Preliminary results using Acceleware's AxRTM API

Babatunde Arenrin*, Gary Margrave and Rolf Maier
arenrin@ucalgary.ca

ABSTRACT

We describe a seismic modelling program based on modules designed to perform forward modelling and reverse time migration, and show results derived from the code. We perform forward modelling and reverse time migration on a simple velocity model consisting of three horizontal layers. We migrated a shot record with a split spread configuration, having 800 receivers 10 m apart.

The migrated shot record from AxRTM produced desirable results, however, there are some low frequency artefacts that need to be filtered for improved result. The forward modelling and reverse time migration program have been benchmarked and compared to each other.

Code snippets

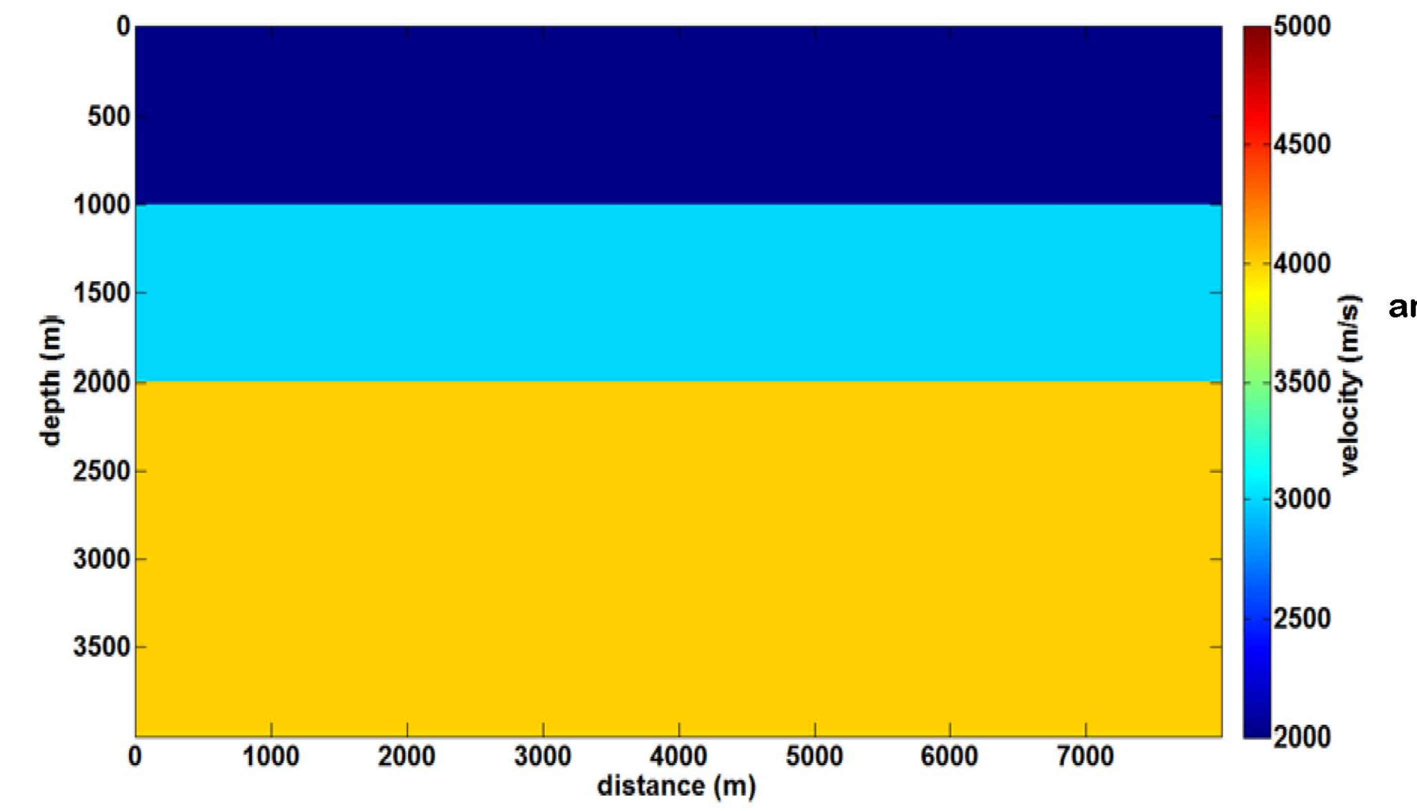
The sample code below shows how the forward modelling code performs time steps, printing an occasional status message. The last call finishes the job before the result is extracted from the internal buffers and written to disk.

```
/*-------------------------------------------------------------------------*
 * 3F. Forward Pass                                                        *
 *-------------------------------------------------------------------------*/
 printf (" * Forward Pass\n");

 /* Perform timestepping */
 ax_uint_t        timestep = 0;
 ax_uint_t        batchSize = 50;
 ax_uint_t        remaining;

 AX_CALL (AxGetNumTimestepsRemaining (jobHandle, &remaining));

 do {
     if (remaining < batchSize)
         batchSize = remaining;

     printf ("    - Performing forward timesteps %d to %d\n", timestep,
             timestep+batchSize-1);
     AX CALL (AxDoTimesteps (jobHandle, batchSize));

        timestep += batchSize;

        AX_CALL (AxGetNumTimestepsRemaining (jobHandle, &remaining));
   } while (remaining > 0);

   AX_CALL (AxFinalizeForwardPass (jobHandle));/* Finalize Forward Pass stage*/
```

FIG. 1. Code snippet for the forward modelling engine.

Synthetic examples

To demonstrate what is obtainable using AxRTM, we produced a shot gather and a migrated image for a horizontal three-layer model. We also benchmarked the results with the forward modelling (Afd_shotrec_alt) and migration program (PSPI) in Matlab that is available in the CREWES toolbox. The migrated images are presented in Figures 6 and 7.



and

FIG. 2. Velocity model for the tests.

Shot records

Figure 2 shows the velocity model. Figure 3 is a shot record using AxRTM, and Figure 4 is from Afd_shotrec_alt from CREWES Matlab toolbox. In both cases the parameterization set are exactly the same i.e. similar propagation steps, sample intervals, and wavelet (minimum phase).

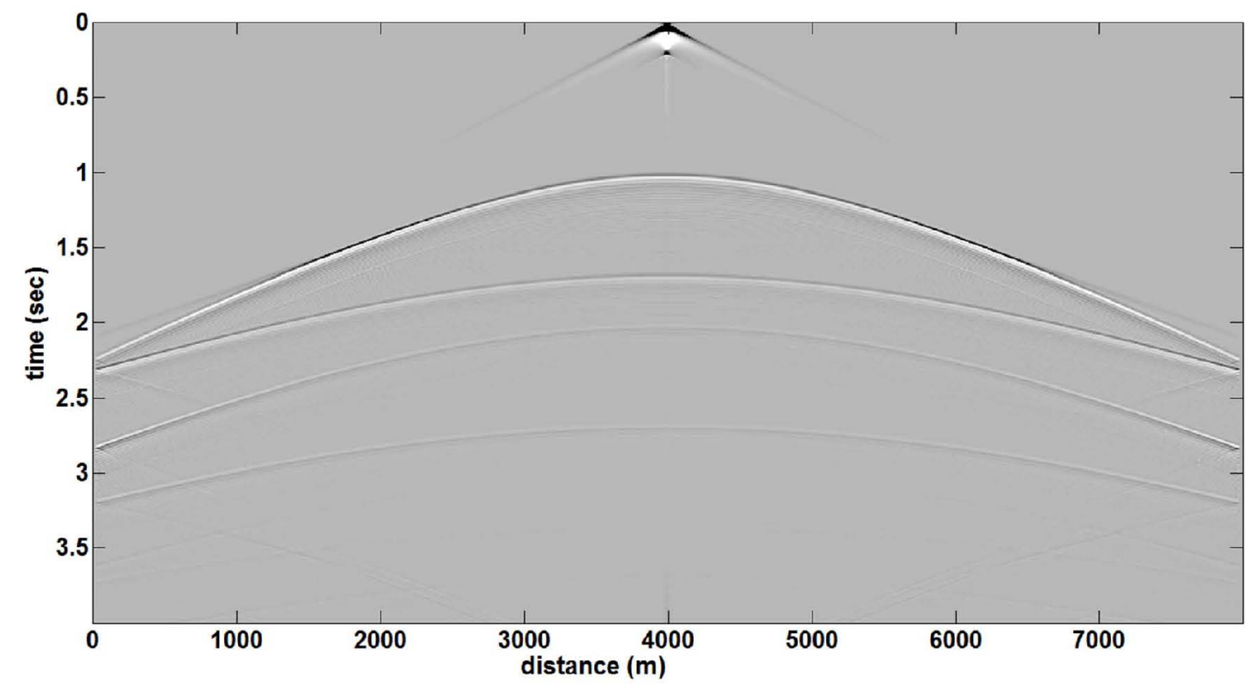


FIG. 3. Shot record from AxRTM.

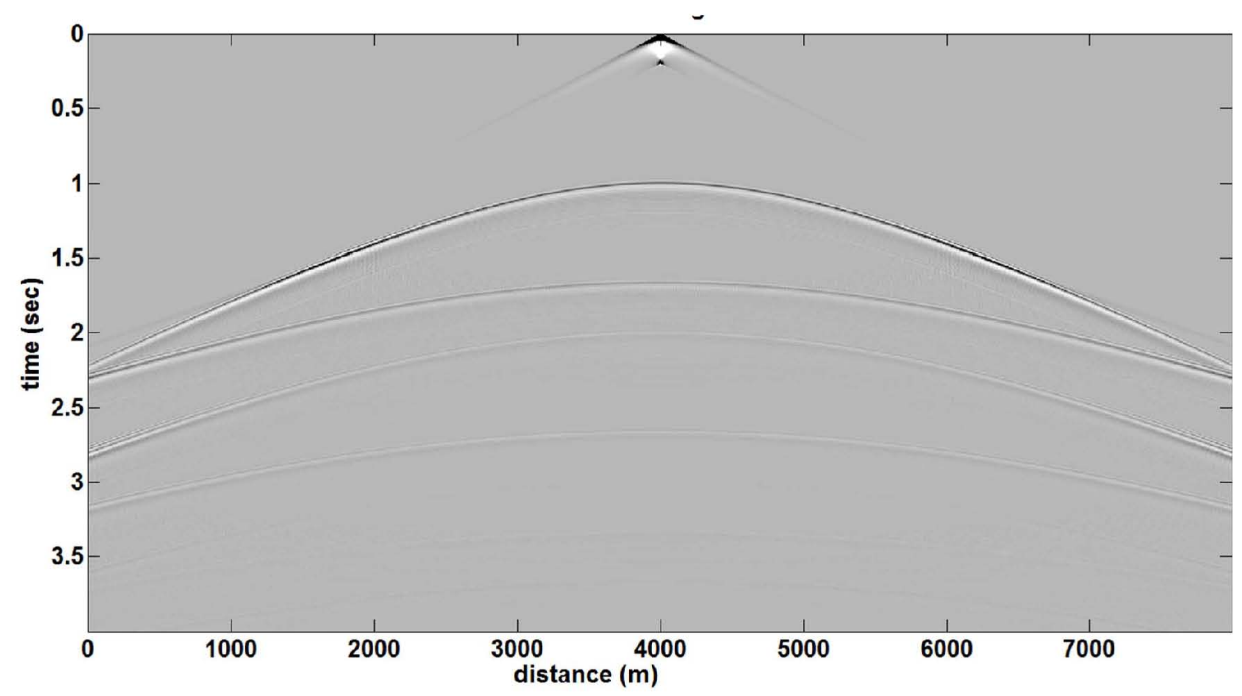


FIG. 4. Shot record from Afd_shotrec_alt from the CREWES Matlab toolbox.

Amplitude spectra of trace at 500 m

After amplitude balancing, Figure 5 shows an overlay of the amplitude spectrum of the traces. We observed from Figure 5 that despite the bandwidth of the two traces being slightly different, they are in phase with each other.

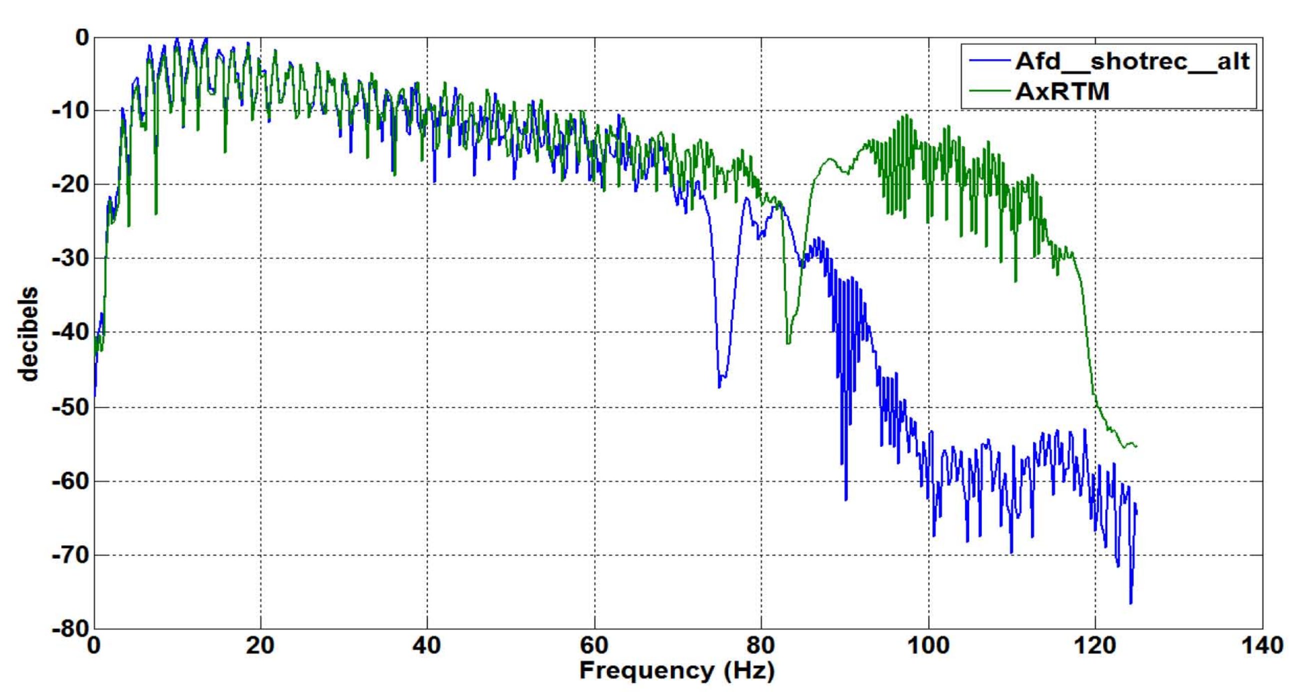


FIG. 5. Amplitude spectrum of the trace at 500 meters. Spectrum of AxRTM (green) and Afd_shotrec_alt (blue).

Migrated images

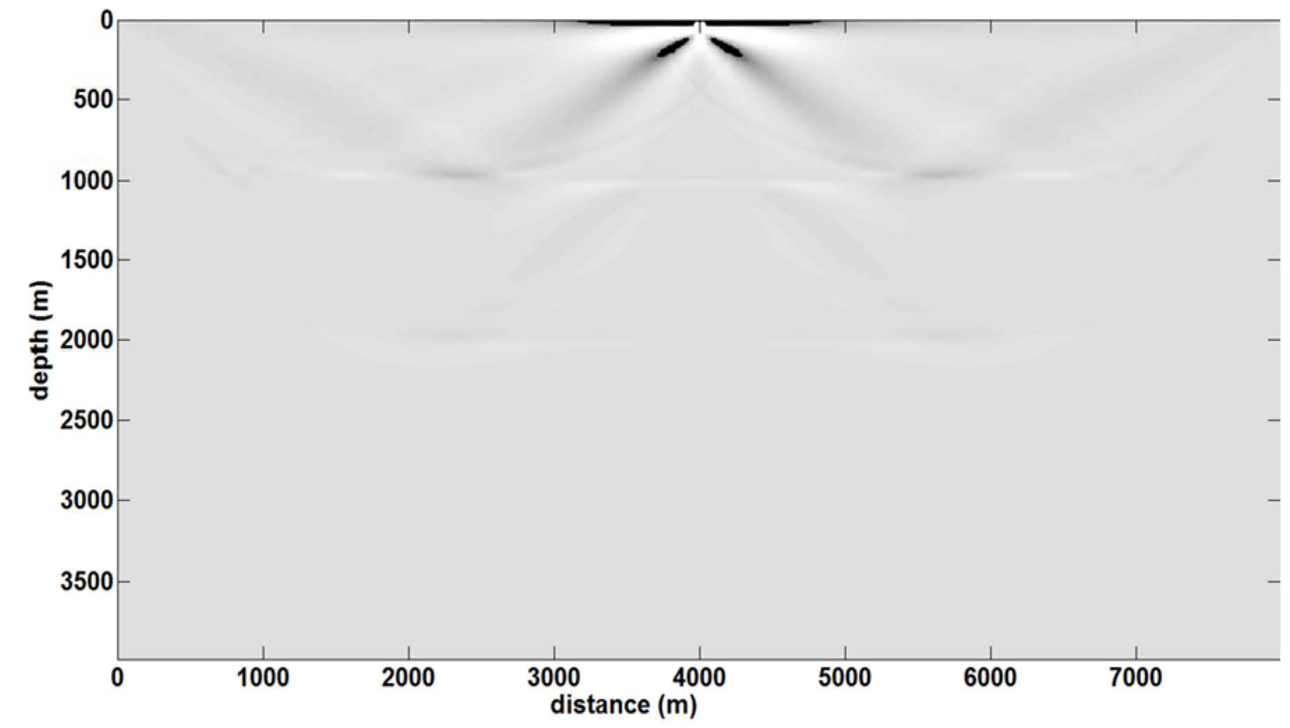


FIG. 6. Migrated shot from AxRTM.

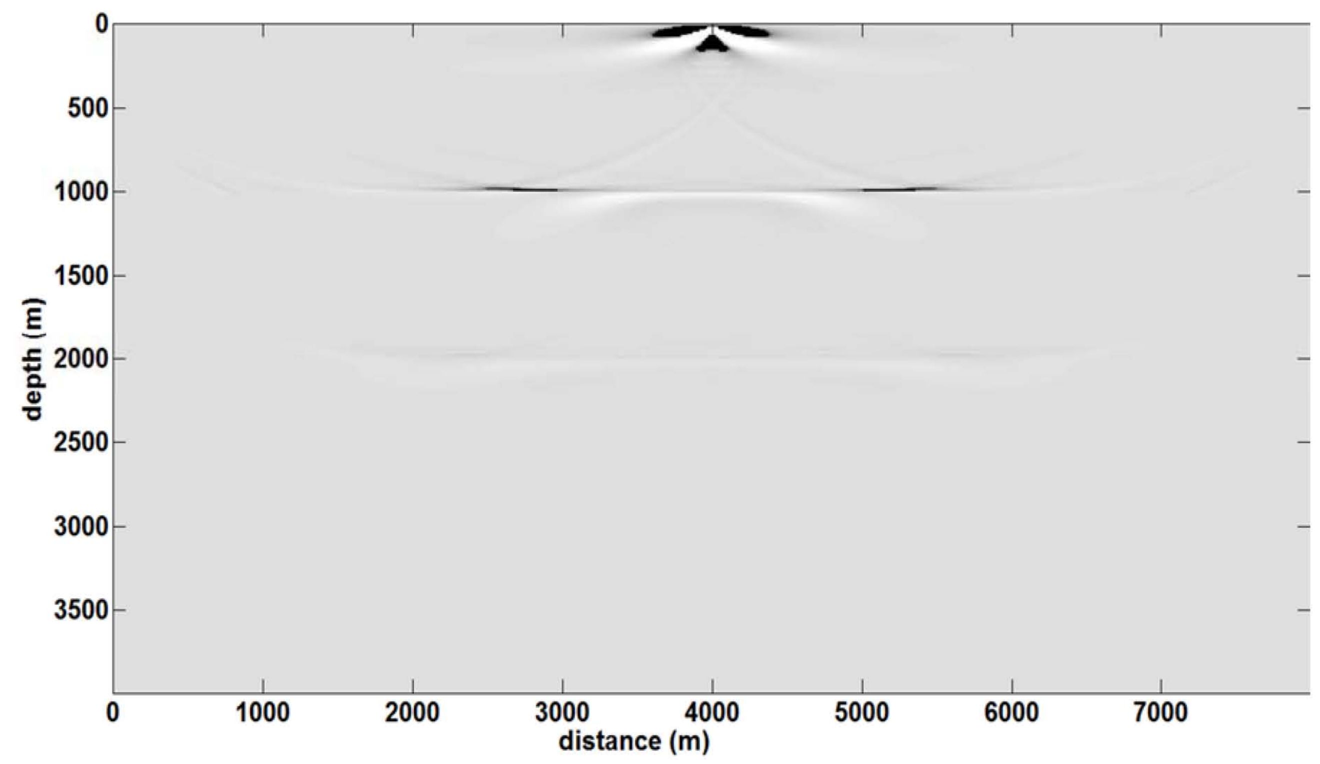


FIG. 7. Migrated shot from PSPI code written in Matlab.

Comparing the migrated images above, the migrated image from AxRTM's reverse time migration appears to suffer from low wavenumber artefacts in the upper section of the image. This is not a problem since a filter can be used to get rid of the artefacts. The PSPI algorithm on the other hand seems to be less affected by cross correlation artefacts. However, both algorithms seem to have produced a decent image of the shot gather.

DISCUSSIONS

Tests showed that Acceleware's forward modelling and reverse time migration program (AxRTM) runs at least ten times faster than the corresponding Matlab programs. However, the Matlab programs ran on CPUs while Acceleware's programs were made to run on a single graphics card. Allowing for multiple graphics cards in the case of a single shot test slows down a run significantly because of the time taken to initialize the cards. We have not done large array examples.

CONCLUSIONS

We have been able to show that AxRTM API from Acceleware can be used for forward modelling and reverse time migration. The results from AxRTM are comparable to the results from forward modelling and migration codes written in Matlab from the CREWES toolbox. However, the migrated image from AxRTM suffers from artefacts or crosstalk inherent in any reverse time migration algorithm. However, the package comes with the flexibility of applying a filter to the migrated image.