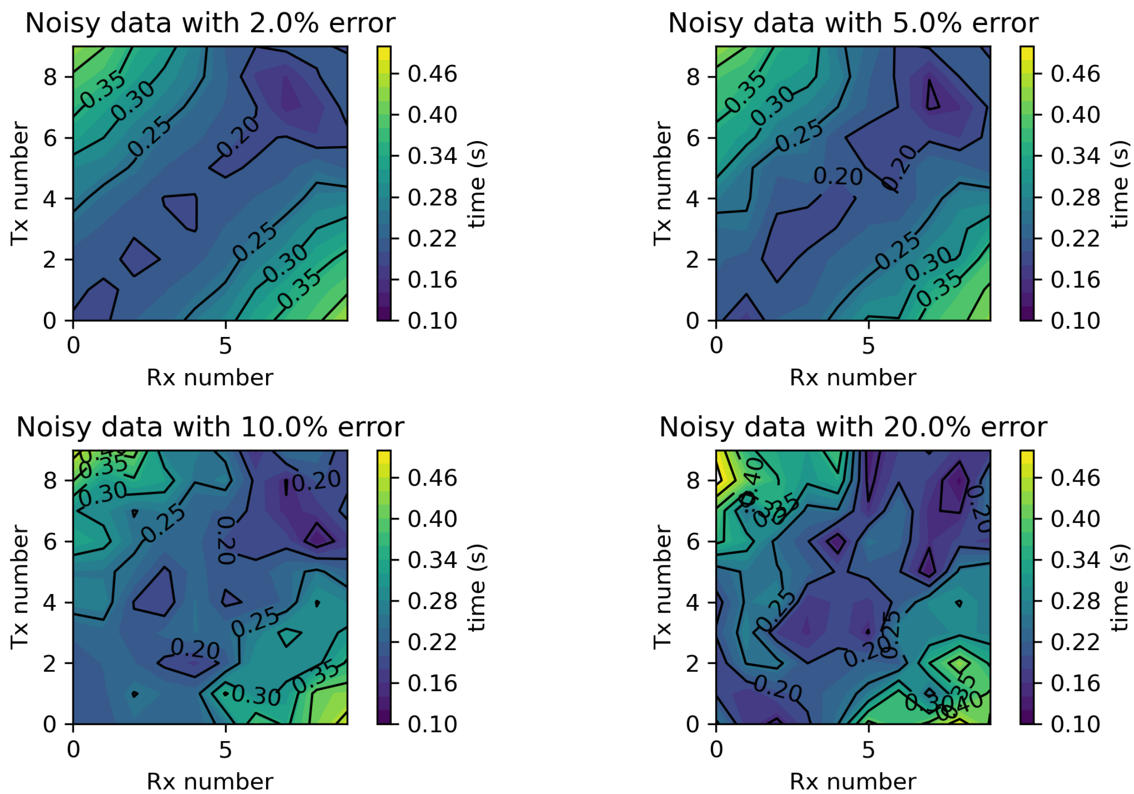
d. Now we will examine detectability of a target. Let's explore what happens when you add noise. Generate plots for a couple of choices of noise levels, where the noise is added as a percentage (e.g. `np.random.randn(len(clean_data)) * relative_error * np.abs(clean_data)`, where percentage error is `100% * relative_error`). At what noise level do you no longer have confidence that you can see the target?

I am unable to confidently detect the target (without false positive anomalies) at relative noise 10%.



e. Next, we will move the target. Move it so that it is centered within the top 100m of the survey area and again plot the data with and without the block. At what noise levels would you now expect would be the maximum at which we could expect to see the target? Is this similar or different that if the block is in the center? Why?

Since the survey is less sensitive to targets off-centre, I expect to lose sensitivity to the target at smaller noise levels if the

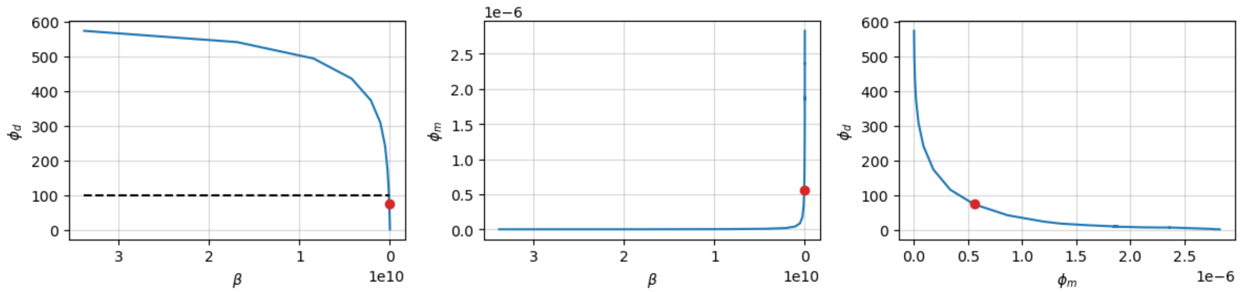


target is moved within the top 100m of the survey. As expected, the data produce false positive anomalies at 5% relative noise, leading to lower confidence in the data. This is lower than 10% relative noise when target was in the centre of the survey.

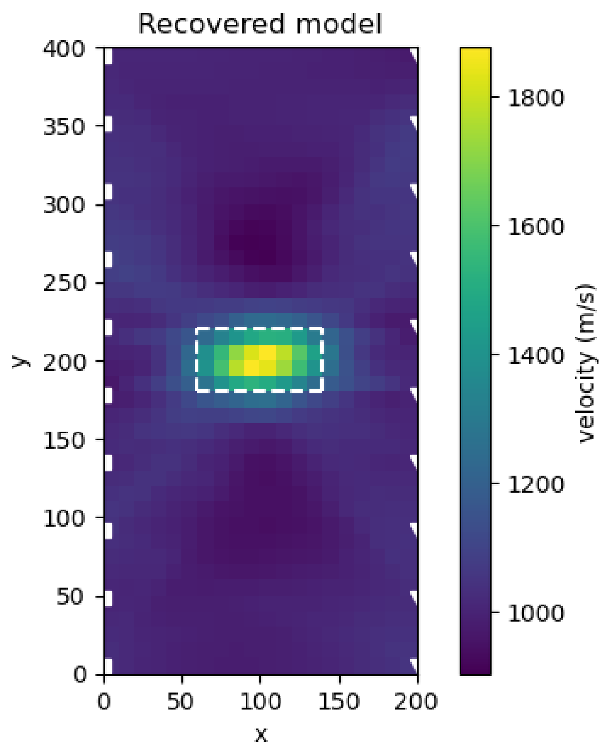
Q2. In this question, we will explore solving the inverse problem using a "smoothest model" approach. Position the block near the center of the domain and add noise at a level where you are still confident where we can detect the target in our data.

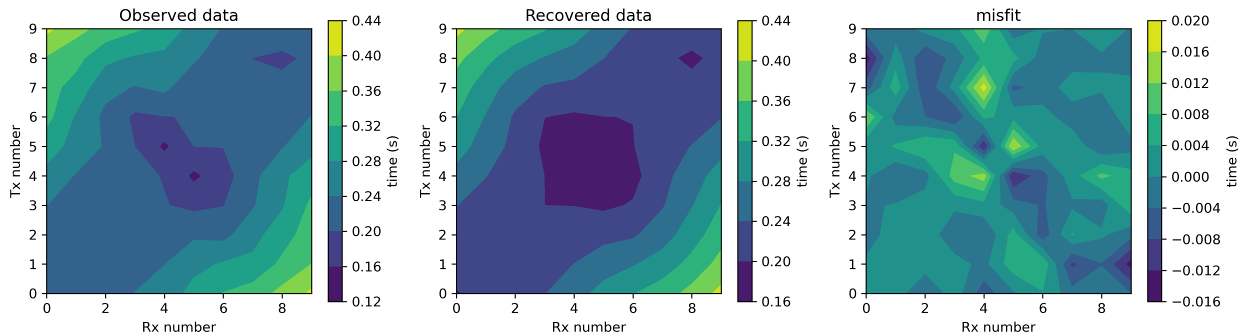
a. Start by setting up a "smoothest" model inversion. Using an accurate estimate of the uncertainties. Describe your setup. Generate plots of the Tikhonov curves. Select an iteration that adequately fits the data and show the observed and predicted data, and their difference. Finally plot the recovered model. Describe why you picked this model.

For a smoothest model inversion, I set the smallness parameter α_s to zero and I set the smoothness parameters α_x and α_z to 1 in my regularization term.



The best recovered model is below.

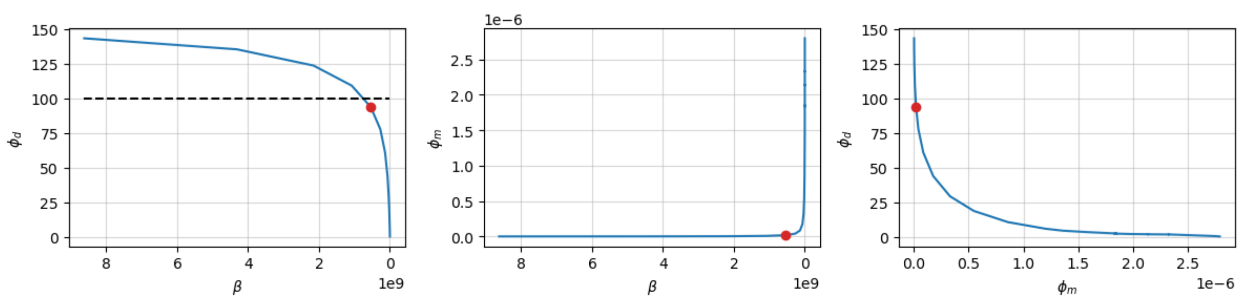




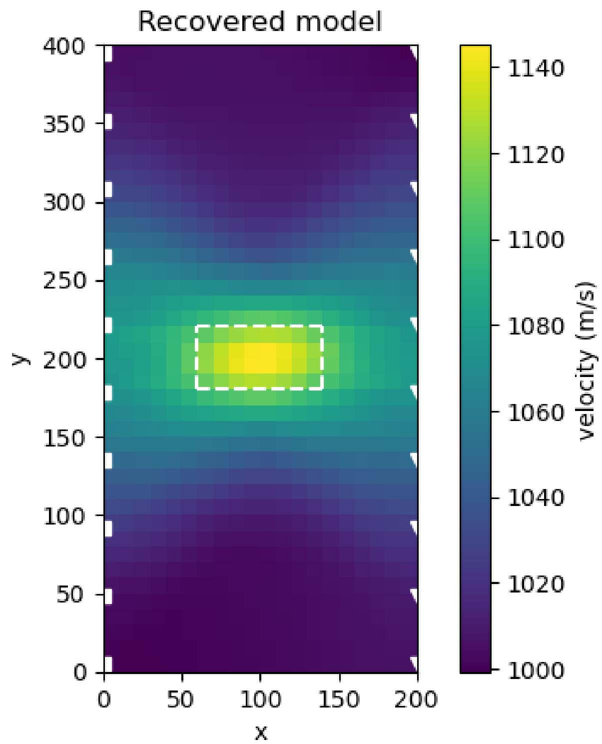
I chose this model as this iteration corresponded to the "elbow" of the Tikhonov curve; the data misfit was lower than the number of data; and model misfit did not include too much structure; and the difference between observed and predicted data was close to the assigned uncertainties.

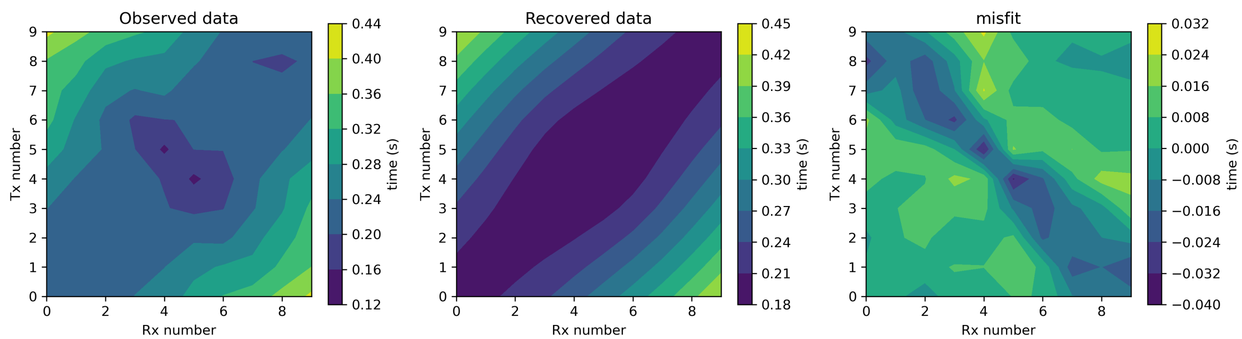
b. Now, lets explore that happens if an incorrect noise level is used. When creating the `data` object that is used in the inverse, lets set the `relative_error` to be too large. Choose `relative_error=2*true_elative_error`. Plot the Tikhonov curves. Select an interation that is close to the data misfit. What features in the Tikhonov curves, data, and model suggest that we are under-fitting the data? Based on this, why might you want to run an inversion past a χ-factor of 1? The

Tikhonov curves for an error (0.04%) twice the true relative error (0.02%):



Following is the recovered model and data for an iteration with data misfit close to the target misfit:



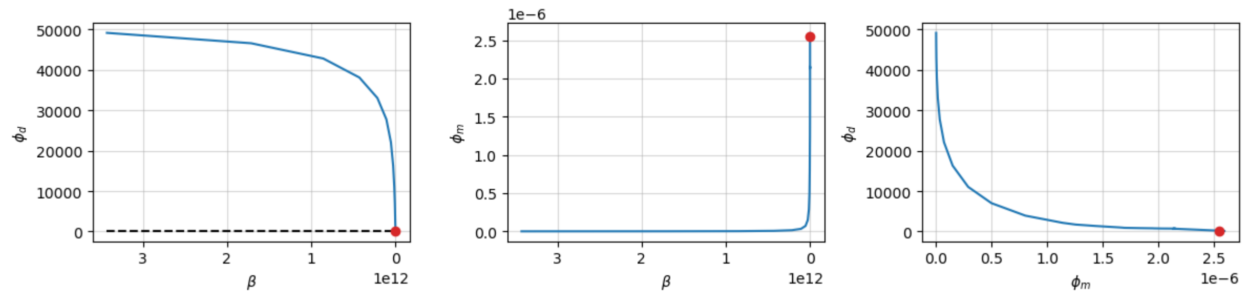


For this iteration, the Tikhonov curve shows that while data misfit is below the target misfit, the model misfit is close to zero as well. So little structure is expected in the recovered model. The recovered model and data plots show that the data is being underfit. The recovered data and model is smoothed out. The velocity in the recovered model is lot lower than the true velocity.

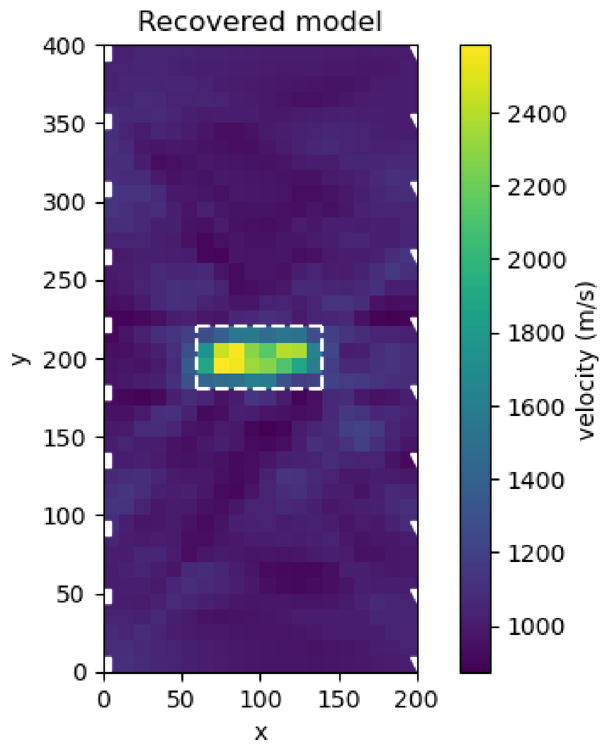
Since we likely want the more structure to be incorporated, for cases when we do not accurately know the uncertainties, choosing a model below the target misfit is prudent. This can be achieved by running the inversion past a χ-factor of 1.

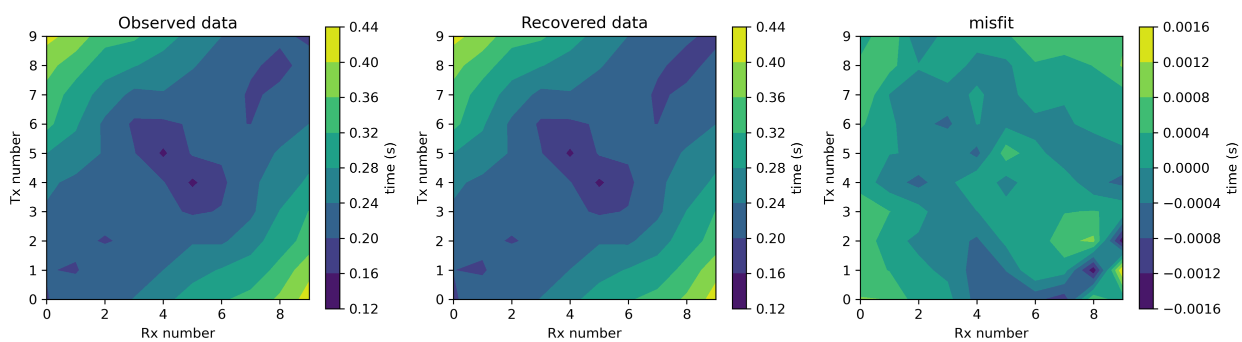
c. Now, lets explore that happens if an incorrect noise level is used. When creating the `data` object that is used in the inverse, lets set the `relative_error` to be too small. Choose `relative_error=0.1*true_elative_error`. Plot the Tikhonov curves. Select an iteration that is either close to the data misfit or where the data misfit levels off (you can try increasing the `maxIter` to increase the maximum number of iterations that the inversion takes). What features in the Tikhonov curves, data, and model suggest that we are over-fitting the data? Look back to some earlier iterations. When does it start to become obvious that you are over-fitting the data?

When using a noise level that is too small, the inversion never hits the target misfit. The Tikhonov curves are below:



Following is the recovered model and data for an iteration with data misfit close to the target misfit:

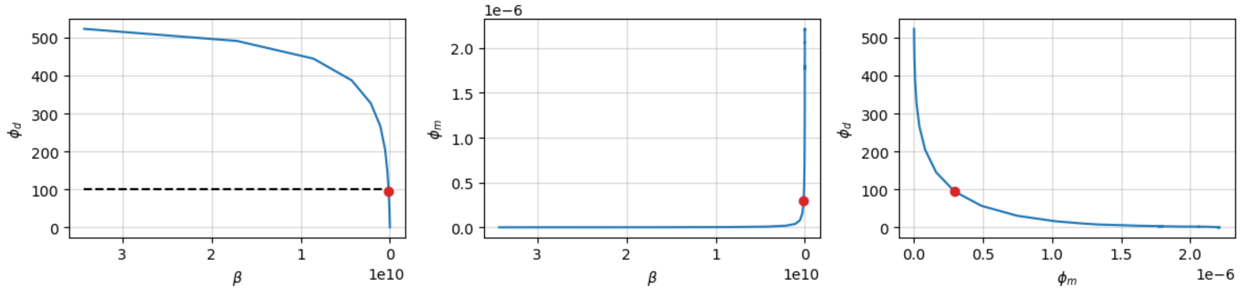




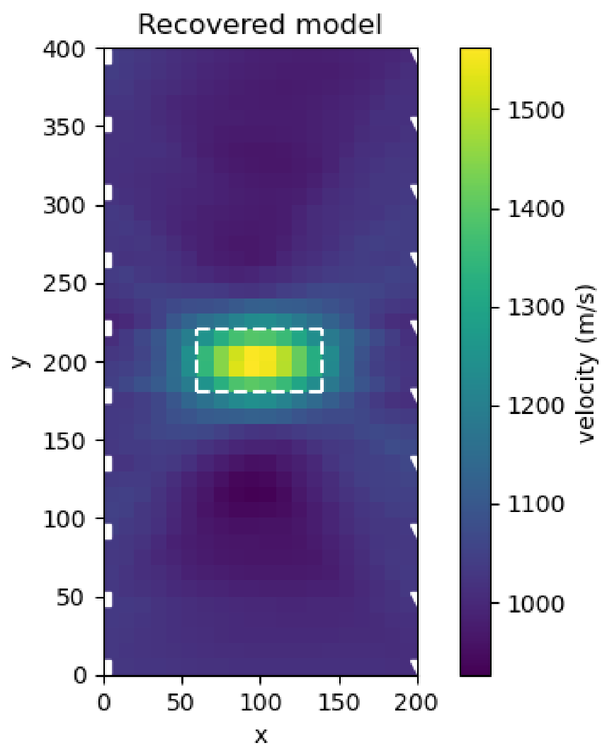
Overfitting of the data is most obvious in the last plot where the predicted and observed data is very similar and the difference between the two is much lower than the true relative errors. The recovered model also show high amount of structure which can be indicative of overfitting.

Q3. In this question, we will explore solving the inverse problem that balances the α-values between the smallest and smoothest regularization terms. Use the same position of the block and noise levels in the data as before.

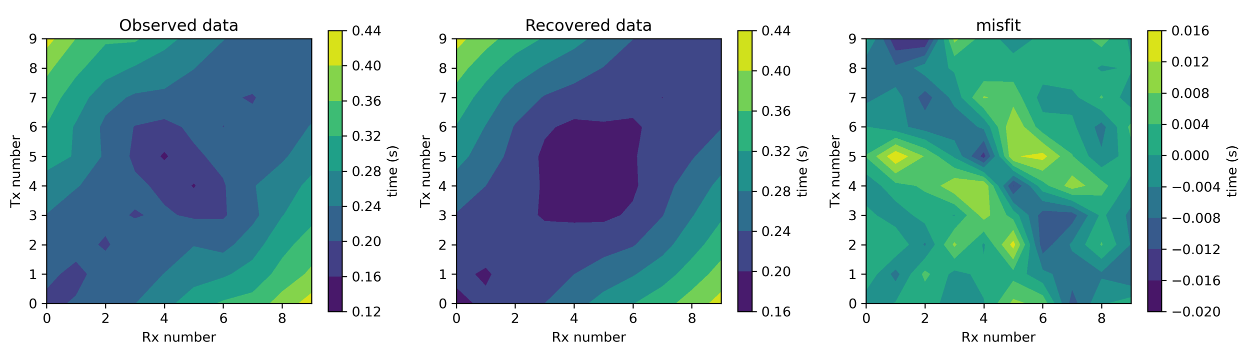
a. First, perform an inversion using the correct assignment of uncertainties and show the Tikhonov curves, observed and predicted data and recovered model.



ϕ_d
β
1e10
ϕ_m
1e−6



Recovered model
y
x
velocity (m/s)



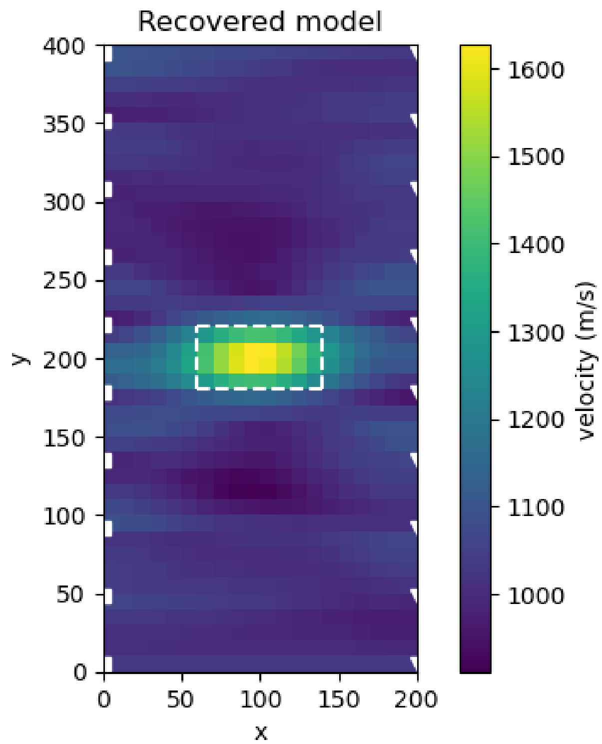
Observed data
Recovered data
misfit
Tx number
Rx number
time (s)

b. Repeat the exercise of over and underestimating the noise when creating the data object that is fed into the inversion. Describe how the results are similar / different than in the previous question.
In this case, the Tikhonov curves, observed and predicted data plots for both underfitting and overfitting case is similar to what was observed for the smoothest model inversion. However, the recovered models are different. For the balanced inversion case here, smearing or smoothing of features in both X and Z direction are present. This smoothing is more strong in the X direction than Z, which reflects the sensitivity of the survey.
Plots for these cases are not included here but are present in the jupyter notebook attached to this report.

c. When first encountering a new data set, would you recommend starting with balanced α-values, or a smoothest model approach? What is your reasoning?
When encountering a new dataset, I would start with smoothest model approach to not bias my results and encourage inversion to minimize smallness in any direction without having a-priori knowledge. As such, the results from the smoothest model approach are helpful to check whether any unexpected result is from smallness regularization or incorrectly assigned uncertainties.

d. Now set the noise levels back to the appropriate choice. Try substantially increasing α_x while keeping $\alpha_y = 1$. Plot the resultant model that fits the data and describe why you see the features you see in the recovered model.

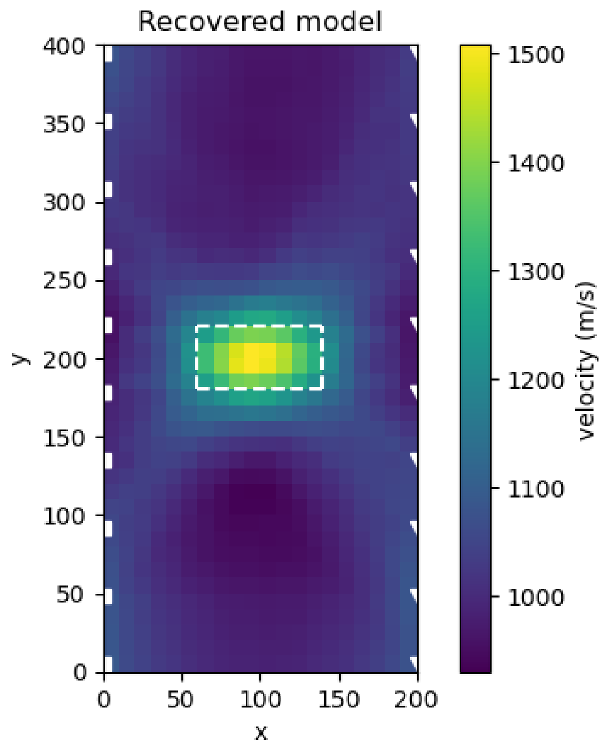
Following is the best fitting model when $\alpha_x = 10^3$ and $\alpha_y = 1$:



The model reflects the fact that we encouraged the regularization to minimize changes in the X direction and as such a strong smoothing of the target in this direction is observed.

e. Now set the noise levels back to the appropriate choice. Try substantially increasing α_y while keeping $\alpha_x = 1$. Plot the resultant model that fits the data and describe why you see the features you see in the recovered model.

Following is the best fitting model when $\alpha_y = 10^3$ and $\alpha_x = 1$:

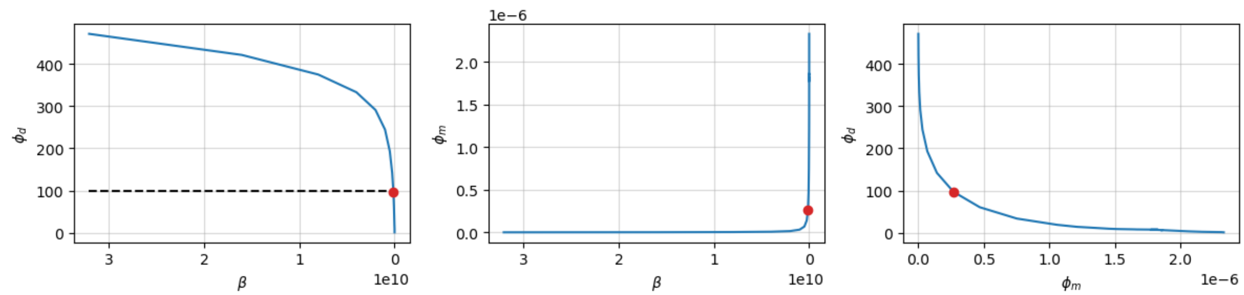


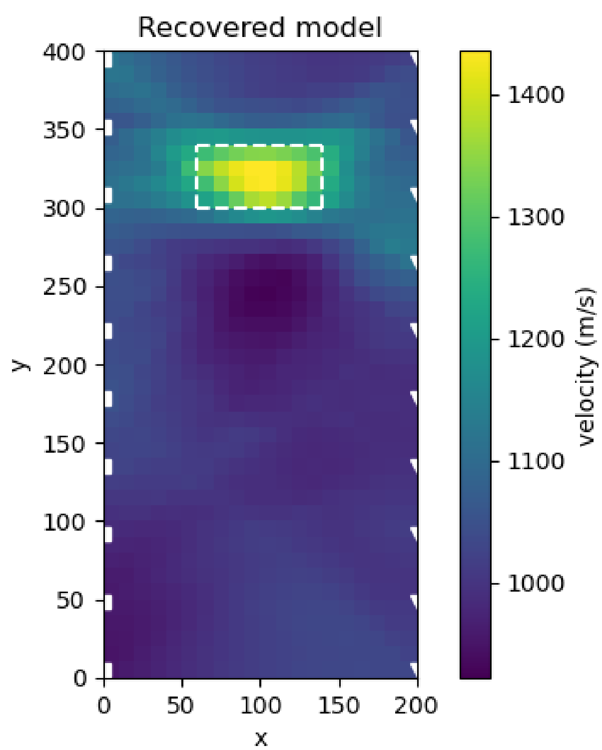
Here, the model reflects the fact that we encouraged the regularization to minimize changes in the vertical Z direction instead. Smoothing in the Z direction is observed in this case but it is not as strong as that observed when $\alpha_x >> \alpha_y$. This reflects the lower sensitivity of the survey in the Z direction than in X direction.

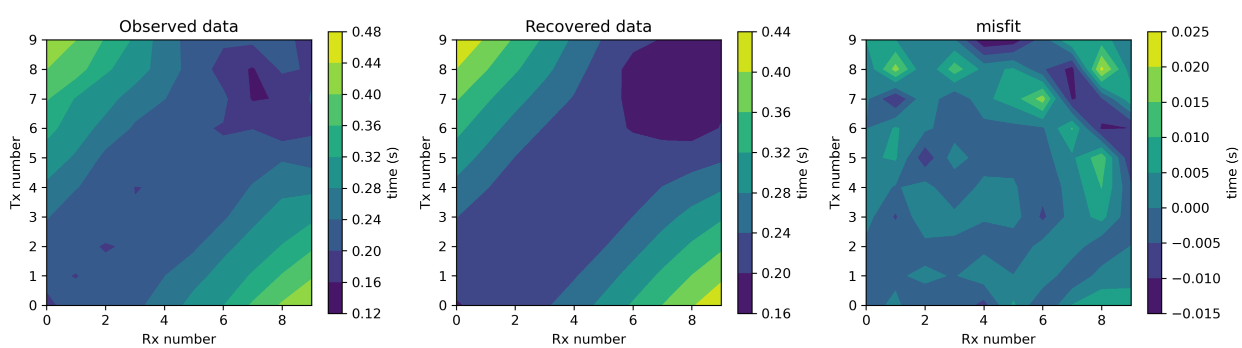
Q4. Now, lets move the block. Position it so that it is within the top 100m of our survey area. Keep the rest of the model parameters the same.

a. Run a smoothest model inversion using the appropriate assignment of uncertainties. Plot the Tikhonov curves, predicted and observed data and recovered model that fits the data. Describe what is similar and different to the result obtained when the target is in the center of the target region? Why might this be the case?

The plots are below:



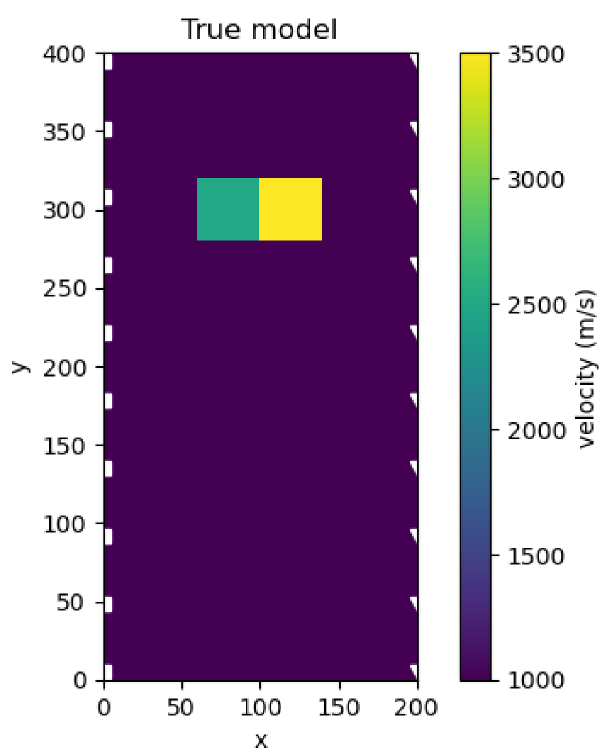


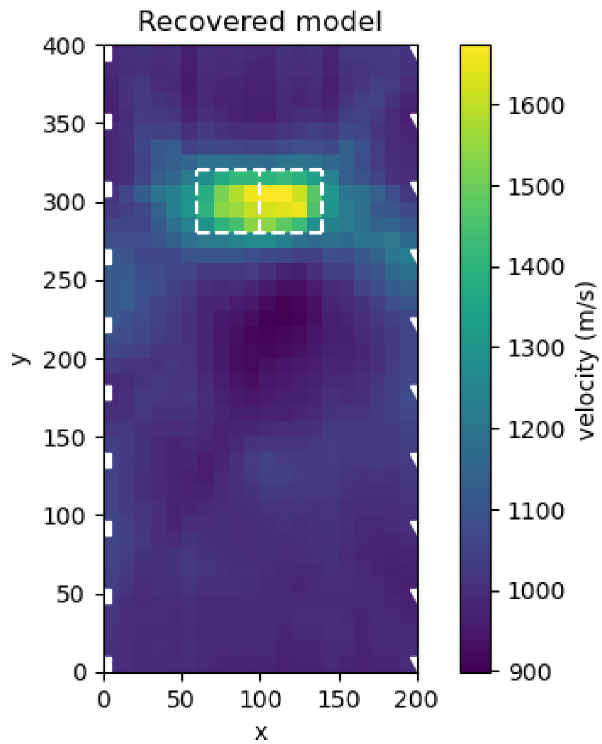


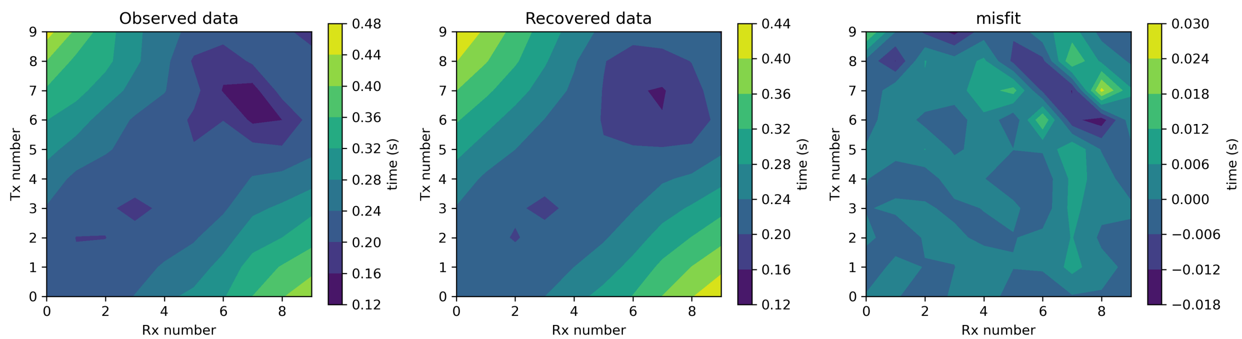
Similar to the smoothest model inversion earlier, the target location is recovered in the correct position. However, the recovered model is more smoothed out in this case than

earlier. The recovered velocity in the core region is lot lower than the true target and the background velocity is greater than the true background as well.
In addition, the smoothing is primarily along X-direction and diagonal direction. This again reflects the fact that the survey has lower sensitivity in this region than in the centre of the survey.

Q5. Time to play! Pick a concept that we discussed in class (e.g. influence of the mesh, what the influence of the reference model is, or how the survey design influences the results obtained). Describe the concept that you will explore, your setup and generate some plots that are illustrative of the concept you are demonstrating.
I decided to test the sensitivity of the survey in case where the true model is modified by shifting it to the top in addition to setting half of the target block to have an even higher velocity. The true and recovered model along with the data plots are shown below.







I observe the smoothing and underestimation of true velocity as seen in Q4 when I shifted the block to top where the survey has lower sensitivity. I ran an inversion with $\alpha_y >> \alpha_x$ to minimize smoothing in the X direction. While this helped the recovered model, without knowing the true model I would not be confident in interpreting the recovered model as consisting of two different velocity blocks.