

# Reverse time migration with random boundaries

*Robert G. Clapp*

## ABSTRACT

Reading wavefield checkpoints from disk is quickly becoming the bottleneck in Reverse Time Migration. We eliminate the need to write the wavefields to disk by creating an increasingly random boundary around the computational domain when propagating the source function. The wavefield that encounters the boundary region is pseudo-randomized. Reflections off the random layer have minimal coherent correlation with the receiver wavefield but can be reformed by running the wave equation in the reverse direction. This allows the source to first be propagated to the maximum recording time and then to be backward propagated simultaneously with receiver wavefield significantly reducing memory and IO requirements. I demonstrate the methodology on the Sigsbee synthetic and show that it significantly reduces coherent correlation artifacts.

## INTRODUCTION

Reverse Time Migration (RTM) (Baysal et al., 1983) is quickly becoming the standard for high-end imaging. At the core of the RTM algorithm is a modeling kernel. The simplicity of the the modeling kernel has led to high-performance implementation on Field Programmable Gate Arrays (FPGA) (Nemeth et al., 2008), General Purpose Graphics Processing Units (GPGPU) (Micikevicius, 2008), and conventional processors. Of growing significance is the problem that the source field must be propagated from  $t = 0$  to  $t = \text{maxtime}$  while the receiver wavefield must be propagated from  $t = \text{maxtime}$  to  $t = 0$  since the fields must be correlated at equivalent time positions. One propagation must be stored either completely or in a check-pointed manner to disk.

Symes (2007) and Dussaud et al. (2008) discuss checkpointing methods to handle the different propagation directions. Dussaud et al. (2008) and Clapp (2008) suggest an alternate approach of propagating the source wavefield to the maximum recording time and then reversing the propagation to make it consistent with the receiver propagation direction. The use of damping schemes around the boundary results in the need to inject energy from the non-damped, forward propagated wavefield, inside the boundary region. The RAM requirement with this scheme is less than conventional checkpointing approaches but still imposes significant disk IO requirements.

In this paper, I discuss an alternate approach. I replace the conventional damped region with an increasingly random velocity region. Rather than eliminate reflections

I distort them to minimize coherent correlations with the receiver wavefield that cause artifacts. I begin by describing the construction of the random boundary. I then demonstrate the amplitude behavior of the time reversed wavefield. I conclude by showing the methodology applied in a 2-D synthetic.

## BACKGROUND

The basic idea behind RTM is to propagate a source function within the computational domain from time  $t = 0$  to  $t = \text{maxtime}$ , storing the wavefield  $s(t, \mathbf{x})$  at time steps consistent with the time sampling of the data  $dt$ . The recorded data is injected into a second computational domain and propagated from  $t = \text{maxtime}$  to  $t = 0$  and stored in  $r(t, \mathbf{x})$ . The final image  $I$  is constructed

$$I(\mathbf{x}) = \sum_{t=0}^{\text{maxtime}} s(t, \mathbf{x})r(t, \mathbf{x}), \quad (1)$$

or some similar imaging condition. The problem is that  $s(t, \mathbf{x})$  and  $r(t, \mathbf{x})$  are too large to store in memory, and often too large store on disk. The image can be updated while propagating the receiver wavefield but the different propagation directions of the source and receiver wavefields, still introduce a large storage requirement. To reduce the storage requirement, checkpointing schemes are used. The source wavefield is stored at various intervals  $d_{check}$  during forward propagation. When propagating the receiver wavefield these checkpoints are read from disk and re-propagated into a buffer to be correlated with the receiver propagation. Algorithm 1 illustrates this approach. There are several undesirable features to this approach. First, the source

---

### Algorithm 1 Standard RTM with checkpointing

---

```

for all shots do
  while  $t < \text{maxtime}$  do
    Advance source wavefield to check point ( $d_{check}$ )
    store wavefield
  end while
   $t = \text{maxtime}$ 
  while  $t > 0$  do
    Read source wavefield at  $t - d_{check}$ 
    Advance and buffer source wavefield to  $t$ 
    for  $t = 0$  to  $t = d_{check}$  do
      Advance receiver wavefield  $-dt$ 
      Correlate source and receiver wavefield at constant  $t$ 
    end for
     $t = t - d_{check}$ 
  end while
end for

```

---

wavefield must be re-propagated. Second, buffering of the re-propagated source wavefield introduces a large memory requirement. Finally, reading the snapshots from disk, particularly when the propagation is done on an accelerator, can/does make RTM IO bound. A wavefield can be propagated forward or backwards in time. On first glance, an obvious solution to the storage and IO requirements is to propagate the source wavefield to  $t = \text{maxtime}$  and then reverse the propagation eliminating the need for checkpointing. The flaw in this hypothesis comes from the boundary conditions we conventionally use.

## BOUNDARY CONDITIONS

Ideally, we would like to emulate an infinite computational domain when propagating our seismic wavefield. Computationally, this is an unachievable goal; instead we try to minimize artifacts, reflections from the boundaries, caused by our limited computational domain. The most effective method is the Perfectly Matched Boundary method from the electro-magnetic community (Berenger, 1994). This method amounts to mapping the coordinate system into the complex domain and changing the propagating wavefield into a decaying wavefield. A second common technique is to introduce a damping region at the edge of the computation domain (Cerjan et al., 1985; Baysal et al., 1984). This is often combined with techniques to kill plane waves that are perpendicular to the computational boundary. All of these techniques have proven effective for modeling seismic data, but force propagation in a single direction.

One technique to allow time reversal is to store the wavefield that has not hit the boundary region at each time step and then re-inject when reversing the propagation direction (Clapp, 2008). This technique has the advantage of eliminating the need for buffering, but still requires a large volume of data to be stored on disk.

Another approach is to rethink what we are attempting to accomplish with our boundary conditions for RTM. Ideally, we would like to eliminate all reflections from the edge of our computational domain, but what we are really concerned with is *coherent* reflections. If we can distort the wavefield coming from the boundaries so it does not coherently correlate with the receiver wavefield, we will have accomplished our goal.

In acoustic modeling, one way to manipulate the boundaries while still allowing time reversal, is to introduce a random component to the velocity field at the boundary. We must be careful in how we modify the boundary, both by staying within the stability constraint of our finite difference method, and by slowly introducing random numbers to avoid immediate reflections off the randomized zone. The basic algorithm for constructing the boundary can be found below. Panel ‘A’ of Figure 1 shows a velocity model constructed with a random boundary and panel ‘B’ shows a cross section through that boundary. Note how the variability of the velocity increases near the edge of the computational domain.

Panel ‘A’ of Figure 2 shows the wavefield at  $t = .63$  after injecting a source in

**Algorithm 2** Creating random boundary

---

```

for all x,y,z do
  if within boundary region then
     $d$ =distance within boundary
    found=false
    while found==false do
      select random number  $r$ 
       $v_{test} = v(x, y, z) + r * d$ 
      if  $v_{test}$  meets stability constraint then
         $v(x, y, z) = v_{test}$ 
        found=true
      end if
    end while
  end if
end for

```

---

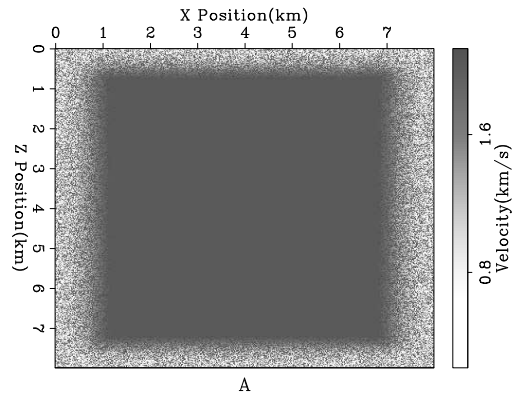
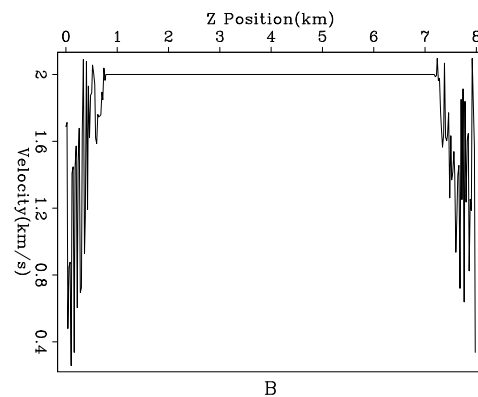


Figure 1: Panel A shows the constant velocity model with a random boundary. Panel B shows a cross section through the velocity model. Note the increasing random nature of the boundary. [ER]



the center of the computational domain. Panel ‘B’ shows the wavefield at  $t = 2.205$  after the wavefield has hit the boundary. Note the absence of a coherent reflection off the boundary. Panel ‘C’ shows the wavefield at  $t = 3.906$ , when no coherent energy is present. The wavefield was then propagated to  $t = 5.0$  and the computation was reversed. Panel ‘D’ shows the reversed wavefield at  $t = .63$ , and panel ‘E’ shows the difference between wavefields scaled by 10,000. The difference between the two images is in the range of the machine precision.

To see if there is a coherent pattern underneath the random looking field, I repeated the experiment 16 times, each with a different random boundary. Panels ‘F-J’ show the average of these 16 experiments at the same times as panels ‘A-E’. Note in panel ‘G’ how the average of the 16 experiments has greatly reduced the energy reflected from the boundary. In panel ‘H’ we can see that there is a low energy, low spatial frequency reflection from the boundary. Panel ‘J’ shows that even the machine precision noise tends to cancel.

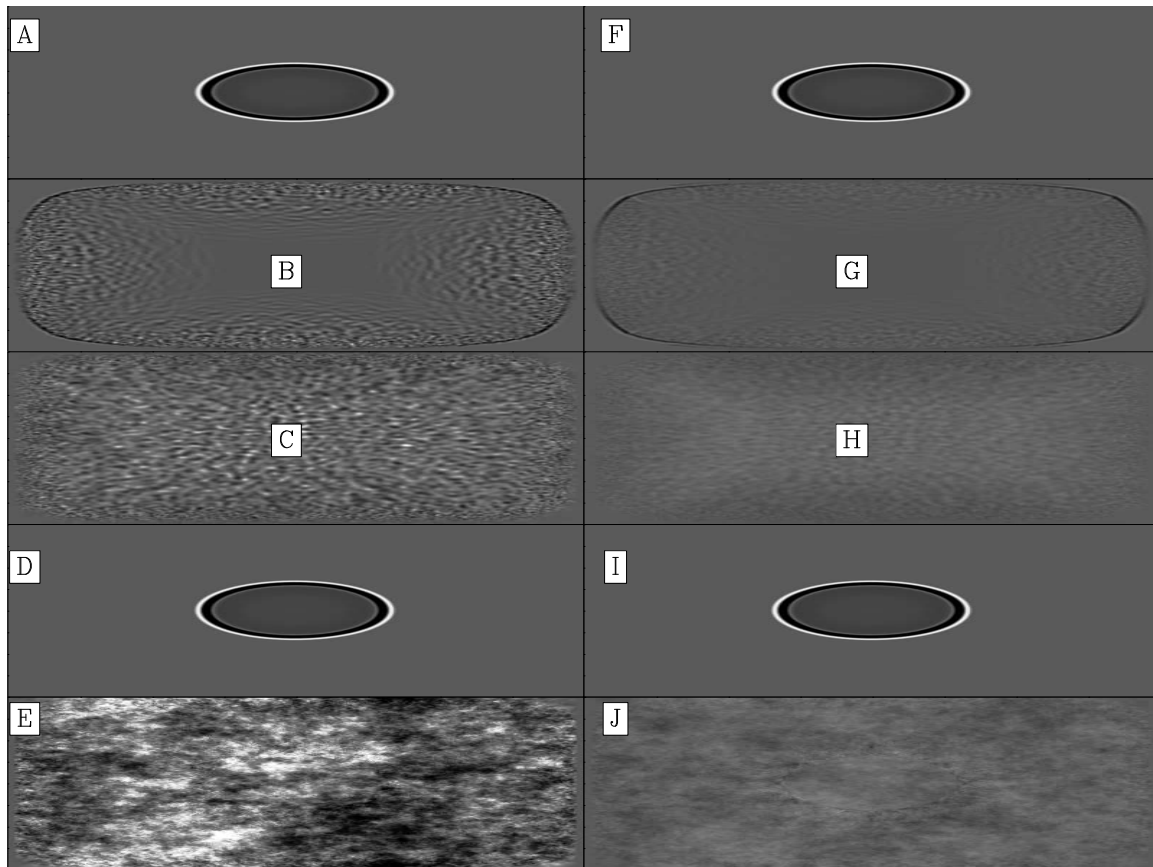


Figure 2: The left column shows snapshots from a single modeling experiment at .63, 2.205, and 3.906. The fourth panel shows the result of time reversing the computation to .63 seconds. The bottom panel shows the difference between first and fourth panel scaled by 10000. The right column shows the summing of 16 modeling experiments each with different random boundaries. [ER]

Panels ‘A’ and ‘B’ of Figure 3 show the results of Fourier transforming the data shown in panels ‘C’ and ‘H’ of Figure 2. Panel ‘A’ demonstrates that the wave field is dominated by low-spatial wave-number features. In panel ‘B’ we see that almost all of the energy is concentrated at the low spatial wave-numbers. This is not surprising. At low enough frequencies, our random boundary does not affect our propagating wavefield. By increasing the size of the random zone, we can damp lower frequencies.

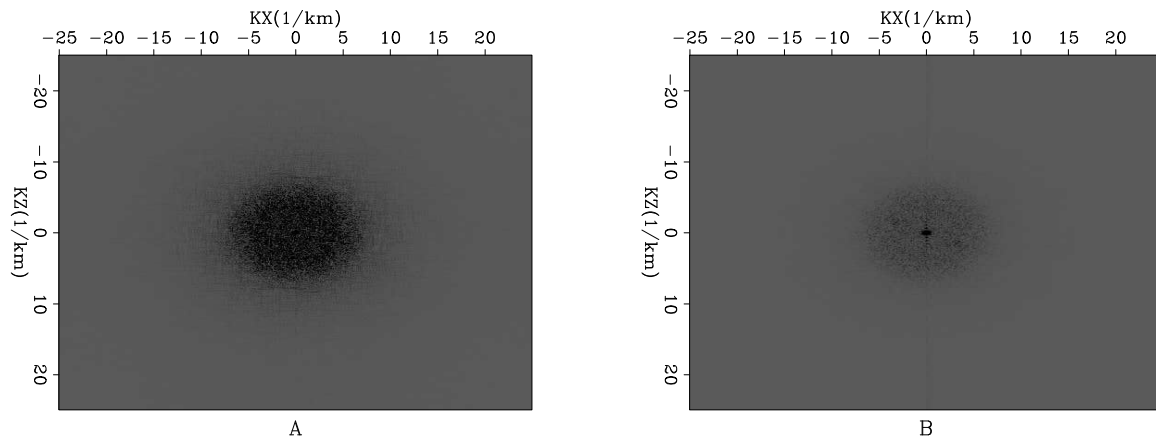


Figure 3: Panel ‘A’ is the result of Fourier transforming the data shown in panels ‘C’ and ‘H’ of Figure 2. Note how the summing of multiple experiment reduces the energy in higher spatial wave-number events. [ER]

Summing multiple experiments is directly applicable to RTM. In RTM we are summing the result from many migrated shots to form our final image. As a result we will get a signal-to-noise boost from neighboring shots having different random patterns.

By making these changes, the RTM algorithm simplifies to the following template.

---

**Algorithm 3** Time reverse RTM with random boundaries

---

```

for all shots do
  Create random boundary around computational domain
  Advance source wavefield to  $t = \text{maxtime}$ 
  for  $t = \text{maxtime}$  to  $t = 0$  do
    Advance source wavefield  $-dt$ 
    Advance receiver wavefield  $-dt$ 
    Correlate source and receiver wavefield
  end for
end for

```

---

## SYNTHETIC EXAMPLE

To test the methodology, I implemented both algorithm 1 and 3. I used the first 4.5 seconds of the Sigsbee synthetic, limiting the computational domain to the first 4.5 km. As a baseline I performed RTM using a reflecting boundary condition. Locations ‘A’ and ‘B’ in Figure 4 show where reflections from the top of the domain coherently correlated with the receiver wavefield.

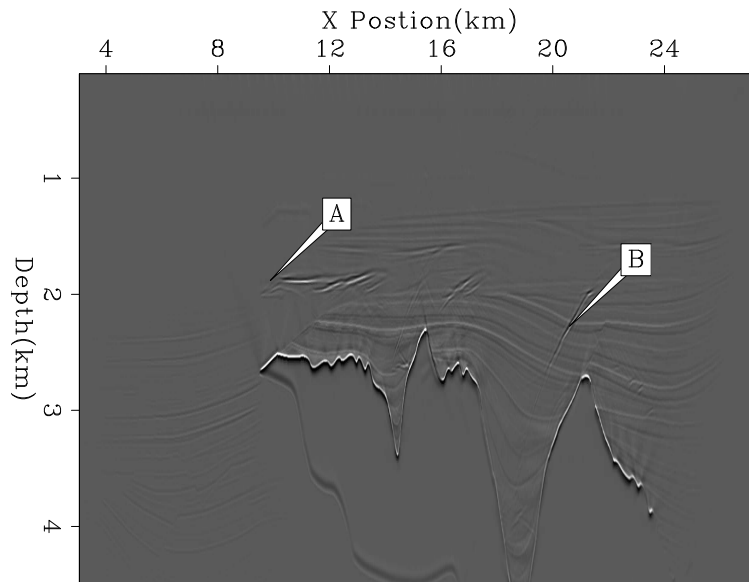


Figure 4: The result of RTM migration using a reflecting boundary. Note the obvious boundary artifacts at ‘A’ and ‘B’. [CR]

Figure 5 shows the RTM result using algorithm 1. In this case I applied a damped exponential in a 30 point region around the computational domain. Note how the artifacts at ‘A’ and ‘B’ are significantly reduced. Finally, Figure 6 shows the result of algorithm 3. Note how the energy at ‘A’ and ‘B’ are further reduced. Some additional coherent correlations are present in the water layer but are not visible lower in the image once sufficient reflected energy is present.

## OTHER APPLICATIONS

There are two notable additional applications to this approach. We can think of the random boundary region as a series of point sources that are excited at different times. By cross-correlating the energy at any two locations we can generate a two-way interferometric Green’s function.

In addition, the random boundaries offer the potential to use implicit rather than explicit propagation. We can treat our medium as a single 1-D trace (helixize (Claerbout, 1998) the computational domain). We can then use 1-D theory to calculate causal-minimum phase filters that can be applied using polynomial division. This

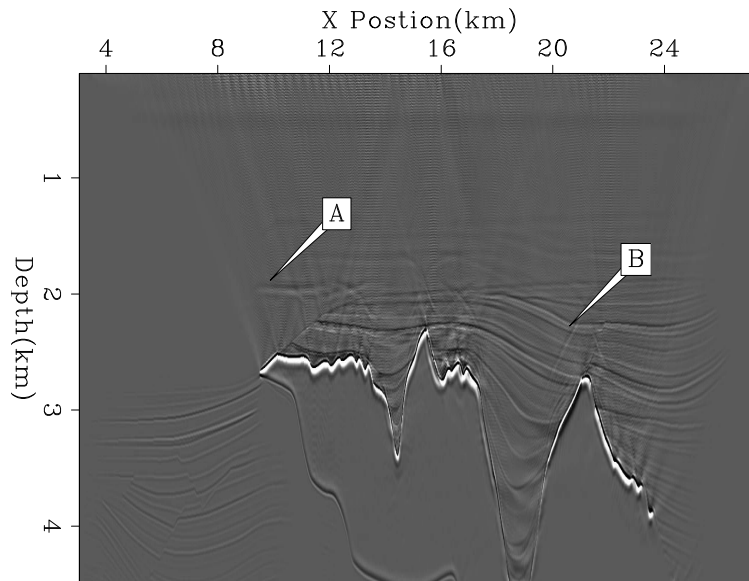


Figure 5: The result of RTM migration using a damping boundary condition following the methodology described by algorithm 1. Note how the boundary artifacts have been reduced at 'A' and 'B'. [CR]

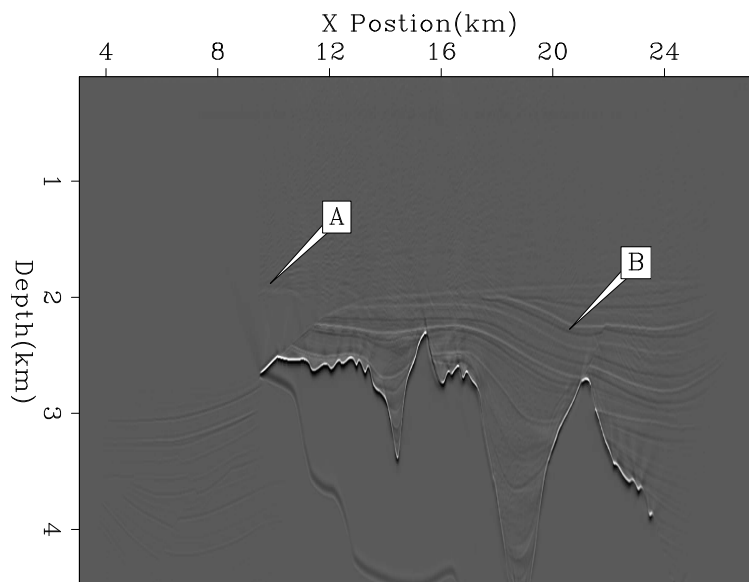


Figure 6: The result of RTM migration using a random boundary condition using the methodology described by algorithm 3. Note the greatly reduced artifacts at 'A' and 'B' [CR]



technique was applied to downward continuation by Rickett et al. (1998) but suffered from obvious wraparound problems. The random boundaries discussed in this paper would minimize this problem.

## CONCLUSIONS

Pseudo-random boundary conditions effectively distort an incoming wavefield. I use these boundary conditions to propagate the source wavefield in RTM both forward and backwards. The distorted wavefield correlates poorly with the receiver wavefield, minimizing boundary artifacts. I hypothesize that these boundaries have additional applications for implicit finite difference and interferometry.

## ACKNOWLEDGMENTS

We would like to thank the SmartJV for the use of the Sigsbee synthetic. research.

## REFERENCES

- Baysal, E., D. D. Kosloff, and J. W. C. Sherwood, 1983, Reverse time migration: *Geophysics*, **48**, 1514–1524.
- , 1984, A two-way nonreflecting wave equation: *Geophysics*, **49**, 132–141.
- Berenger, J., 1994, A perfectly matched layer for the absorption of electromagnetic waves: *Journal of Computational Physics*, **114**, 185–200.
- Cerjan, C., D. Kosloff, R. Kosloff, and M. Reshef, 1985, A nonreflecting boundary condition for discrete acoustic-wave and elastic-wave equations (short note): *Geophysics*, **50**, 705–708.
- Claerbout, J. F., 1998, Multi-dimensional recursive filtering via the helix: *Geophysics*, **63**, 1532–1541.
- Clapp, R. G., 2008, Reverse time migration: Saving the boundaries: 2008, **136**, 136–144.
- Dussaud, E., W. W. Symes, L. Lemaistre, P. Singer, B. Denel, and A. Cherrett, 2008, Computational strategies for reverse-time migration: 78th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, SPMI 3.3.
- Micikevicius, P., 2008, 3d finite difference computation on gpus using cuda: 2nd Workshop on General Purpose Processing on Graphics Processing Units, Expanded Abstracts, 79–84.
- Nemeth, T., J. Stefani, W. Liu, R. Dimond, O. Pell, , and R. Ergas, 2008, An implementation of the acoustic wave equation on fpgas: 78th Ann. Internat. Meeting, Expanded Abstracts, 2874–2877, Soc. Expl. Geophys.
- Rickett, J., J. Claerbout, and S. B. Fomel, 1998, Implicit 3-D depth migration by wavefield extrapolation with helical boundary conditions: 68th Ann. Internat. Mtg, 1124–1127, Soc. of Expl. Geophys.

Symes, W. M., 2007, Reverse time migration with optimal checkpointing: *Geophysics*, **72**, SM213–SM221.