14th Biennial International Conference and Exposition
SPG 2023

Evaluation of RTM implementation approaches using conventional and boundary wavefield savings

Richa Rastogi[*1], *Abhishek Srivastava*[1], *Nithu Mangalath*[1], *Monika Pokharkar*[1], *Laxmaiah Bathula*[1], *Suhas Phadke*[1] *and Saheb Ghosh*[2]
[1]*Centre for Development of Advanced Computing (C-DAC), Pune*
[2]*GEOPIC, ONGC, Dehradun*
[*]*richar@cdac.in*

Keywords
RTM, HPC, Boundary Saving RTM, Seismic Imaging, Seismic Migration

Summary

In this paper, we report comparison of developed Conventional wavefield Saving (CS) and Boundary wavefield Saving (BS) approaches for Reverse Time Migration (RTM) implementations using synthetic and field data. Both the approaches are evaluated for their HPC resource utilization and accuracy of the outcome. 2D acoustic wave equation with constant density is used for two-way wave propagation for RTM. The finite difference scheme which is solved using staggered grid approximation is accurate to 10^{th} order in space and 2^{nd} order in time. The developed applications are parallelized over number of CPU nodes of a HPC cluster. The Memory/storage requirement and compute time for BSRTM and CSRTM approaches are discussed in this paper.

Introduction

Seismic Migration is the key technology for producing an optimal image of the earth's subsurface containing complex geologies like salt bodies, folded sedimentary layers or faulted structures. These complex geological structures are illuminated by various seismic wave paths, and it is difficult for conventional one-way or ray based seismic migration algorithms to image such complex structures. RTM algorithm is based on the solution of two-way wave equation which improves imaging and further deals with the complicated wave propagation and imaging tasks, especially for complex media. It is a preferred solution for imaging seismic data in areas of complex wave phenomena. Although, RTM provides more accurate image, meeting its high compute resource and memory requirements is still a challenge.

Many strategies like full wavefield storage, source wavefield reconstruction and non-reconstructive imaging, have been evaluated to make the source wavefield accessible when the receiver wavefield is generated (Bao D and George A, 2015). (Clapp, 2009) demonstrated RTM with random boundaries, which allowed source wavefield propagation simultaneously with receiver wavefield and decreased memory usage. However, it is prone to generate background noise when strong reflecting interfaces are encountered near imaging boundary area. (Tang and Wang, 2012) demonstrated that saving wave fields on boundaries of target area is effective in reducing memory requirement without loss of accuracy. (Yang et al., 2014) also demonstrated validity of boundary saving technique combined with staggered finite difference scheme and Convolutional Perfectly Matched Layer (CPML) boundary condition. (Tan and Huang, 2014) proposed new method for 3D RTM, in which boundary wavefields only at one or two layers of spatial grid points are stored for source wavefield reconstruction. For an 8^{th} order accurate FD scheme in space, the memory utilization is only 37.5% to that of the conventional approach.

The presented research work discuss and compare the CSRTM and BSRTM implementations of in-house developed SeisRTM software. The key features of SeisRTM software are described in the theory and methodology section. CSRTM and BSRTM implementations are demonstrated using high resolution Marmousi2 model[*] and field data. Accuracy of the results and the computational resources evaluation is performed using PARAM Shivay[†] cluster stationed at Supercomputing Centre, Indian Institute of Technology (BHU), Varanasi, India.

Theory and Methodology

In RTM, the source wavefield is propagated forward in time (0 to t_{max}), while the receiver wavefield is propagated backwards in time (t_{max} to 0) for all shots. Here, t_{max} is the recording time of seismic data. These two wavefields are cross-correlated at corresponding time steps before imaging. For current RTM implementation, the source and receiver wave fields are propagated using acoustic wave equation in isotropic medium with constant density given in Equation 1.

$$\frac{1}{v^2(\mathbf{U})}\frac{\partial^2 p(\mathbf{U},t)}{\partial t^2} = \nabla^2 p(\mathbf{U},t) + f(t)\delta(\mathbf{U}=\mathbf{u_s}) \qquad (1)$$

Where, v is acoustic velocity, p is acoustic pressure, t is time, U is dimensional parameter and u_s is shot location. Equation 1 is implemented using staggered grid Finite Difference (FD) method which is accurate to 2^{nd} order in time and 10^{th} order in space (Londhe et al., 2021). Here, boundary reflections are absorbed using CPML boundaries. Numerical stability and dispersion checks are verified using Equation 2.

$$dt \leq \frac{0.759 \times min(dx,dz)}{v_{max}}, \quad dh \leq \frac{v_{min}}{3.19 \times f_{max}} \qquad (2)$$

Normalized Cross-correlation (NCC) imaging condition given in Equation 3 is applied for imaging.

[*]https://wiki.seg.org/wiki/AGL_Elastic_Marmousi
[†]https://iitbhu.ac.in/cf/scc/param_shivay/architecture

$$I(x,z) = \frac{\sum_t S(t,x,z)R(t,x,z)}{\sqrt{\sum_t S^2(t,x,z)\sum_t R^2(t,x,z)}} \quad (3)$$

Where, I is Shot Image Gather, S is source wavefield, R is receiver wavefield, t is FD time step and (x,z) are grid points for a 2D medium.

SeisRTM Software

SeisRTM is an in-house developed HPC seismic imaging software suite developed by C-DAC with the following key features :

- 2D isotropic (ISO) and anisotropic (VTI and TTI) RTM and Modelling
- 3D isotropic RTM and Modelling
- CSRTM and BSRTM approaches with an adaptive design to run in-memory or storage for computations
- Restart facility for the failed jobs
- Choice on imaging conditions such as Cross Correlation (CC), Normalized Cross Correlation (NCC)
- Handles standard SEG-Y headers version 1.0
- Command Line Interface (CLI) and Graphical User Interface (GUI)
- Tools for pre and post processing of seismic data

SeisRTM distributes the shot workload among cluster nodes using Message Passing Interface (MPI) libraries. The required FD computations for migration are further parallelized using OpenMP library. The main challenge for implementation of RTM is the computation of forward and reverse wavefields and their cross-correlation at each FD time step for each shot. The size and resolution of the underlying velocity model for each shot is responsible for compute time and required memory/storage.

RTM Implementations

Migration Strategy

SeisRTM migrates shots using a velocity sub model instead of complete model (Rastogi et al., 2022). It helps in reducing overall memory/storage requirement and compute time for migration of each shot. In addition, it reduces numerical errors from the region outside the receiver spread which in turn improves the migrated stack image. Computational load balancing for migration of uneven shot gathers has been achieved using round-robin distribution. In this paper, resource utilization has been presented for two different RTM approaches for efficient computations.

CSRTM

For CSRTM implementation, the source wavefield is saved in memory/storage depending upon its size. The size of source wavefield is equal to the size of underlying velocity sub model for each shot. The saved wavefield is cross-correlated with corresponding backward propagated receiver wavefield during migration at each time step. CSRTM approach for accessing source wavefield for a particular shot gather is illustrated in Figure 1 along with its pseudo-code.

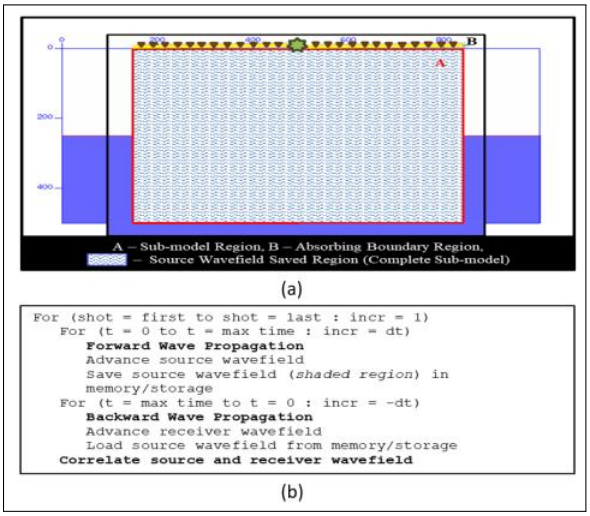

Figure 1: (a) Velocity model under region A is saved for CSRTM implementation;(b) Psuedo code of CSRTM implementation

BSRTM

For BSRTM implementation, sub-model boundary region of the source wavefield is saved in the memory/storage depending upon FD order for each time step and size of the array. Here, the size of saved boundary is governed by the underlying velocity sub model for each shot. The source wavefield is reconstructed for all grid points at each time step using an additional FD modelling before cross-correlating with the receiver wavefield. BSRTM approach for accessing source wavefield for a particular shot gather is illustrated in Figure 2 along with its pseudo-code.

Memory / Storage

Memory/storage utilization are computed using Equation 4 for CSRTM and Equation 5 for BSRTM implementations.

$$CSRTM\ memory = M + [(NX + 2NPML) \times (NZ + 2NPML) \times DTS] \times sizeof(float)\ bytes \quad (4)$$

$$BSRTM\ memory = M + [2 \times (FDOS - 1) \times (NX + NZ) \times FDTS] \times sizeof(float)\ bytes \quad (5)$$

Where, M bytes is the common memory required by both the approaches and it depends on type of implementation and other parameters; NX is number of grid points in X-direction, NZ is number of grid point in Z-direction, $NPML$ is size of CPML

boundary, DTS is number of data time steps, $FDOS$ is FD order in space, $FDTS$ is number of FD time steps.

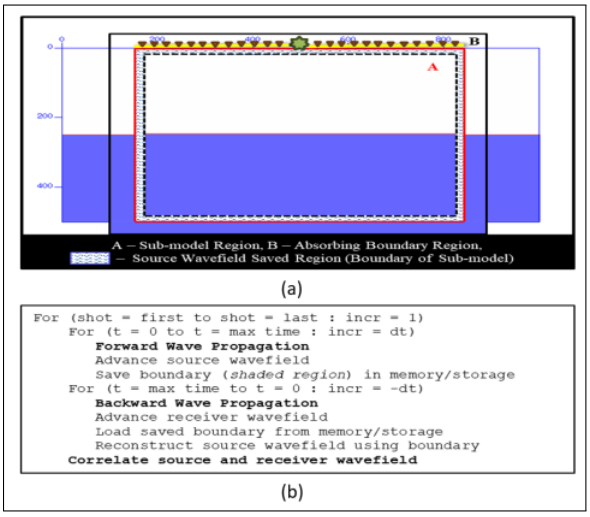

Figure 2: (a) Velocity model's shaded region is saved for BSRTM implementation;(b) Psuedo code of BSRTM implementation

Post Processing

During RTM, low frequency image noise is generated at shallow depths due to high velocity contrasts and cross correlation. We used laplacian filter (Liu et al., 2010) to reduce this noise. For further refinement of RTM images, we have applied appropriate muting to all shot image gathers. The final RTM image is produced by summation of all image gathers of processed shot gathers.

Experiments and results

The developed RTM implementations were tested using synthetic and field data for accuracy of results and evaluated for HPC resource requirements. The underlying system used for this purpose was Intel™ 2*Xeon SKL G-6148 CPU Nodes with 192/768 *GB* memory[‡]. 20 CPU nodes of HPC system were used to make the required runs of SeisRTM software.

Synthetic Data Example

We performed the first experiment using a high resolution synthetic P-wave velocity model, Marmousi2, shown in Figure 3 (a). The model, data, RTM parameters of Marmousi2 and RTM parameters for runs are given in Table 1. The migrated image sections for CSRTM, BSRTM and difference of the two sections are shown in Figure 3 (b), 3 (c) and 3 (d) respectively.

The outcome of both the migration are comparable in terms of reflections and their amplitudes. A difference section was generated using both CS and BS RTM outcome. The amplitude difference is found negligible which is evident from Figure 3 (d) The compute time and memory/storage of the single shot on 192 *GB* and 768 *GB* memory nodes are given in Table 2.

Model parameters	X direction (*Distance*)	29000 m
	Z direction (*Depth*)	3500 m
	Grid points along X (NX)	23 281
	Grid points along Z (NZ)	2 801
	Grid spacing along X (dx)	1.25 m
	Grid spacing along Z (dz)	1.25 m
Data Parameters	Data sampling rate (s)	0.002
	Data time steps (DTS)	5 000
	Number of shots	462
	Number of receivers	480
	Shots, receiver spacing (m)	50, 25
	Near, far offset (m)	12.5, 3 000
FD Parameters	FD sampling rate (s)	0.0001
	FD time steps (FDTS)	100 000
	FD order space (FDOS), time	10, 2
	CPML points (NPML)	64
RTM Parameters	RTM wavelet	Ricker
	Maximum frequency (Hz)	70
	RTM Imaging condition	NCC

Table 1: Model, Data and RTM parameters of Marmousi2 model

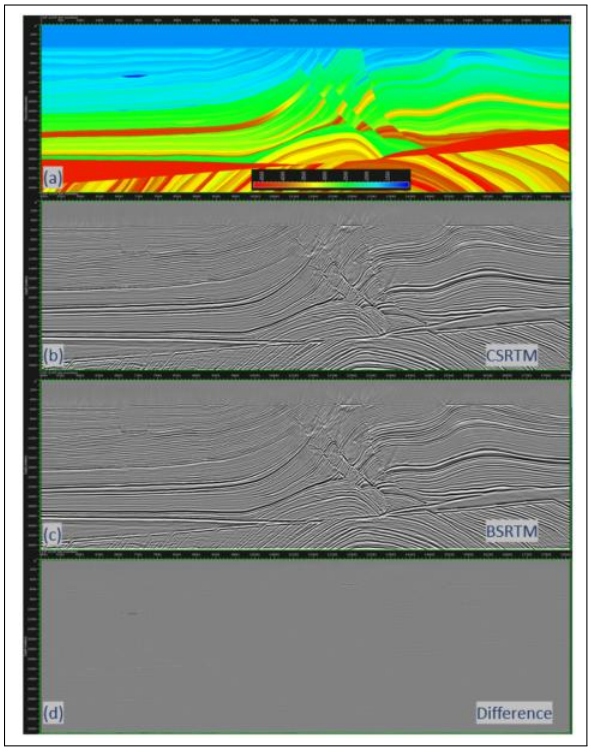

Figure 3: The Marmousi2 (a) P-wave velocity model, migrated stack section using (b) CSRTM, (c) BSRTM and (d) Differnce section

[‡]https://iitbhu.ac.in/cf/scc/param_shivay/architecture

Compute Resource/RTM		CS	BS	Findings
Memory/Storage (GB)		271	40.78	6.65X (BS better)
Compute Time (mins)	192 GB Node	39.21	26.9	1.46X (BS better)
	768 GB Node	17.64	26.9	0.65X (CS better)

Table 2: Memory/storage and compute times for CSRTM and BSRTM for marmousi2 data

Field Data Example

We performed our second experiment on field data. The P-wave velocity model of marine field data is shown in Figure 4 (a). The model, data and RTM parameters of field data runs are given in Table 3.

Model parameters	X direction (*Distance*)	65250 m
	Z direction (*Depth*)	15000 m
	Grid points along X (*NX*)	10 440
	Grid points along Z (*NZ*)	3001
	Grid spacing along X (*dx*)	6.25 m
	Grid spacing along Z (*dz*)	5 m
Data Parameters	Data sampling rate (s)	0.002
	Data time steps (DTS)	5 000
	Number of shots	2260
	Number of receivers	656
	Shots, receiver spacing (m)	25, 6.25
	Near, far offset (m)	112.5, 8300
FD Parameters	FD sampling rate (s)	0.0004
	FD time steps (FDTS)	25006
	FD order space (FDOS), time	10, 2
	CPML points (NPML)	64
RTM Parameters	RTM wavelet	Ricker
	Maximum frequency (Hz)	70
	RTM Imaging condition	NCC

Table 3: Model, Data and RTM parameters of field data

For field data RTM, we pre-processed the input data using reciprocity principle (Rastogi et al., 2023) for better imaging outcome. The migrated stacked images of CSRTM and BSRTM are shown in Figure 4 (b) and 4 (c), respectively whereas difference of the above outcome is given in Figure 4 (d). The compute time and memory/storage of the single shot on 192 *GB* memory nodes are given in Table 4.

Discussion

Synthetic Data

In case of synthetic data, the size of source wavefield to be saved for CSRTM turns out to be 271 *GB* in case of single shot migration. Therefore, we run CSRTM on two different memory nodes to see the effect of memory availability on the compute time of RTM. We found that the memory requirement of single shot is more that the node memory in case of 192 GB memory node, the RTM stores the wavefield on storage which increases the compute time. Whereas, when the same case was run using high memory node it runs 2.26*X* faster due to in-memory computations refer Table2.

Compute Resource/RTM		CS	BS	Observation
Memory/Storage (GB)		163	9.48	17.19X (BS better)
Compute Time (mins)	192 GB Node	5.27	8.38	0.63X (CS better)
	768 GB Node	-	-	-

Table 4: Memory/storage and compute times for CSRTM and BSRTM for field data

The size of source wavefield to be saved for BSRTM turns out to be 40.78 *GB* which is 6.65*X* less than the CSRTM for a single shot migration. As the memory requirement for BSRTM was less than the low memory node's capacity, all the computations were run in-memory and the computing time achieved was 26.9 *mins*. We compare the runtime of CSRTM and BSRTM with respect to low and high memory nodes. It was observed that BSRTM runs 1.46*X* faster when the source wavefield size is more than the memory capacity of the node and 0.65*X* slower when the source wavefield size was within the limits of node memory for CSRTM.

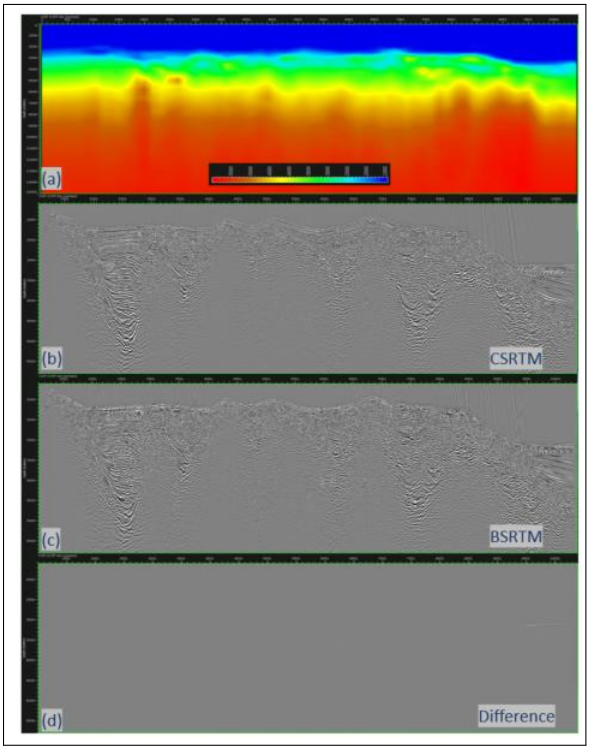

Figure 4: The Marine (a) P-wave velocity model, migrated stack section using (b) CSRTM, (b) BSRTM and (d) Differnce section

Field Data

In case of field data, the size of source wavefield to be saved was 163 *GB* for CSRTM and 9.48 *GB* for BSRTM for single shot migration. There was reduction of 17.19 *X* in memory

requirement from CSRTM to BSRTM. As the memory needed for migration fits in the low memory node's capacity, both the RTMs were run using low memory nodes and the computations were performed in-memory. The compute time observed in case of BSRTM is 8.38 *min* and in case of CSRTM 5.27 *min* which was 0.63*X* slower for a single shot migration, refer Table 4.

Conclusion

In this paper, we have demonstrated results and features of in-house developed RTM software suite SeisRTM. Two different implementations of RTM which were based on conventional and boundary wavefield saving approaches were described and discussed for their accuracy and compute resource requirements. The experiments were conducted using high resolution Marmousi2 model and field data. In both the cases, outcome were found comparable in terms of amplitudes and confirmed using a difference section of the two outcomes. It was observed that memory requirement of BSRTM is less than the CSRTM in synthetic and field data case. Whereas, the compute time depends on the size of source wavefield, if it is less than the node memory, CSRTM performs better than BSRTM as seen in synthetic data example with 768 GB node memory otherwise BSRTM performs better as in the case of 192 GB node memory. In case of field data, where source wavefield sizes were less than the node memory for both CSRTM and BSRTM, the former performs better.

It can be inferred that the higher memory nodes are preferred for CSRTM computations. Equivalent quality RTM outcome can be produced using BSRTM in a slightly higher compute time using less node memory. BSRTM can also be beneficial in case of 3D RTM, where the source wavefield size become larger than the node memory capacity. In future, we are getting ready to run 3D SeisRTM with BSRTM.

References

Bao D, N., and M. George A, 2015, Five ways to avoid storing source wavefield snapshots in 2D elastic prestack reverse time migration: GEOPHYSICS, **80**, S1–S18.

Clapp, R. G., 2009, Reverse time migration with random boundaries: SEG Technical Program Expanded Abstracts 2009, 2809–2813.

Liu, H., H. Liu, Z. Zhen, and C. Yong-Fu, 2010, Denoising and storage in seismic reverse time migration: Chinese Journal of Geophysics, **53**.

Londhe, A., R. Rastogi, A. Srivastava, K. Khonde, K. M. Sirasala, and K. Kharche, 2021, Adaptively accelerating FWM2DA seismic modelling program on multi-core CPU and GPU architectures: Computers & Geosciences, **146**, 104637.

Rastogi, R., A. Srivastava, M. Gawade, N. Mangalath, L. Bathula, B. Mahajan, and S. Phadke, 2022, 2D isotropic and vertical transversely isotropic RTM using SEG Hess VTI Model: Second International Meeting for Applied Geoscience & Energy, 2782–2786.

Rastogi, R., A. Srivastava, S. Phadke, B. Mahajan, L. Bathula, and S. Ghosh, 2023, Improved RTM imaging of marine streamer data using principle of reciprocity: European Association of Geoscientists & Engineers, **2023**, 1–5.

Tan, S., and L. Huang, 2014, Reducing the computer memory requirement for 3D reverse-time migration with a boundary-wavefield extrapolation method: GEOPHYSICS, **79**, S185–S194.

Tang, C., and D. Wang, 2012, Reverse Time Migration with source wavefield reconstruction in target imaging region.

Yang, P., J. Gao, and B. Wang, 2014, RTM using effective boundary saving: A staggered grid GPU implementation: Computers & Geosciences, **68**, 64–72.

Acknowledgments

Authors would like to thank Geo data Processing & Interpretation Centre (GEOPIC), ONGC, Dehradun for providing the field data for this study. Authors are also thankful to National Supercomputing Mission (NSM), MeitY, Government of India, for funding this research work, PARAM SHIVAY for HPC facility and Centre for Development of Advanced Computing (CDAC), Pune, for permission to publish this work. Authors would like to acknowledge the contribution of Dr. Suhas Phadke during development of SeisRTM software and Dr. Akshara Kaginalkar for her timely guidance during the techno-administrative support for the project.