Efficient 3D-migration of Raptor data

In the previous note we covered the processing steps up to post-processing, jumping the migration procedure. In this note we'll look specifically at migration and how modern and intuitive software makes this a straightforward process.

3D-migration

Migration is the process by which hyperbolic anomalies are collapsed into points using a known velocity and a mathematical algorithm. Nowadays, it's not used extensively in 2D-profiling, since operators learn quickly to recognise the hyperbolic shapes commonly formed by utility lines and other targets.

In our view, the value of migration is most pronounced in top views (C-scans) of data, where linear targets and edges spread out laterally in un-migrated data. The process also has the potential to clean up the asymptotes from the sides of ditches or other buried targets, which are not necessarily linear. It's worth noting that in the past it was quite common to migrate 2D profiles and then apply interpolation to the migrated sections, but this is not what we mean with true 3D processing.

To make a stringent migration, one must know the wave velocity throughout the whole surveyed site. In 2D-data, it's possible to apply a layered velocity model before migration, while in a large 3D-data set, this becomes very difficult, if not impossible. Add to this that migration is a quite time-consuming process and that we know of no software able to handle variable velocities over large areas effectively. So, we should have some tools and strategies to make this process smooth and effective.

It's common to apply hyperbola fitting on 2D sections to estimate local velocity. While this may work well in many cases, we propose a more robust method, via test migration of selected 2D sections.

Modern and interactive software will allow the user to select and visualise any 2D cut from the top view. As illustrated in
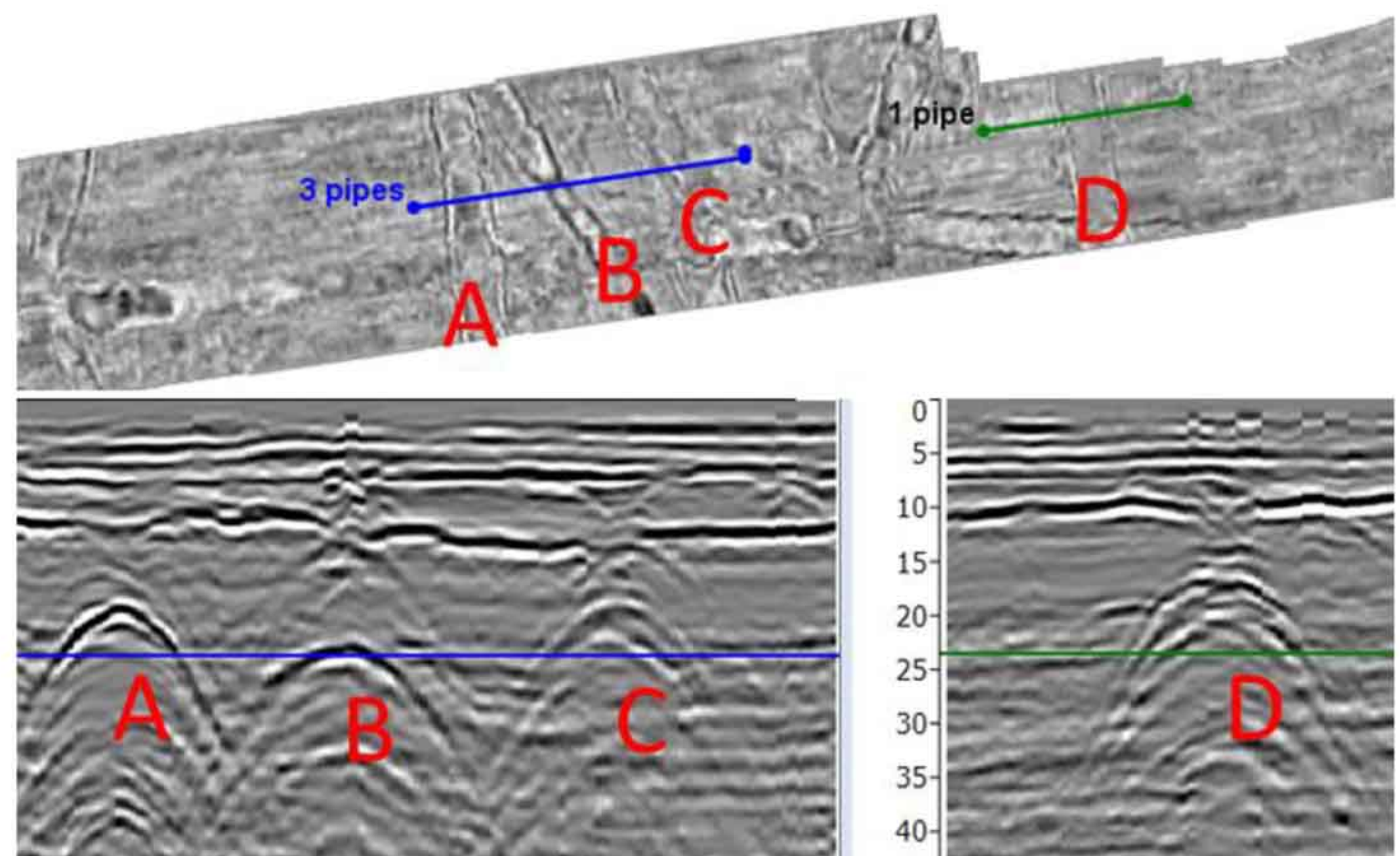

Figure 1, Top: C-scan with marked 2D cuts across four linear targets. Bottom 2D-views of the cuts marked in C-scan. Horizontal lines in the 2D views shows the depth/time of the C-scan

Figure 1, we see two lines crossing four linear targets and the corresponding radargrams showing the hyperbolas. While the first three targets may easily be estimated with hyperbolas, although a keen eye will note the variations in the diameters, the fourth does not look that clean at all. It may be the roof of a culvert rather than pipe.

Method

Since 2D migration is fast enough to make interactive tools possible, we provide a slider for varying the velocities in the radargrams shown in Figure 1. Then we fine-tune the velocity to find the setting

that compresses the anomalies the most. We don't think about this as finding the true velocity in a stringent way, but rather to compress the hyperbolas the most. Figure 2 shows a result when focusing on

Figure 2, Test migration of the two radargrams from figure 1, left migrated with 75 m/µs and right with 84 m/µs

targets A and D, where the resulting velocities are 75 m/µs and 84 m/µs, respectively. So, which one to use? A variation of almost 10 m/µs is quite significant, and a velocity of 75 m/µs is considered low,

in a case like this. Now, it's easy to find out by using the slider to adjust the velocity up to 84 m/ µs, which gives the result shown in Figure 3. Now target A shows the typical 'smiles' characteristic of too high a velocity, while the other targets compress better.

It's possible to delve into a lot of detail concerning the correct velocity adjustment. For example, the profile doesn't cross each target perpendicularly. However, there's often a compromise to be made, which in reality is not crucial, as long as migration velocity is treated separately to the velocity used for depth calculations and awareness of the compromise is kept.

Figure 3, first radargram migrated with velocity set to 84 m/µs shows better compression of target A and B, but starts to look over-migrated

Figure 4 shows the result of 3D-migration with velocity set to 80 m/µs and the targets compress nicely, regardless of their directions. The latter is the strength of true-3D migration based on proper channel spacing, good positioning, and practical software able to stitch/bin the data into a 3D volume.

Figure 4, result of 3D-migration with velocity set to 80 m/µs

Takeaway

In 3D-applications, the migration procedure shines when it comes to visualising targets in the top-view because when working with interpretation, it's the view most utilised. There's no deep expertise needed for applying it correctly when modern, interactive software makes the process swift and intuitive. In this example, we did not show the full strength of it, and that will be more obvious when we come to AVI-exports and deep-slice proces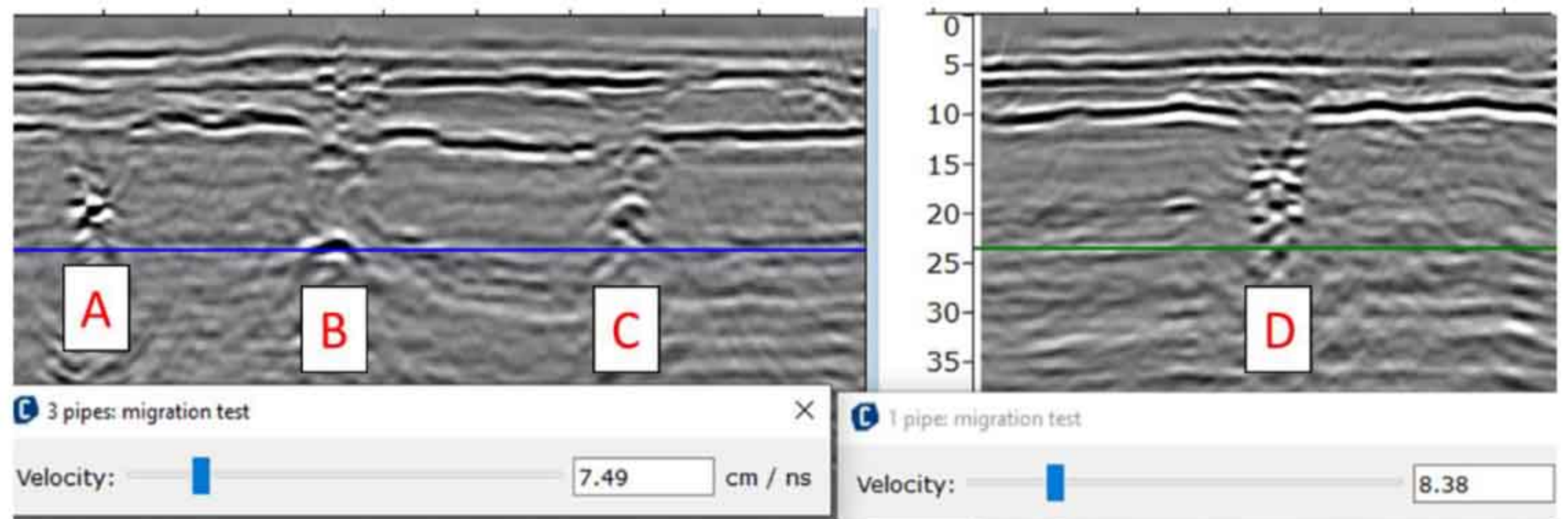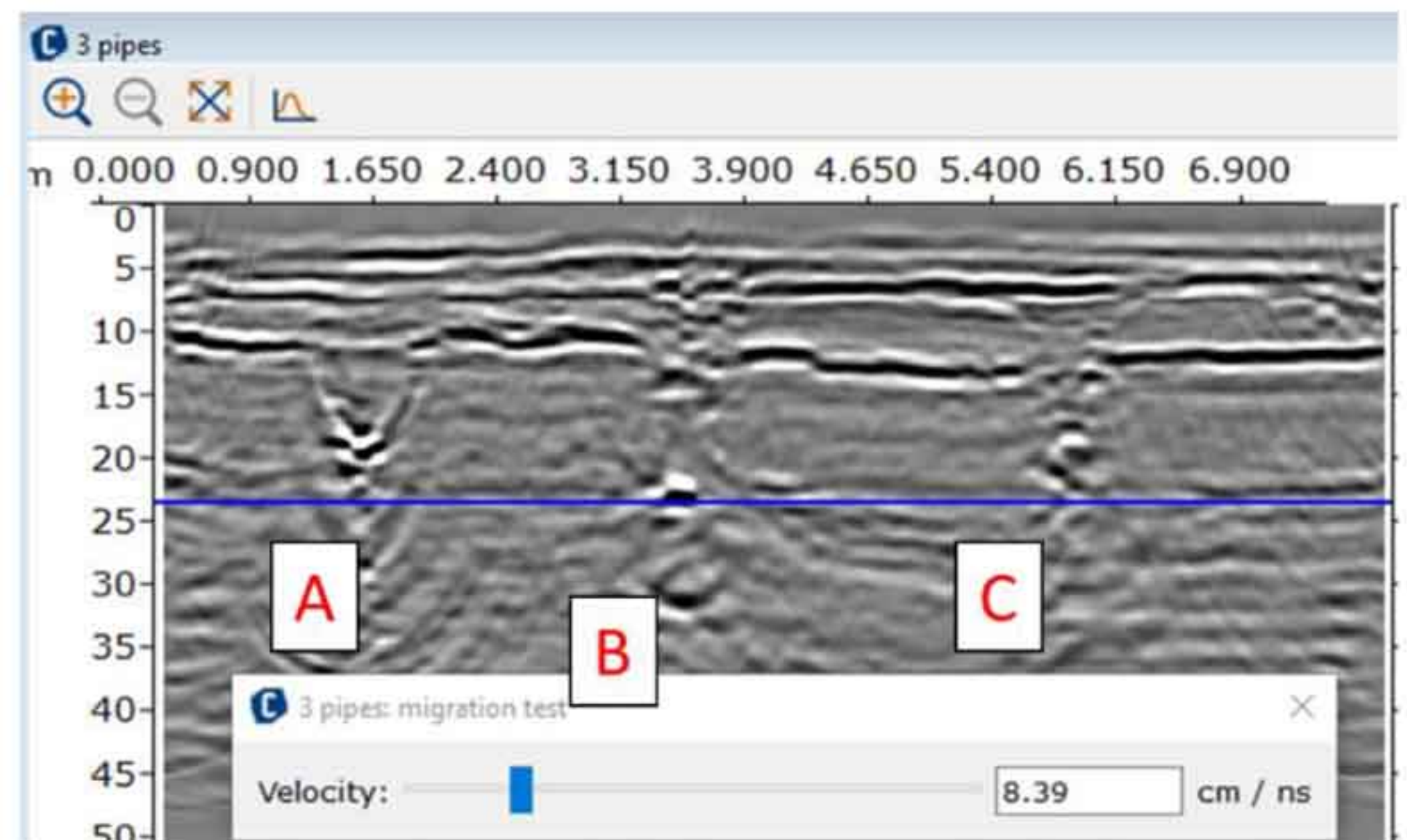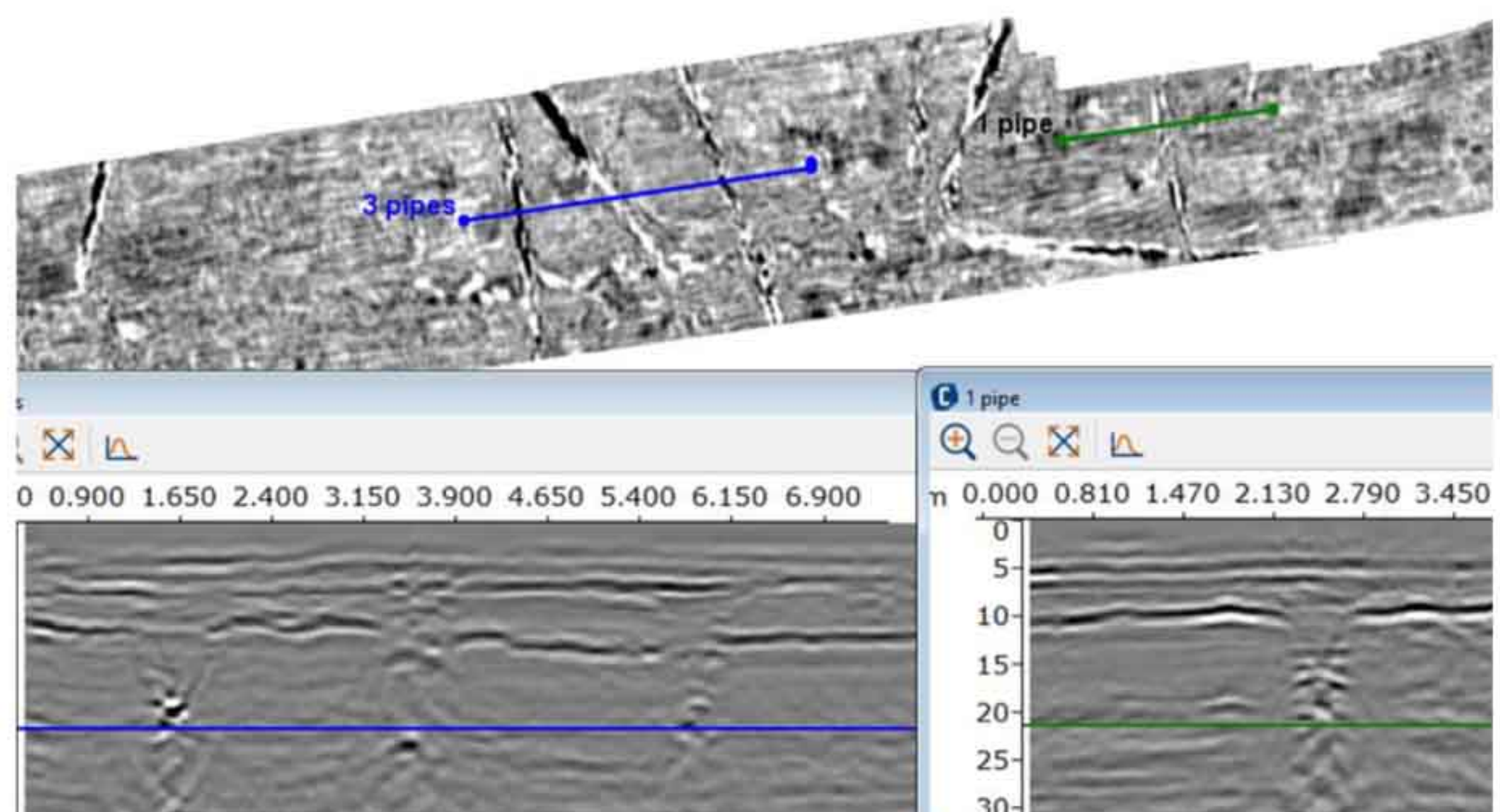sing, and those discussions will follow in subsequent notes.