GUI testing on test model (three block)

To start exploring the interface of our newly developed package, let's look at some of the tools we have included. One of these tools is the Acquisition Geometry feature as in Fig. 1, which opens a window for users to input relevant information via the keyboard and generate geometry files. Additionally, the tool includes a plotting feature that enables users to visualize the geometry.

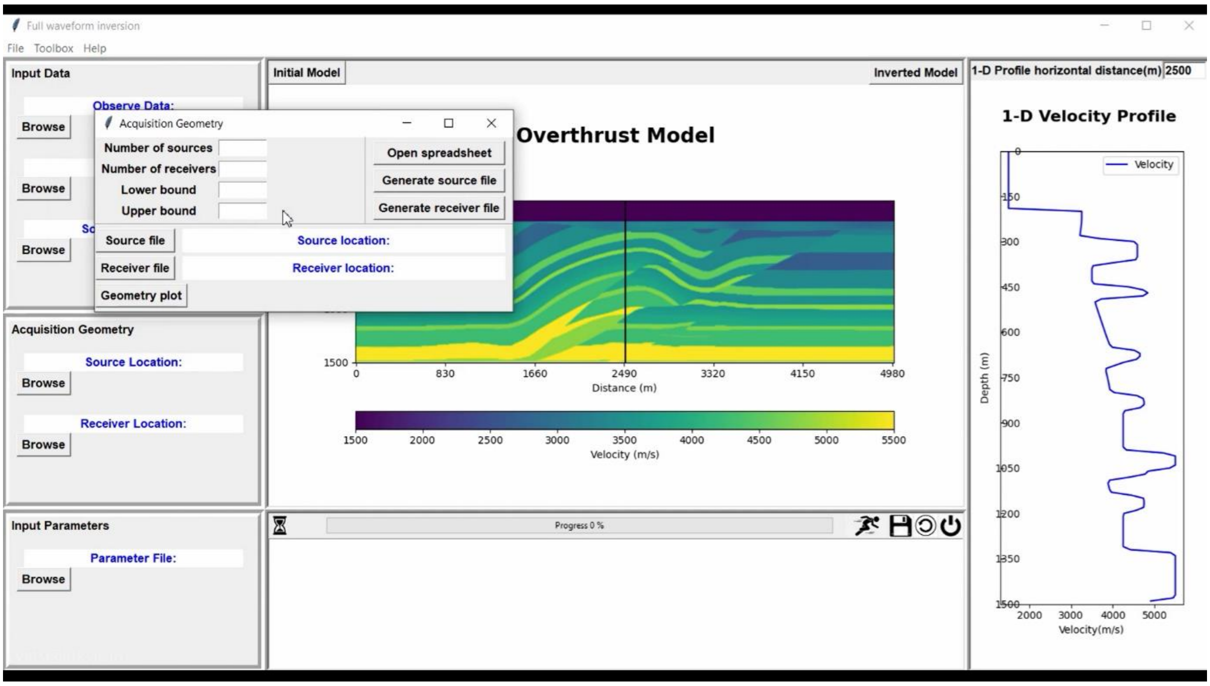

Figure 1: Acquisition geometry.

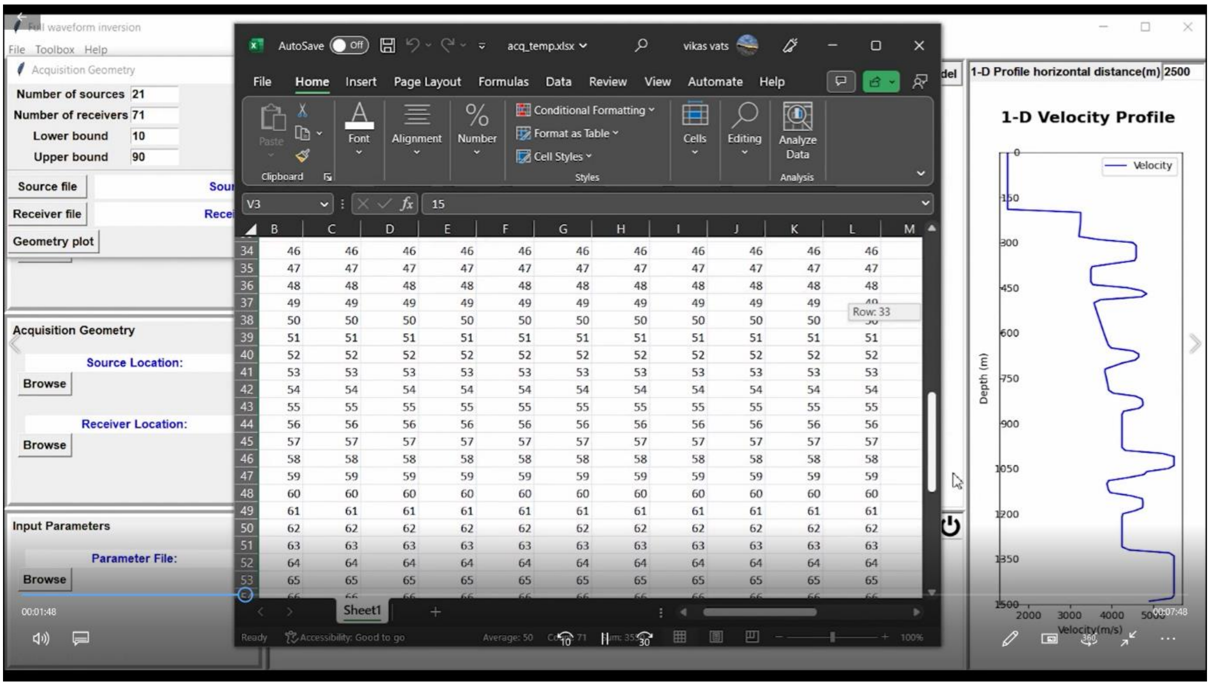

Figure 2: Geometry file

To generate accurate geometry files Fig. 2, we require users to provide the number of sources and receivers along with their positions, which should match the observed data. Additionally, users need to input lower and upper bounds, which depend on the number of grids reserved for the absorbing boundary conditions at the edges of model. For instance, if we reserve 10 grid points for absorption, none of the receivers or sources should lie within these grids. Hence, our acquisition setup must be within the 10 to 90 grids range for this model.

The observed data for test model contains 32 sources and 76 receivers, with sources placed every 2nd grid points from the 18th to 80th grid, and receivers at every grid point from the 13th to 88th grid. We have reserved 10 grid points at the boundary for absorbing layers, and if any receiver or source falls within this range, the package displays a warning. Once the user inputs the configuration, the package generates a spread based on the receiver locations along the source.

This spreadsheet as shown in Fig. 2 has been created using the specified configuration, with the first row indicating the source locations and the subsequent columns indicating the corresponding receiver locations for each source. I have already filled in the required source and receiver locations. Once these locations are inputted, you can save the file and use the Generate buttons to create the source and location files on your local system.

You can also plot the geometry by selecting the generated and source and receiver files, which I have already chosen. Simply click on the designated button and a plot will be generated with the receivers and sources marked. The y axis represents the number of shots, while the x axis represents the receivers. The green triangles indicate receiver locations, and the red stars indicate the source locations. This plot is an excellent tool for visualizing the data acquisition setup.

A help section has been included in the menu bar. By clicking on it, users can access tutorial videos and helpful documentation via a redirection link of GitHub repository.

To demonstrate the efficacy of our FWI package, we will utilize it to invert a simple model with a depth of 500 meters and a lateral extent of 1000 meters. We possess the observed data for this model, which includes three low-velocity blocks located at different depths, with a constant background velocity of 2000 meters per second. These blocks exhibit a velocity of 1800 meters per second.

For this scenario, we will begin with a starting model characterized by a constant velocity of 2000 meters per second.

To enable numerical modelling, we have discretized the model into square grids measuring 10 cross 10 meters. This discretization results in a total of 50 depth grids and 100 lateral grid points. It is crucial to note that all distances are measured in terms of these grids for simplicity. Moreover, we have designated specific grids at the model's boundary to enforce absorbing boundary conditions. This technique effectively suppresses artificial reflections arising from the model's edges.

The parameter file is being generated by selecting parameters from the toolbox as shown in Fig. 3 & 4. which are saved for future use. Number of samples, representing the number of samples recorded, here total 700 samples are recorded. Sampling interval is the time duration between two consecutive samples, which is 0.001 second over here. Dimension of model is measured in grid points which is 50 along the depth and 100 laterally. Grid size referring to the spacing between two grid node which is 10 m in this case. Although we have placed the receiver and source at the surface, we are utilizing the 6th order FD method, which reserves three grid points at the surface. This implies that for the surface, the depth of both the source and receiver needs to be set to three grid points. Number of sources and receivers are 32 and 76 respectively.

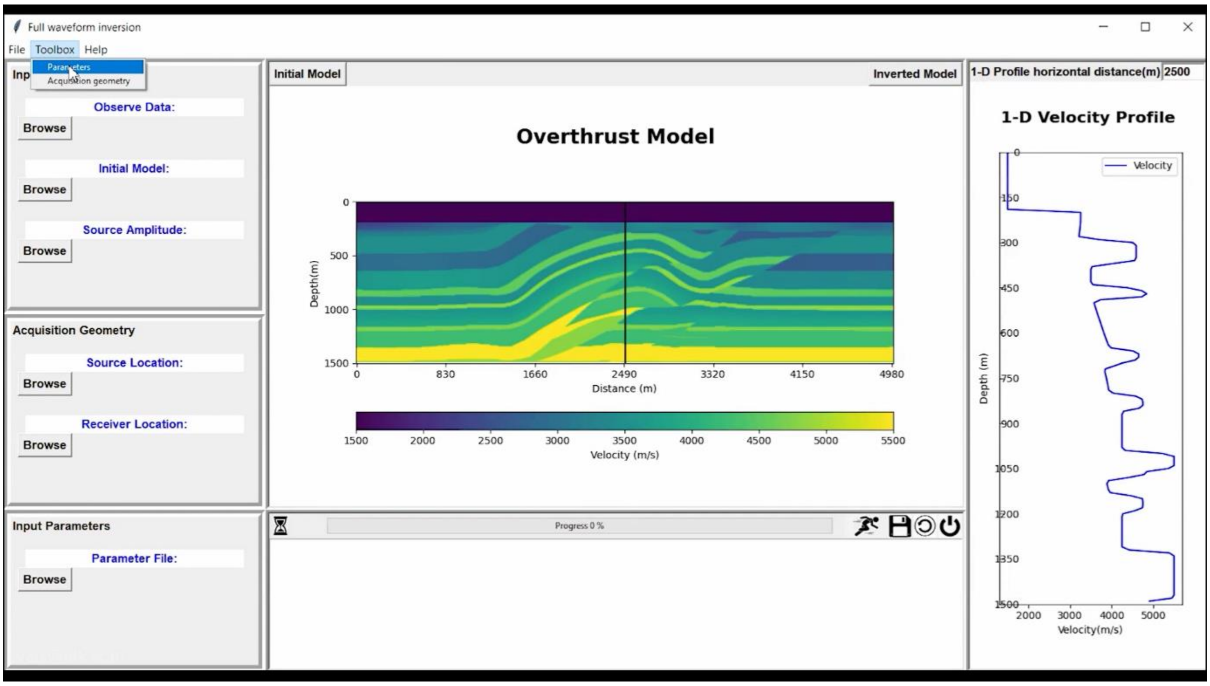

Figure 3: Toolbox

Figure 4: Parameter

We have determined that the number of processors to be used will be based on availability, and it must be a factor of the number of shots. In this instance, we have chosen to use 4 processors, which is also a factor of 32 (the number of shots). Number of iterations is determined by the user, and typically, accuracy increases with an increasing number of iterations. In this case, we have set the number of iterations to 10. The boundary key determines whether boundary conditions will be applied or not. A value of 1 indicates that the boundary conditions are turned on, while a value of 0 indicates that they are turned off. In this case, we have reserved 10 grid points as a layer for absorbing boundaries. This means that the absorbing layer will consist of 10 grid points.

The choice of objective function is determined by selecting either 1 or 0. This selection determines the type of function used to calculate the residual. This package also offers flexibility in muting the gradient calculation for specific areas of the model at the edges. This can be achieved by muting the layer, and it is recommended to mute the absorbing boundary layers when calculating the gradient. An option is also available for selecting the data order, whether in row major or column major format. A value of 1 typically represents row major format.

The user can save these parameters in a file by clicking on save button.

Users can load existing parameter files to edit the parameters. Additionally, a feature is included in this window which provides all necessary information about a parameter by hovering the cursor over it.

Once all input files are ready, the inversion process can be started using the interface shown in Fig. 5 of our developed package. Inputs can be loaded through this window, and several other important buttons have been added to control the flow of the inversion process.

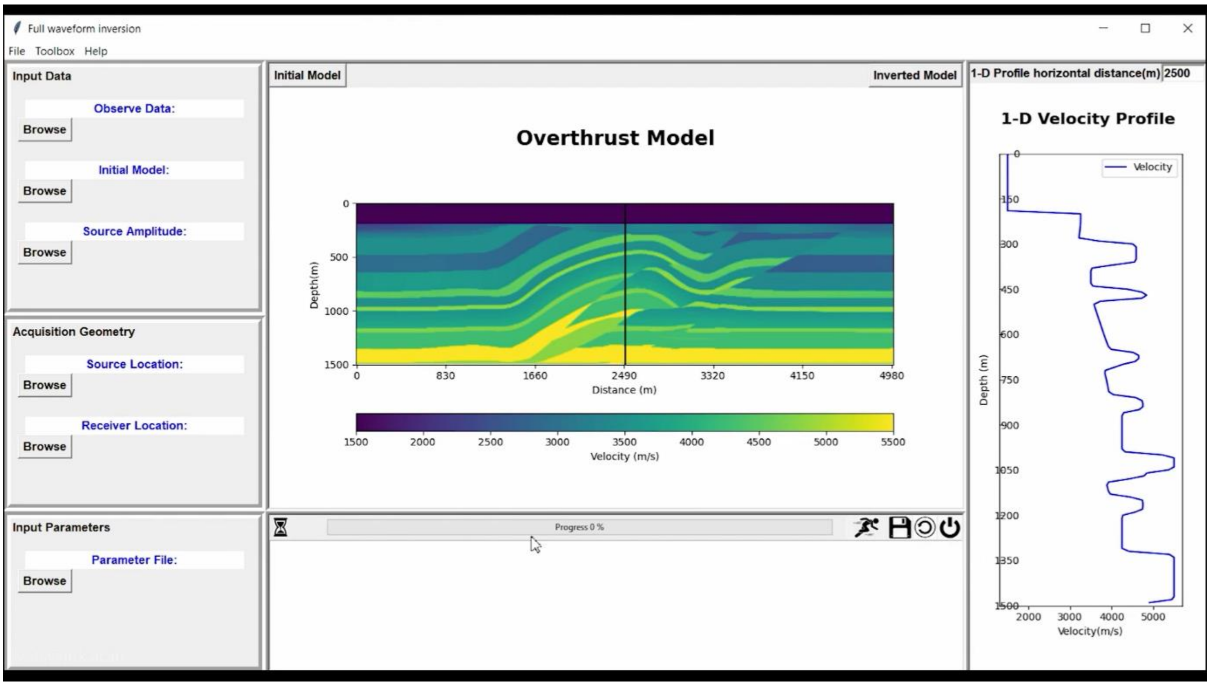

Figure 5: Window

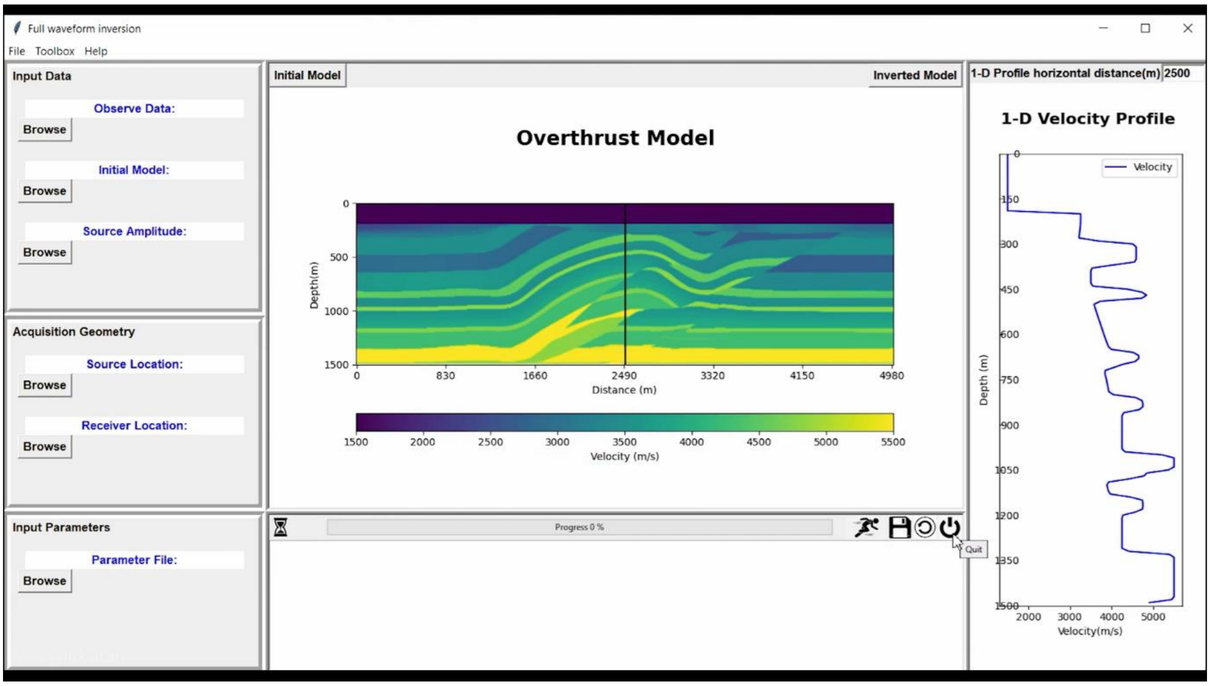

Figure 6: Quit button

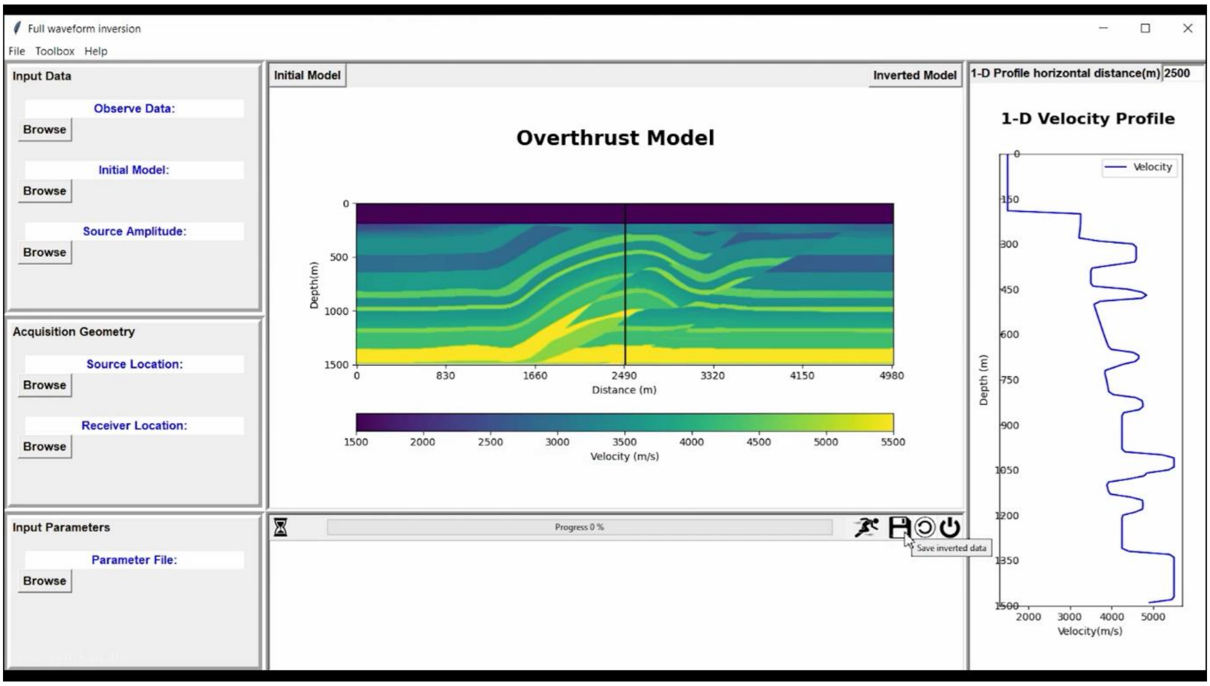

Figure 7: Save button

These buttons serve different functions: such as browsing the input data files, plotting of input and the recovered model, starting the inversion procedure. If the user attempts to click buttons without loading the input files, an error message will be displayed. The output can be saved, and the application can be quit during the process as shown in Fig. 6. The progress bar updates and a window at bottom of interface displays the receiver and source position, elapsed time and cost function after every iteration, allowing for easy monitoring of the process. Live plotting of the inverted models is also possible. This interface includes a slider that selects a 1D profile from the 2D structure sown in Fig. 8, allowing for comparison between the inverted and initial models. The value of the slider can be set with the input field.

The required files can be loaded by browsing through the system using the buttons in the interface window.

To recover the real model, the user needs to initiate the inversion process by clicking on run button.

The inversion process is initiated by clicking on the Run button, which updates the status of the cost function and elapsed time in this window. The updated model is also live plotted in this section. The 1D velocity profile corresponding to the slider is plotted in this section, where the red line represents the initial model, and the blue line represents the recovered model. This feature is useful for analysing the inversion of a specific model.

Once the inversion process is completed, the resulting model can be saved.

The package's effectiveness in inverting the three-block model were evaluated by comparing the recovered model with the actual model. The inversion process began with a uniform model, but it successfully reconstructed the three low velocity blocks present within the actual model. This indicates that the package is efficient and reliable. The final results are shown in Fig. 9.

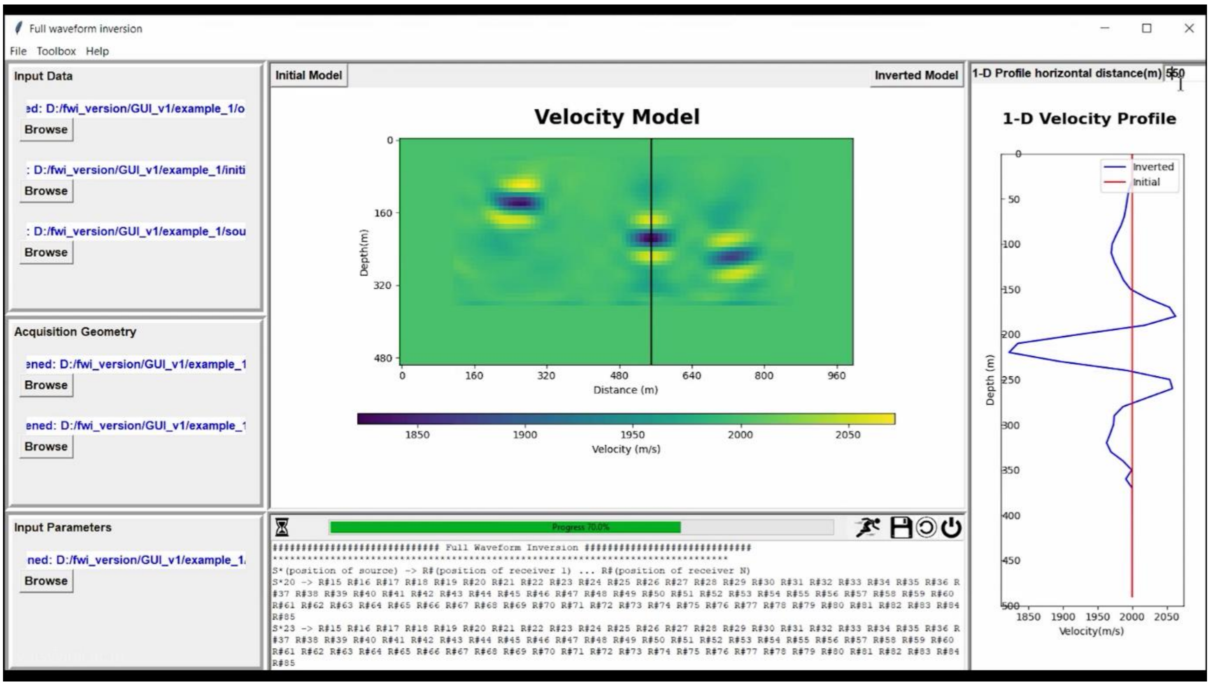

Figure 8: Inversion

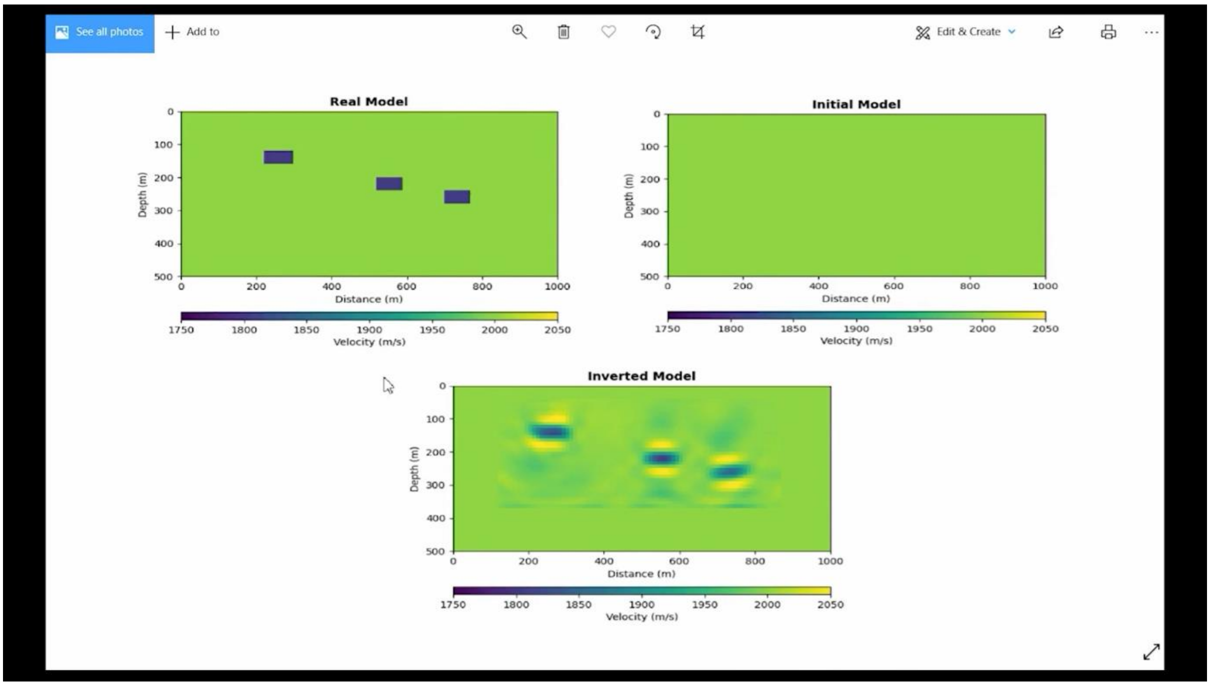

Figure 9: Results