



## Acoustic Least-Squares RTM using Devito

Rafael A. de Cristo\* - SENAI-CIMATEC, Peterson N. Santos - SENAI-CIMATEC, Reynam C. Pestana -CPGG/UFBA

Copyright 2021, SBGf - Sociedade Brasileira de Geofísica.

This paper was prepared for presentation at the 17<sup>th</sup> International Congress of the Brazilian Geophysical Society, held Online, Brazil, November 8-11, 2021.

Contents of this paper were reviewed by the Technical Committee of the 17<sup>th</sup> International Congress of The Brazilian Geophysical Society and do not necessarily represent any position of the SBGf, its officers or members. Electronic reproduction or storage of any part of this paper for commercial purposes without the written consent of The Brazilian Geophysical Society is prohibited.

### Abstract

**A good resolution of the subsurface image helps to reduce uncertainties, which will guide a good interpretation of the reservoir rocks, spotting possibly an oil & gas accumulation. In this context Least-Squares Reverse Time Migration ( LSRTM ) plays an important role in attenuating migration artifacts, helping to focus the reflectors and improve spatial resolution. In this work it has been used the LSRTM acoustic case using a state-of-art finite-difference (FD) code-generation, Devito, which allows to solve the wave equation, using symbolic expressions. The LSRTM using Devito has been performed in two 2D models, which show its particularities and difficulties. In one of the examples, a synthetic velocity model from the Búzios field was used to validate the LSRTM. Also, some piece of codes has been provided to show the way Devito handles with the wave equation and how easily it is coded.**

### Introduction

Reverse time migration (RTM) is a seismic imaging method to map the subsurface reflectivity using recorded seismic waveforms (Baysal et al. (1983)). Since the marine acquisition is made in a sparse grid, artifacts come out, misplacing the reflectors and leading to an incorrect interpretation of the reservoir rocks.

The first author to use the LSM method was Nemeth et al. (1999), over the Kirchhoff migration in order to overcome problems like truncated recording aperture, coarse source and/or receiver sampling.

In his famous paper Dai and Schuster (2013) implemented LSRTM, reducing acquisition footprint, RTM artifacts and improving the image resolution. LSRTM implementation uses a linearized wave equation using the Born approximation ( $p \approx p_0$ ), where the  $p_0$  is the background wavefield. Then the reflectivity model is updated iteratively combining a step-length and a search direction depending on the method ( gradient, conjugate gradient, L-BFGS).

In this work to solve the acoustic wave equation we used a state-of-art finite difference solver called Devito, which allows the user implement differential equations, in a symbolic way the same manner is written analytically. Devito, based on the equations, generates a c-optimized code making it way easier to implement than the other

conventional platforms.

In this scenario two 2D models have been used to validate the LSRTM, using the Born approximation for forward and adjoint modelling. The steepest descent method was chosen to update the RTM image with an appropriate step-length proposed by (Barzilai and Borwein (1988)). Also will be shown the fashion way Devito implements the acoustic wave equation will be discussed here how is generated the optimized c-code.

### Theory

The LSRTM method attempts to improve the subsurface image resolution iteratively. Since it is a linear inversion scheme, it should be made a linearized acoustic wave equation. It is simple done by setting a relation on the velocity model as  $c = c_0 + \delta c$  (de Oliveira et al. (2016)) which produces on the wavefield the result of  $p = p_0 + \delta p$ . In this work the approximation has been done on the squared slowness  $m = \frac{1}{v^2}$ .

The acoustic wave equation using the squared slowness considering a density constant can be written as:

$$m_0(\mathbf{x}) \frac{\partial^2 p_0(\mathbf{x}, t)}{\partial t^2} - \nabla^2 p_0(\mathbf{x}, t) = s(\mathbf{x}, t), \quad (1)$$

where  $s$  is the source,  $m$  is the squared slowness,  $\mathbf{x}$  is the vector position and  $p_0$  is the background wavefield.

A perturbation in the squared slowness of  $m = m_0 + \delta m$  produces a perturbation in the wavefield as  $p = p_0 + \delta p$  that obeys:

$$m(\mathbf{x}) \frac{\partial^2 p(\mathbf{x}, t)}{\partial t^2} - \nabla^2 p(\mathbf{x}, t) = s(\mathbf{x}, t), \quad (2)$$

using the perturbation of the squared slowness and the perturbation on the wavefield into (2):

$$(m_0(\mathbf{x}) + \delta m(\mathbf{x})) \frac{\partial^2 (p_0(\mathbf{x}, t) + \delta p(\mathbf{x}, t))}{\partial t^2} - \nabla^2 (p_0(\mathbf{x}, t) + \delta p(\mathbf{x}, t)) = s(\mathbf{x}, t). \quad (3)$$

Equation (3) can be divided into two set of equations:

$$\begin{cases} m_0 \frac{\partial^2 p_0(\mathbf{x}, t)}{\partial t^2} - \nabla^2 p_0(\mathbf{x}, t) = s(\mathbf{x}, t), \\ m_0 \frac{\partial^2 \delta p(\mathbf{x}, t)}{\partial t^2} - \nabla^2 \delta p(\mathbf{x}, t) = -\delta m \frac{\partial^2 (p_0(\mathbf{x}, t) + \delta p(\mathbf{x}, t))}{\partial t^2} \end{cases} \quad (4)$$

Considering the Born's approximation ( $p \approx p_0$ ) on the

equation (4)

$$\begin{cases} m_0 \frac{\partial^2 p_0(\mathbf{x}, t)}{\partial t^2} - \nabla^2 p_0(\mathbf{x}, t) = s(\mathbf{x}, t), \\ m_0 \frac{\partial^2 \delta p(\mathbf{x}, t)}{\partial t^2} - \nabla^2 \delta p(\mathbf{x}, t) = -\delta m \frac{\partial^2 p_0(\mathbf{x}, t)}{\partial t^2}. \end{cases} \quad (5)$$

where  $p_0$  is the wavefield from a smoothed squared slowness  $m_0$ , also known as background wavefield and  $\delta p$  is known as scattered wavefield. Equation (5) is for Born forward modelling, the adjoint modelling equations is defined as:

$$m_0(\mathbf{x}) \frac{\partial^2 q(\mathbf{x}, t)}{\partial t^2} - \nabla^2 q(\mathbf{x}, t) = \mathbf{d}, \quad (6)$$

where  $\mathbf{d}$  is the recorded data. Solving equation (6) and (5) the image can be constructed as:

$$\mathbf{I} = \sum_{shots} \sum_t \frac{\partial^2 p_0(\mathbf{x}, t)}{\partial t^2} q(\mathbf{x}, t). \quad (7)$$

Since LSRTM is solved iteratively, is needed to define a search direction and a step-length so that the migration image is updated as:

$$\mathbf{I}_{k+1} = \mathbf{I}_k - \alpha_k \mathbf{g}_k. \quad (8)$$

where  $\mathbf{g}_k$  is the gradient and  $\alpha_k$  is the step-length. The gradient computation is simply taking equation (5) and instead of injecting the shot recorded  $\mathbf{d}$ , injects the residue  $\mathbf{d}_{calc} - \mathbf{d}_{obs}$  (backward in time).

In this work the step-length for the first iteration was:

$$\alpha = \frac{0.05}{\max(\mathbf{g})}, \quad (9)$$

and for the other iterations, was used the approach of Barzilai and Borwein (1988):

$$\alpha_k^{BB1} = \frac{\mathbf{s}_{k-1}^T \mathbf{s}_{k-1}}{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}}, \quad (10)$$

where  $\mathbf{s}_{k-1} = \mathbf{m}_k - \mathbf{m}_{k-1}$  e  $\mathbf{y}_{k-1} = \mathbf{g}_k - \mathbf{g}_{k-1}$

A second option is:

$$\alpha_k^{BB2} = \frac{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}}{\mathbf{y}_{k-1}^T \mathbf{y}_{k-1}} \quad (11)$$

Afterwards an adaptive method based on Barzilai and Borwein (1988) has been proposed by Zhou et al. (2006), which is the step-length used in this work as follow,

$$\alpha_k = \begin{cases} \alpha_k^{BB2}, & \text{if } 0 < \frac{\alpha_k^{BB2}}{\alpha_k^{BB1}} < 1 \\ \alpha_k^{BB1}, & \text{else} \end{cases} \quad (12)$$

A good way to know if the result is going in the right direction is to evaluate the misfit function, if it decreases along with the iteration it says that the problem is converging to the minimum, providing the right solution. It is defined as the  $L_2$  norm of the observed data  $d_{obs}$  and the synthetic data  $d_{calc}$ .

$$E(\mathbf{m}) = \frac{1}{2} \|d_{obs} - d_{calc}\|^2 \quad (13)$$

## Finite-Difference solver : Devito

Devito (Louboutin et al. (2018)) is an open-source Python project based on domain-specific language and compiler technology. Driven by the requirements of rapid HPC applications development in exploration seismology, the language and compiler have evolved significantly since inception. Sophisticated boundary conditions, tensor contractions, sparse operations and features such as staggered grids and sub-domains are all supported; operators of essentially arbitrary complexity can be generated.

Devito utilises SymPy to allow the definition of operators from high-level symbolic equations and generates optimised and automatically tuned code specific to a given target architecture.

### Code generation using Devito

In this section some examples will be provided how easily Devito can write complicated partial differential equations (PDE), the corresponded c-code and how optimized it is solved.

Let's consider the wave equation in (1). On Devito it can be written as:

```
1 from devito import TimeFunction
2
3 # Define the wavefield with the size of the
4   model and the time dimension
5 u = TimeFunction(name="u", grid=model.grid,
6                 time_order=2, space_order=2)
7
8 # We can now write the PDE
9 pde = model.m * u.dt2 - u.laplace
10 pde
```

Listing 1: Acoustic wave equation with constant density in Devito

$$\text{Out}[6]: \quad -\frac{\partial^2}{\partial x^2} u(t, x, y) - \frac{\partial^2}{\partial y^2} u(t, x, y) + \frac{\partial^2}{\partial t^2} u(t, x, y) - \text{vp}^2(x, y)$$

Figure 1: Symbolic acoustic wave equation in Devito.

Basically it generates the symbolic expression, now the discretized way should be under equation & solve command as follow:

```
1
2 # This discrete PDE can be solved in a time-
3   marching way updating u(t+dt) from the
4   previous time step
5 # Devito as a shortcut for u(t+dt) which is u.
6   forward. We can then rewrite the PDE as
7 # a time marching updating equation known as a
8   stencil using customized SymPy functions
9 from devito import Eq, solve
10
11 stencil = Eq(u.forward, solve(pde, u.forward))
```

Listing 2: Discretized acoustic wave equation with constant density in Devito

To solve the wave equation, let's define the source and receiver using Devito.

```

1
2 # Finally we define the source injection and
3   receiver read function to generate the
4   corresponding code
5 src_term = src.inject(field=u.forward, expr=src
6   * dt**2 / model.m)
7
8 # Create interpolation expression for receivers
9 rec_term = rec.interpolate(expr=u.forward)

```

Listing 3: Source and receiver definition.

```

1
2 from devito import Operator
3
4 op = Operator([stencil] + src_term + rec_term,
5   subs=model.spacing_map)
6
7 op(time=time_range.num-1, dt=model.critical_dt)

```

Listing 4: Finally we solve acoustic wave equation using the operator command from Devito.

```

1 /* Begin section0 */
2   for (int x = x_m; x <= x_M; x += 1)
3   {
4     #pragma omp simd aligned(u,vp:64)
5     for (int y = y_m; y <= y_M; y += 1)
6     {
7       float r10 = dt*dt;
8       float r9 = vp[x + 2][y + 2]*vp[x + 2][y
9 + 2];
10      u[t2][x + 2][y + 2] = r10*r9*(1.0e-2F*(u
11 [t0][x + 1][y + 2] +
12      +u[t0][x + 2][y + 1] + u[t0][x + 2][y +
13 3] + u[t0][x + 3][y + 2]) - 3.99999991e-2F*u
14 [t0][x + 2][y + 2] + ((-2.0F*u[t0][x + 2][y
15 + 2] +
16      +u[t1][x + 2][y + 2])/r10)/r9);
17    }
18  }
19 /* End section0 */

```

Listing 5: Optimized C code for wavefield computation.

**NUMERICAL RESULTS**

The LSRTM using Devito has been tested in two different velocity models, each of them having its own acquisition parameters.

*Three layer velocity model*

On this example, the LSRTM has been tested using a three layer velocity model with a box in the middle as shown in the figure (2), the true and corresponded velocity model. The acquisition parameters are specified in the following table.

LSRTM parameters for the three layer model	
# of shots	21
# of receivers	101
Grid	10mx10m
$f_{peak}$	25Hz
Record length	1s
sample rate	1.7ms
Model extension	1kmx1km
LSRTM iterations	50

The initial migration has a low frequency backscattering noise and the reflectors are not well focused (Figure

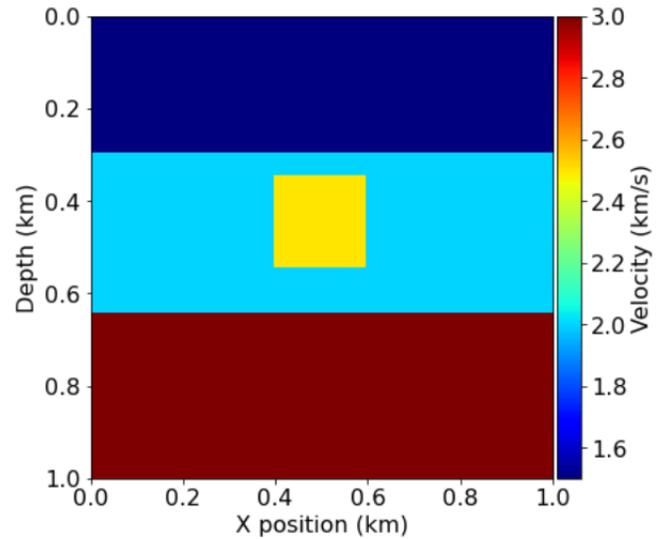


Figure 2: True velocity model.

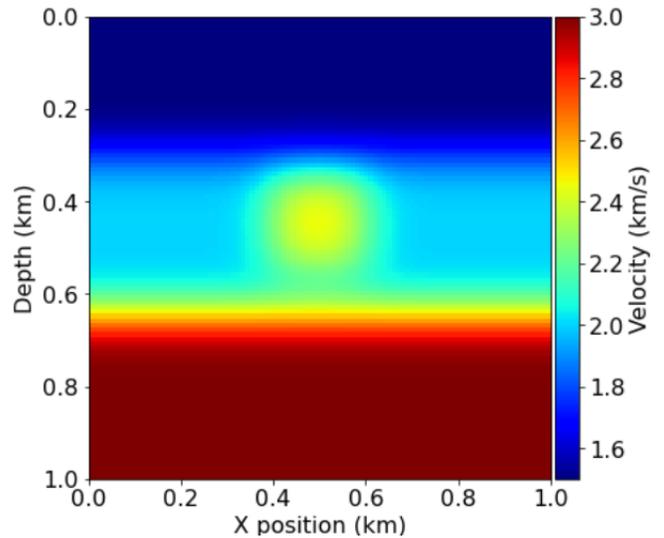


Figure 3: Smoothed velocity model to perform forward and adjoint Born modelling.

6). After 50 iterations of the LSRTM (Figure 5) the low-frequency noise has been very attenuated and the reflectors are more well focused.

A profile has been taken from the middle of the data (around 500 m in  $x$ ) to compare the LSRTM results against the true reflectivity and the initial migration. LSRTM profile fits better with the true reflectivity amplitude curve, clearly seen on the two reflectors at depth of 300m and 600m.

The misfit function image shows the convergence of the LSRTM implemented on Devito along the 50 iterations.

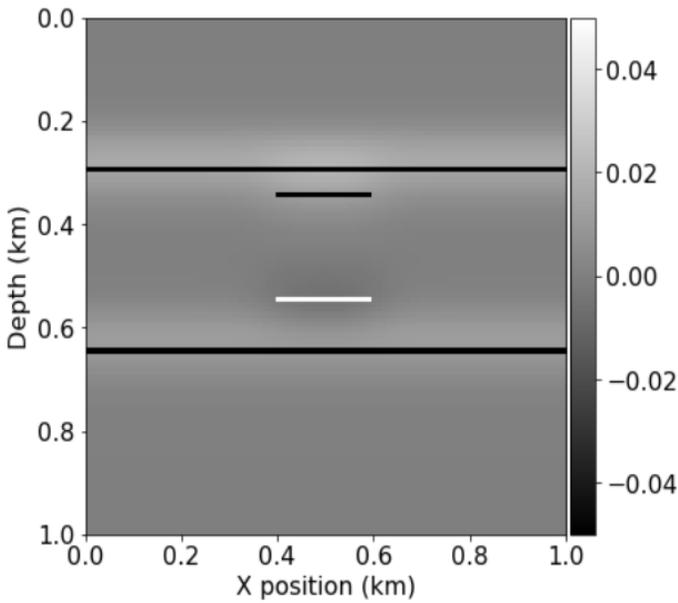


Figure 4: True reflectivity model is written as  $dm = m - m_0$

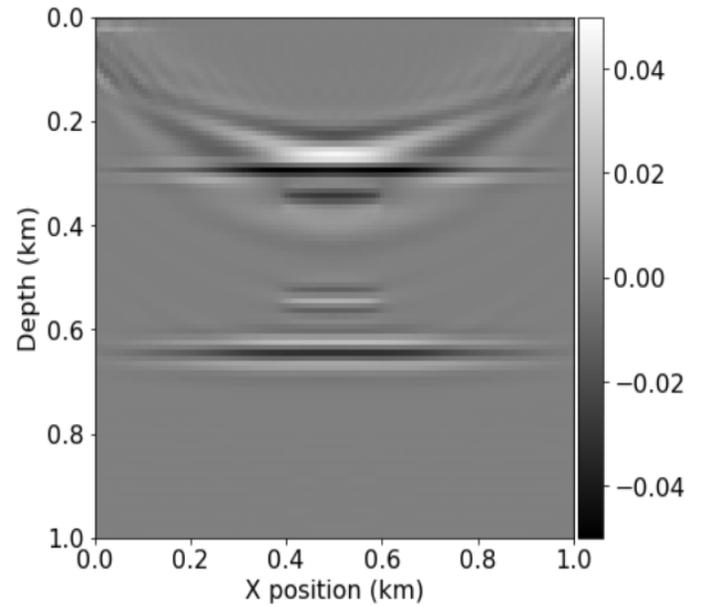


Figure 6: Initial migration. Low frequency backscattering noise is predominant close to the first reflector and the reflectors are not well focused.

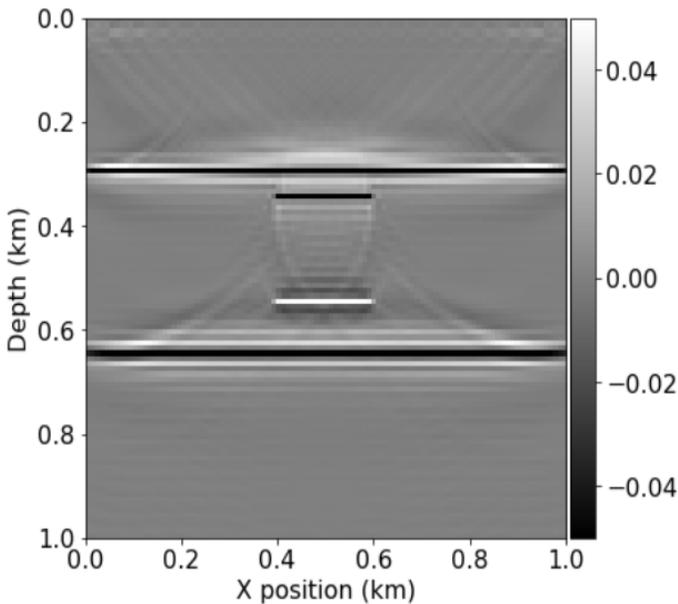


Figure 5: LSRTM result after 50 iterations. The backscattering noise has been severely attenuated and the reflectors are very well focused, being closer to the true reflectivity model.

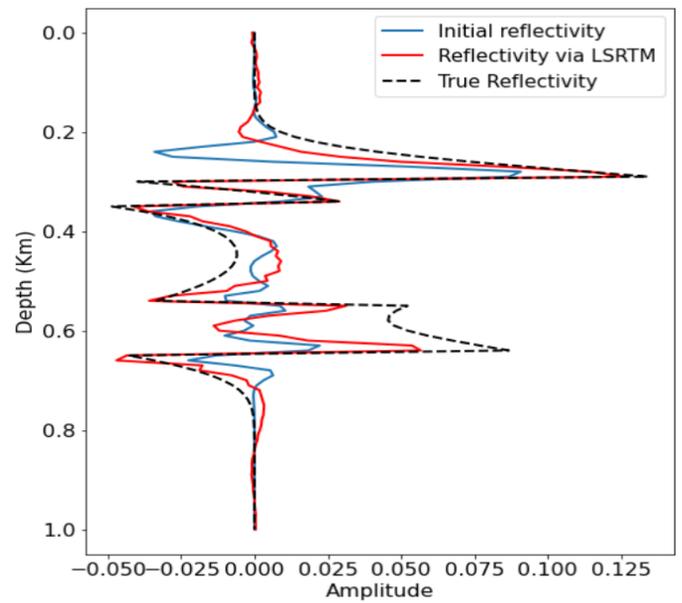


Figure 7: Reflectivity comparison: Initial, LSRTM and true reflectivity at  $x = 500m$ .

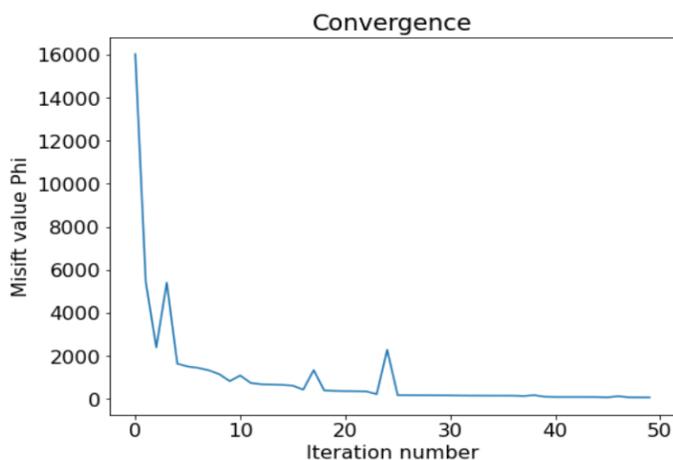


Figure 8: Misfit function along the LSRTM iterations.

*Búzios velocity model*

This velocity field is based on the real velocity field from the Búzios field, using well data information, sonic well information and seismic horizons (Karsou et al. (2019)). To make it even more close to the Búzios field velocity model some elements have been added as a stratified salt and an overhang.

Identifying a correctly top and base of salt is very important to obtain a realistic migration and also to spot correctly an oil gas accumulation. In this sense LSRTM will play an important role to delineate those elements in the Búzios velocity model. An overhang well imaged will help also to identify correctly the top of salt, making the image closer to the true earth's reflectivity.

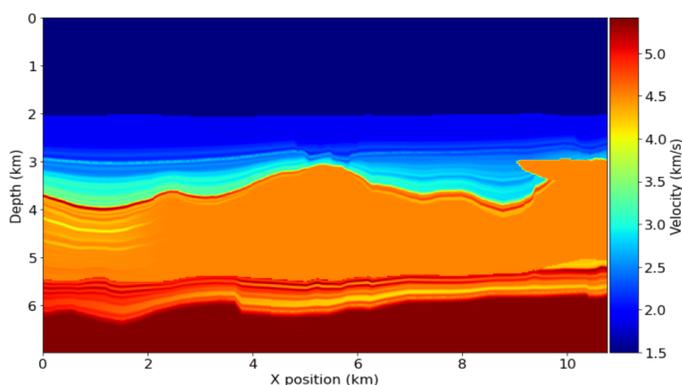


Figure 9: True velocity model created using real data set of the Búzios dataset field.

The LSRTM over the Búzios (Figure 13) model showed a very good improvement, removing the backscattering noise, improving the overhang, helping to focus the top of salt and the base of the salt.

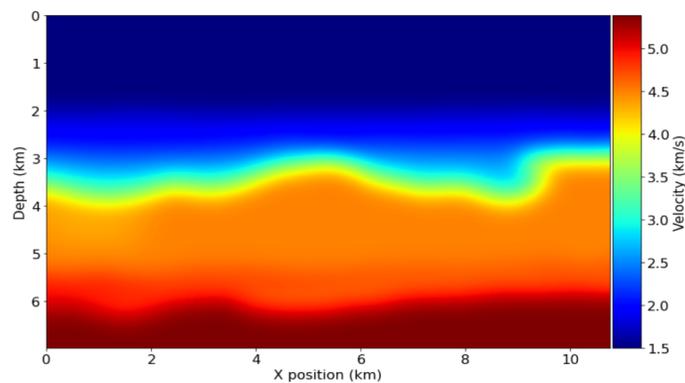


Figure 10: Smooth velocity model.

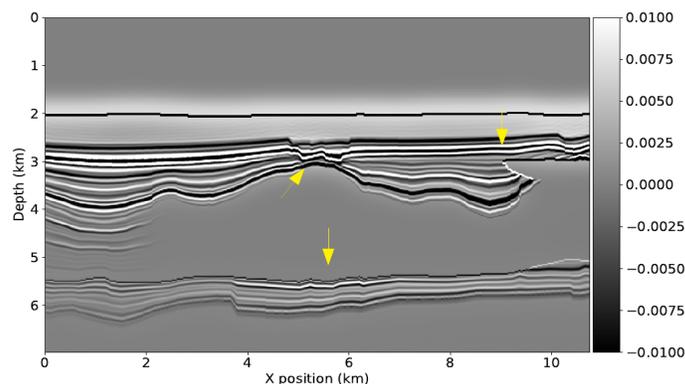


Figure 11: True reflectivity for Búzios velocity model.

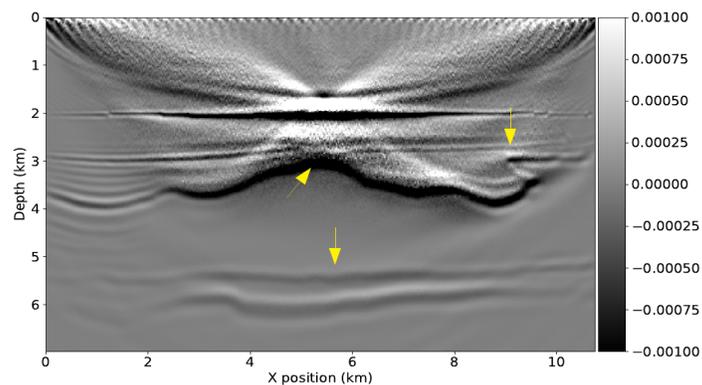


Figure 12: Initial migration of the Búzios velocity model. It is clearly seen that the backscattering noise is very strong and top of salt is totally blurred.

LSRTM parameters for the Búzios velocity model	
# of shots	51
# of receivers	431
Grid	25mx25m
$f_{peak}$	15Hz
Record length	8s
sample rate	1ms
Model extension	10.7kmx7km
LSRTM iterations	100

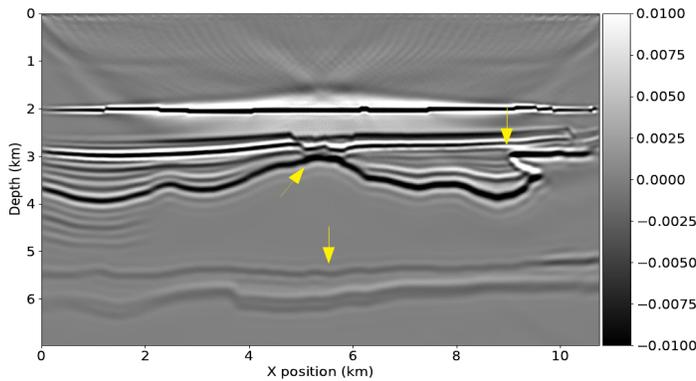


Figure 13: LSRTM after 100 iterations. Top of salt, overhang and base of salt has been improved after backscattering noise attenuation. Also base of salt has been more focused and top of salt as well.

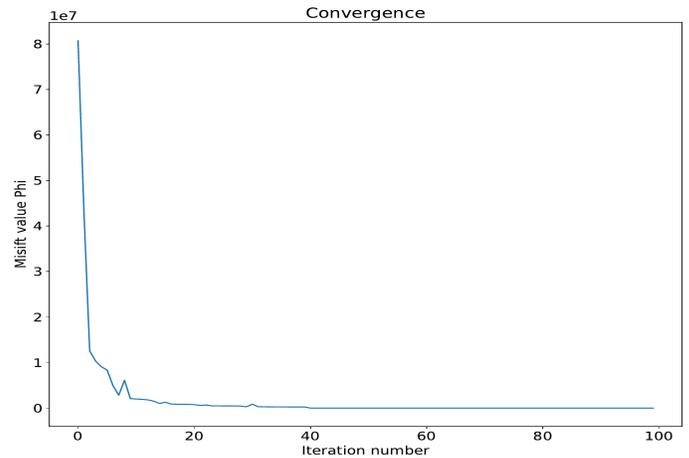


Figure 15: Misfit function along with the LSRTM iterations.

### Acknowledgements

This work was supported by CENPES/Petrobras and ANP through the DSL project at SENAI-CIMATEC. The authors also wish to thank to the developer team of the Devito community. A special acknowledgment to the geophysical team at SENAI-CIMATEC for discussions and contributions. Special thanks to the whole Devito development team and to GISIS to the Búzios velocity model.

### References

- Barzilai, J. and J. M. Borwein, 1988, Two-point step size gradient methods: *IMA journal of numerical analysis*, **8**, 141–148.
- Baysal, E., D. D. Kosloff, and J. W. Sherwood, 1983, Reverse time migration: *Geophysics*, **48**, 1514–1524.
- Dai, W. and G. T. Schuster, 2013, Plane-wave least-squares reverse-time migration: *Geophysics*, **78**, S165–S177.
- de Oliveira, A. E. L., R. da Cruz Pestana, and A. W. G. dos Santos, 2016, Least-square reverse time migration (lsrtm) in the shot domain: *Brazilian Journal of Geophysics*, **34**, 277–287.
- Karsou, A., L. Andrade, D. Rodrigues, R. Silva, L. Gadioli, F. Costa, M. Cetale Santos, W. Lupinacci, and D. Filho, 2019, Construction of a velocity model of the brazilian pre salt based on búzios field - preliminary results: , 1–5.
- Louboutin, M., M. Lange, F. Luporini, N. Kukreja, P. Witte, F. Herrmann, P. Ve-lesko, and G. Gorman, 2018, Devito: An embedded domain-specific language for finite differences and geophysical exploration: *Arxiv preprints: arXiv preprint arXiv:1808.01995*.
- Nemeth, T., C. Wu, and G. T. Schuster, 1999, Least-squares migration of incomplete reflection data: *Geophysics*, **64**, 208–221.
- Zhou, B., L. Gao, and Y.-H. Dai, 2006, Gradient methods with adaptive step-sizes: *Computational Optimization and Applications*, **35**, 69–86.

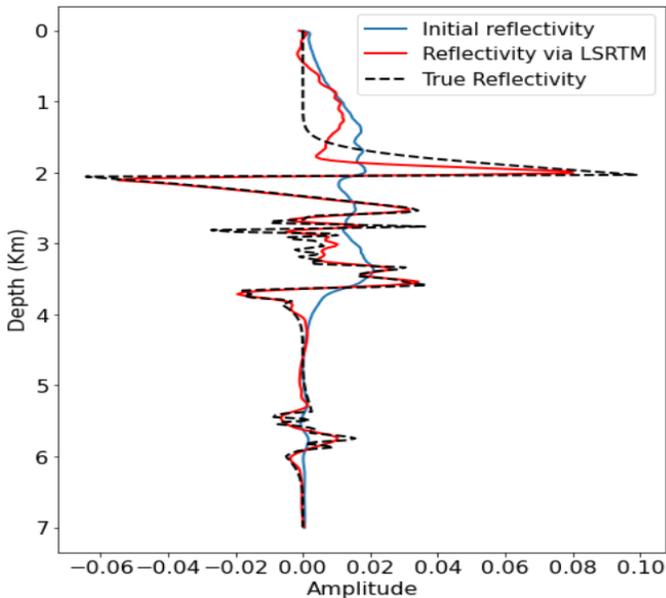


Figure 14: Reflectivity comparison: Initial, LSRTM and true reflectivity at  $x = 7Km$ .

### Conclusions

The LSRTM has been implemented using a state-of-art FD solver, Devito, where the wave equation can be easily implemented in a symbolic way, using a few lines of code.

Two velocity models were used to validate the results. The first one after the LSRTM the backscattering noise present on the two initial migrations has been removed and the LSRTM helped to improve the focusing of the reflectors. Another velocity model, based on the real Búzios velocity model, has been used and the LSRTM helped to remove the backscattering noise, to focus the reflectors and match the amplitude range. Also some key elements have been improved as base of salt, overhang and the top of salt, helping to delineate the salt and the pre-salt where the reservoirs in Brazil basins are located.