ARTICLE INFO:
Received : April 13, 2018
Revised : July 12, 2018
Accepted : October 17, 2018
CT&F - Ciencia, Tecnologia y Futuro Vol 8, Num 2 Dec. 2018. pages 89 - 98
DOI : https://doi.org/10.29047/01225383.85

ctyf@ecopetrol.com.co

ACCELERATED 2D FWI USING THE SYMMETRY ON INNER PRODUCT SPACES

ACELERACIÓN DE LA INVERSIÓN DE ONDA COMPLETA 2D USANDO LA SIMETRÍA DE LOS ESPACIOS FORMADOS POR PRODUCTOS INTERNOS.

Noriega, Reynaldo-Fabian[a]; Abreo, Sergio-Alberto[a]; Ramirez, Ana-B.[a]*

ABSTRACT
Full Waveform Inversion (FWI) is a common technique used in the oil and gas industry due to its capabilities to estimate subsurface characteristics such as material's density and sound velocity with high resolution. The 2D time domain FWI method involves the modeling of the forward wavefield of the source and the backpropagated field of the difference between the modeled and observed data. Therefore, due to its high computational cost in terms of RAM consumption and execution time, the High Performance Computing (HPC) field is very useful to deal with these problems. There are computational state-of-the-art solutions that allow to increase the execution time such as the parallel programming paradigm that involves the use of multicore processor systems. Furthermore, there are mathematical solutions leveraging on the properties of the algorithm used that make it possible to enhance performance of the method. We propose in this paper a new way to compute the FWI gradient, by taking advantage of an inner product property. Additionally, a computational strategy is combined with this proposal in the inversion scheme, thus improving FWI performance.

RESUMEN
La Inversión de Onda Completa (FWI, por sus siglas en inglés) es una técnica común en la industria de los hidrocarburos debido a la capacidad de generar perfiles de alta resolución de las características del subsuelo como densidad y velocidad. La FWI 2D en el dominio del tiempo implica el modelado del campo de presión generado por la fuente y del campo de presión generado por la diferencia entre los datos adquiridos y los datos modelados. Debido a su alto costo computacional en términos de consumo de memoria y el tiempo de ejecución, el área de la computación de alto desempeño (HPC, por sus siglas en inglés) se vuelve útil y necesario para lidiar con estos problemas. En el estado-del-arte existen estrategias computacionales que permiten incrementar el tiempo de ejecución de los algoritmos como el paradigma de la programación en paralelo, en el cual se hace uso de sistemas con procesadores multinúcleo. Por otra parte, también se puede aprovechar propiedades del algoritmo mediante desarrollos matemáticos lo cual impacta positivamente al momento de la implementación. En este trabajo se propone una nueva forma de calcular el gradiente de la FWI aprovechando una propiedad de los espacios producidos por productos internos. Adicionalmente, este planteamiento se combina con una estrategia de implementación para el manejo de memoria RAM en el esquema de inversión, incrementando su desempeño computacional.

KEYWORDS / PALABRAS CLAVE

FWI | Inner product | Computational strategy | RAM Consumption
FWI | Producto interno | Estrategia computacional | Consumo de RAM

AFFILIATION

[a]Universidad Industrial de Santander,
carrera 27 calle 9, C.P 680002, Bucaramanga, Colombia
*email: fabiian.noriiega@gmail.com

1 INTRODUCTION

In the seismic exploration industry, it is necessary to develop techniques to find subsurface characteristics of complex geological areas such as diving wave tomography, Reverse Time migration (RTM), and Full Waveform Inversion (FWI). Full Waveform Inversion Tarantola [1] has been recently used to estimate high resolution velocity models of the subsurface. The 2D inversion process involves the modeling of two pressure wavefields, per source used, to obtain the gradient data required to update the velocity model at each iteration. Therefore, due to its high computational cost (in time domain simulation), it is necessary to find strategies to deal with the huge amount of data obtained from the seismic modeling and the execution time involved in the process.

In the last decade, High Performance Computing (HPC) technologies have grown exponentially to handle thousands of millions of mathematical operations per second. In HPC, the most common method used to reduce the execution time is implementing algorithms by using numerous CPU cores and multicore devices such as Graphic Processing Units (GPUs). This is possible due to the parallel programming paradigm. However, in terms of the FWI algorithm, the pressure wavefield volumes represent a large amount of information and memory use. Hence, memory management strategies are useful to minimize this constraint.

The next section describes general aspects of the FWI method used in this work: the wave equation solution and the gradient estimation Plessix [2], and our redefinition of the gradient computation based on an inner product property. In section three, there is a description of some state-of-the-art computational strategies available and the usefulness of our proposed way to compute the gradient when using the wavefield reconstruction strategy proposed by Noriega, Ramirez, Abreo and Arce [3]. Finally, the experimental environment with its respective analysis are described and the conclusions are shown.

2. THEORETICAL FRAME

Full Waveform Inversion is a non-linear inversion method that iteratively estimates subsurface characteristics such as seismic velocity or density. These parameters are updated until the cost function reaches an adequate value.

Usually, the cost function is defined as

$$\phi(v) = \frac{1}{2}||mod - obs||_2^2 \quad (1)$$

where $v \in R^{N_x x N_z}$ (for the 2D case) is the velocity model, *obs* is the acquired data and mod is the modeled data. Nx and Nz are the number of elements on the model in the x and z coordinates.

The velocity model can be updated in iteration *k+1* using the first two terms of the Taylor series around a previous iteration k velocity model v^k as

$$v^{k+1} = v^k - \alpha \cdot [H(v^k)]^{-1} g(v^k) \quad (2)$$

where $g(v^k)$ represents the gradient of the misfit function, $H(v^k)$ represents the *Hessian matrix*, both evaluated at v^k and α is a scalar factor.

L-BFGS method was defined by Liu and Nocedal [4] to calculate an approximation of the product between the inverse of the *Hessian* matrix and the gradient $([H(v^k)]^{-1} \cdot g(v^k))$. This approximation uses the last m gradients and velocity models to compute the step forward and requires at least two gradients and two velocity models to obtain the search direction *r*.

Table 1 shows the pseudocode for this method, where $s_k = v_{k+1} - v_k$, $y_k = g_{k+1} - g_k$ and $\sigma_k = 1/y_k^t s_k$. The matrix D_k^0 is approximated by a diagonal $D_k^0 = \gamma_k I$, with

$$\gamma_k = \frac{s_{k-1}^t y_{k-1}}{y_{k-1}^t y_{k-1}} \quad (3)$$

Table 1. L-BFGS Algorithm pseudocode. The search direction r is the product between the inverse of the Hessian matrix and the gradient.

```
q ← gk                                   save gradient
for i=k-1,k - 2,…k-m do                  m, gradients history
                  ϵi ← σisIq
                  q ← q-ϵiyi
end for
r ← D0kq
for i = k - m,k - m+1,…, k - 1 do
                     β  ← σiyIr;
                  r ← r + si (ϵi - βi)
end for
```

Plessix [2] proposed a way to compute the velocity gradient by using the first order adjoint state method, as shown in the mathematical expression below

$$g(v^k) = \sum_s \frac{-2}{v(x,z)^3} \int_0^T \frac{\partial^2 P(x,z,t)}{\partial t^2} \lambda(x,z,T-t) dt \quad (4)$$

where *P* is the forward wavefield generated by the source, λ is the backpropagated field of the residual data (the difference between the modeled and acquired data) and s varies according to the number of sources.

The integral operation in the Equation 4 can be written as the inner product between the forward and backpropagated wavefields as

$$g_v = \frac{-2}{v^3} < \frac{\partial^2 P}{\partial t^2}, \lambda > \quad (5)$$

Similarly, for the density information, the gradient can be determined as

$$g_\rho = \frac{-2}{\rho^2} < \frac{\partial^2 P}{\partial t^2}, \lambda > \quad (6)$$

The core of the FWI is the modeling of the wave equation used to obtain the simulated data and the respective adjoint equation and operator used to obtain the inversion gradient. The isotropic acoustic wave equation with variable density is used in this work and the mathematical expressions for the forward modeling and backward modeling are

$$\frac{\partial^2 P}{\partial t^2} = \rho v^2 \left[\frac{\partial}{\partial x}\left(\frac{1}{\rho}\frac{\partial P}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{1}{\rho}\frac{\partial P}{\partial y}\right)\right] \quad (7)$$

and

$$\frac{\partial^2 \lambda}{\partial t^2} = v^2 \left[\frac{\partial}{\partial x}\left(\frac{1}{\rho}\frac{\partial(\lambda\cdot\rho)}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{1}{\rho}\frac{\partial(\lambda\cdot\rho)}{\partial y}\right)\right] \quad (8)$$

respectively, where ρ is the seismic density. Therefore, a joint inversion with both variables, velocity and density is viable.

The inner product exists as a generalization of the dot product applied to vector spaces and it has the same properties as commutative or multiplication by a scalar factor. Its notation is <.,.>. Additionally, there are different types of linear maps between inner product spaces. The one of interest in this work involves a linear operator as follows:

Being A any linear and symmetric operator and the inner space *V*, then:

$$< Ax, y > = < x, A^* y > \quad (9)$$

for all x, y ∈ *V*.

For the second order time derivative wave equation used in this work, the term A is self-adjoint in Equation 9 Bleistein [5]. However, when using the first order time derivative velocity-stress formulation, A=∂/(∂t) and A*= - ∂/(∂t).

Now, it is possible to redefine the gradient computation as

$$g_v = \frac{-2}{v^3} < P, \frac{\partial^2 \lambda}{\partial t^2} > \quad (10)$$

and

$$g_\rho = \frac{-1}{\rho^2} < P, \frac{\partial^2 \lambda}{\partial t^2} > \quad (11)$$

where g_v and g_ρ are the velocity and density gradients, respectively.

HIGH PERFORMANCE COMPUTING IMPLEMENTATION STRATEGIES

The wavefield modeling step, in the *FWI*, has a computational complexity of O(N^3) for the 2D case, which represents significant time when increasing the size of the dataset. In terms of RAM consumption, this wavefield volume can use up to hundreds of *Gigabytes* of memory space. Therefore, there are computational strategies to deal with these problems.

HPC is the field focused on aggregating computing power in a way that the execution performance is much higher than when working with common desktop computers. Therefore, it is possible to reduce the execution time, if so allowed by the algorithm, a couple orders of magnitude van Meel et. Al [6].

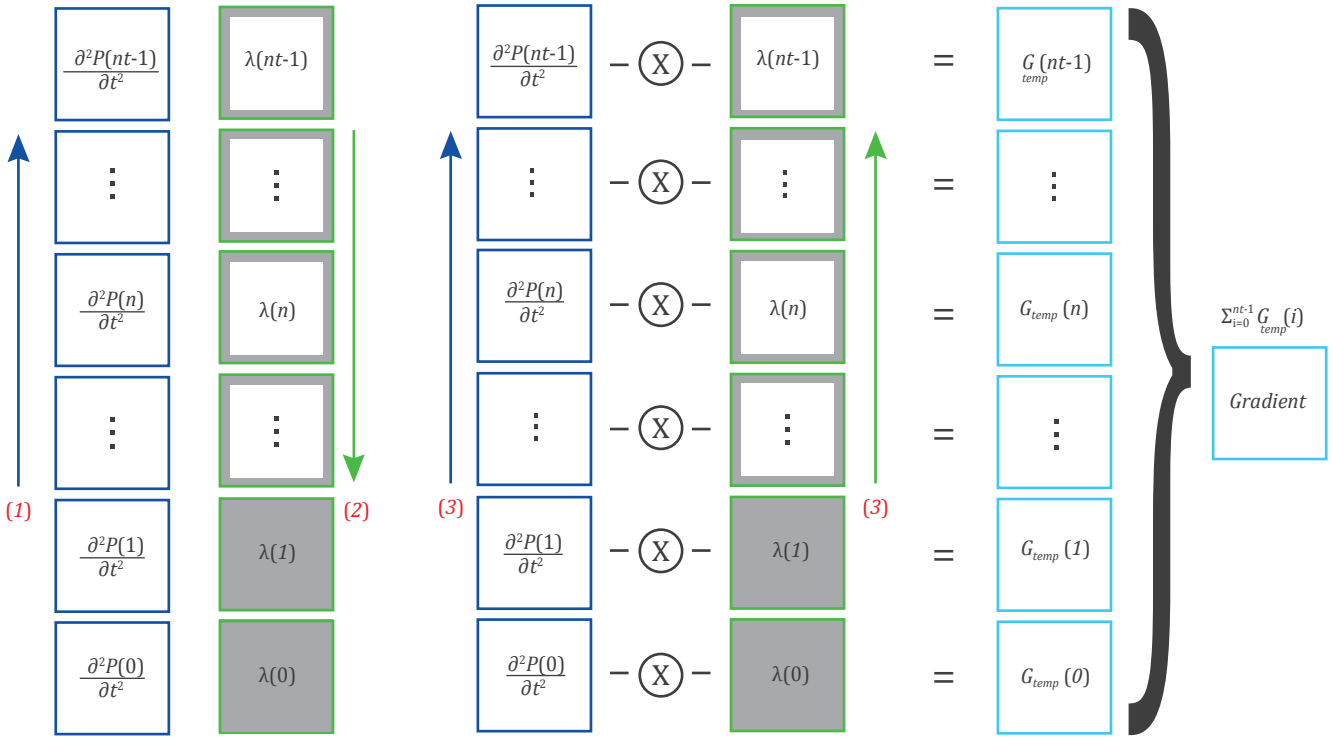


Figure 1. Wavefield reconstruction strategy. First, the forward modeling is performed without saving the pressure wavefield; then, the information at the boundaries of the backward wavefield slides is saved in RAM. Finally, the forward wavefield is recomputed and the backward wavefield is reconstructed while the gradient is calculated. Red numbers are the order in which each modeling is performed. Color arrows indicate the order in which the snapshots of each wavefield are calculated.

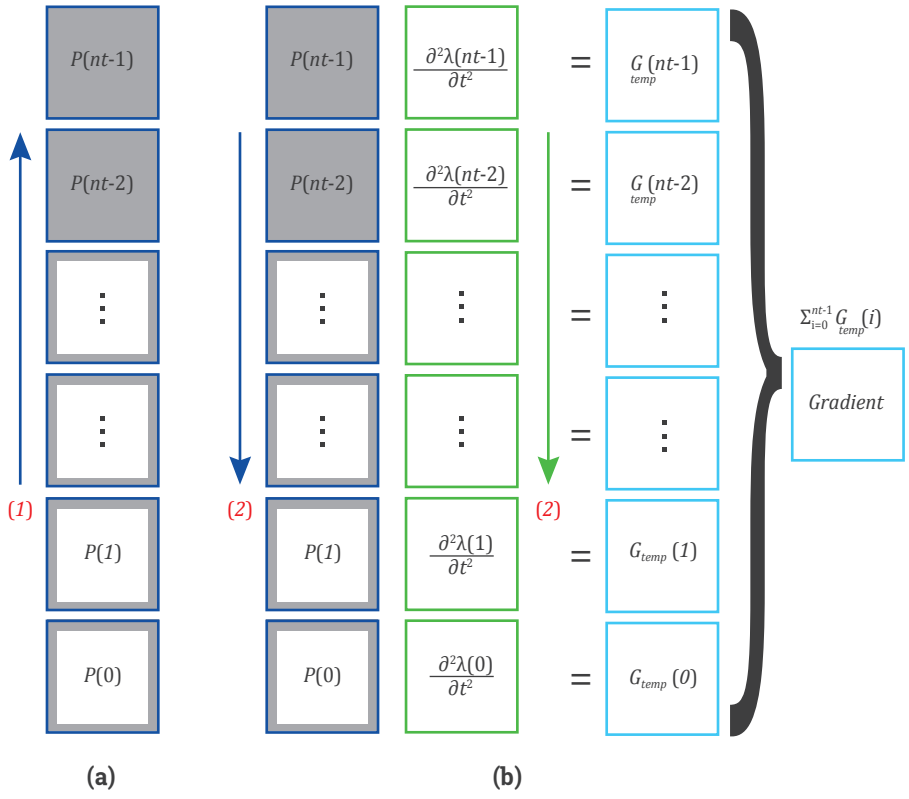


Figure 2. Gradient computation by taking advantage of the inner product property and the wavefield reconstruction strategy. On (a) forward wavefield, boundaries are saved on RAM. On (b) forward wavefield is reconstructed while backward wavefield and gradient are computed. Color arrows indicates the order the snapshots are computed, and red numbers mean the order each modeling is performed.

On the other hand, there are implementation strategies to manage RAM consumption. The Checkpointing strategy showed in Imbert et. al. [7] consists on saving a time snapshot (checkpoint) of the wavefield volume each n time steps. Then, the time slides are recomputed between checkpoints, when needed. Thus, the strategy could save up to 90% of memory consumption in the modeling process. Nevertheless, the execution time will increase exponentially making the strategy worthless for big data sizes.

The wavefield reconstruction strategy Noriega, Ramirez, Abreo and Arce [3], Yang et. Al. [8] (see **Figure 1**) consists of reconstructing the backward wavefield from the information in the boundaries of the area of interest. It is so that memory consumption can be reduced to less than 5% of the original value but it increases the execution time by a factor of nearly 1.55. The execution time increment is a consequence of an additional forward modeling because the numerical reconstruction of the time derivative of the pressure wavefield becomes unstable.

However, the gradient computation proposed in this work (Equations 10 and 11), makes it possible to reconstruct the forward wavefield and apply the time derivative to the backward wavefield as illustrated in **Figure 2**.

3. EXPERIMENT DESCRIPTION

A Canadian overthrust synthetic dataset was created for the CSEG paper by Gray & Marfurt in 1995 (see **Figure 3**). The model is composed of velocity gradients, faults, a complex topography, and mixed areas of low and high velocities. The velocity model was widely circulated among Canadian contractor companies in the mid 1990s. As regards density data, it was necessary to use the Lindseth relation Quijada and Stewart [9] to generate the original density model. The initial velocity model of the inversion process is a smoothed version of the original model (see **Figure 4**).

The ricker wavelet (see **Figure 5**) was used to generate both, the observed data and the modeled data at each FWI iteration. The mathematical expression used to obtain the source is

$$S = (1 - 2\pi^2 f^2 (t - t_0)^2) \; e^{-\pi^2 f^2 (t-t_0)^2} \tag{12}$$

where f is the central frequency, t is the time window and t_0 is a time delay. Sources were shoted from a flat surface corresponding to the upper limit of the model.

A multiscale frequency approach was used to obtain better results Mao et. al. [10]. The frequency sweep process consists on start by using a low central frequency wavelet (f =3[Hz] for our experiments) to model the pressure wavefield, run the FWI and obtain a velocity model. Then, this final model is used as the initial guess for a new inversion process where the central frequency is higher and so on until covering the desired frequency bandwidth.

As regards implementation, the Finite Difference in Time Domain method (FDTD) is used to discretize the wave equation. The time and spatial discretization is second order and eight order stencil, respectively, using the staggered grid method. Then, the mathematical expressions 7 and 8 can be expressed as

$$\frac{\delta^2 P}{\delta t^2} = \rho v^2 \left[\frac{\delta_+}{\delta_x} \left(\frac{1}{\rho} \frac{\delta_- P}{\delta x} \right) + \frac{\delta_+}{\delta_y} \left(\frac{1}{\rho} \frac{\delta_- P}{\delta y} \right) \right] \tag{13}$$

and

$$\frac{\delta^2 \lambda}{\delta t^2} = v^2 \left[\frac{\delta_+}{\delta_x} \left(\frac{1}{\rho} \frac{\delta_- (\lambda \cdot \rho)}{\delta x} \right) + \frac{\delta_+}{\delta_y} \left(\frac{1}{\rho} \frac{\delta_- (\lambda \cdot \rho)}{\delta y} \right) \right] \tag{14}$$

respectively, being

$$\frac{\delta_+}{\delta_x} u\left(i + \frac{1}{2}\right) \approx \sum_{j=0}^{n-1} c_j [u(i+1+j) - u(i-j)] \tag{15}$$

and

$$\frac{\delta_-}{\delta_x} u\left(i-\frac{1}{2}\right) \approx \sum_{j=0}^{n-1} c_j [u(i+j) - u(i+1-j)] \quad (16)$$

where c_j are the first derivative coefficients of the FDTD stencil.

CUDA C programming language is used to take advantage of the computational power of GPUs, by implementing the wavefield modeling with the parallel programming paradigm. Each time snapshot is computed on the GPU, reducing the execution time in comparison to a serial implementation.

The Convolutional Perfectly Matched Layer (CPML) Pasalic et. al. [11] is a well-known boundary condition used to simulate the extension of the FDTD lattice to infinity and it is used as the non-natural boundary in our implementation (see **Figure 3**).

In terms of RAM consumption, the equation to calculate the amount of memory used (in Mebibytes where 1 Mebibyte=2^{20} bytes. Megabyte =10^6 bytes) by this implementation, when there is no reconstruction strategy, is given by expression 17.

$$RAM = \frac{(2 \cdot N_x \cdot N_z \cdot N_t + 22 \cdot N_x \cdot N_z + 2 * N_r \cdot N_t + LBFGS) \cdot 4}{2^{20}} \quad (17)$$

where $N_x \cdot N_z \cdot N_t$ is the size of each, forward and backward wavefield. Additionally, there are other variables, with the same size of the velocity and density models, necessary to compute the pressure wavefields, the information related to boundary conditions, etc. The $N_r \cdot N_t$ term is the space used by each, the modeled and observed data per source used in the process, where N_r is the number of receivers and the term LBFGS is the memory used by the variables associated to the L-BFGS method with a value of $43x2^{17}$ for this work.

The pressure wavefields P and λ are the most expensive variables and that is why the reconstruction strategy is necessary when the model dimensions increase and the available RAM on the GPU is not enough to hold all the data.

Expression 18 is used to compute the RAM consumption when the reconstruction strategy is applied to the implementation.

$$RAM = \frac{(2 \cdot 4 \cdot N_x \cdot N_t + 2 \cdot 4 \cdot N_z \cdot N_t + 22 \cdot N_x \cdot N_z + 2 * N_r \cdot N_t + LBFGS) \cdot 4}{2^{20}} \quad (18)$$

where the term $4 \cdot N_x \cdot N_t$ is the space used by the information saved from each, the top and bottom boundaries of the area of interest each time slide of the pressure wavefield, and $4 \cdot N_z \cdot N_t$ is the space used by the information saved from each, the left and right boundaries of the area of interest for each time slide of the pressure wavefield.

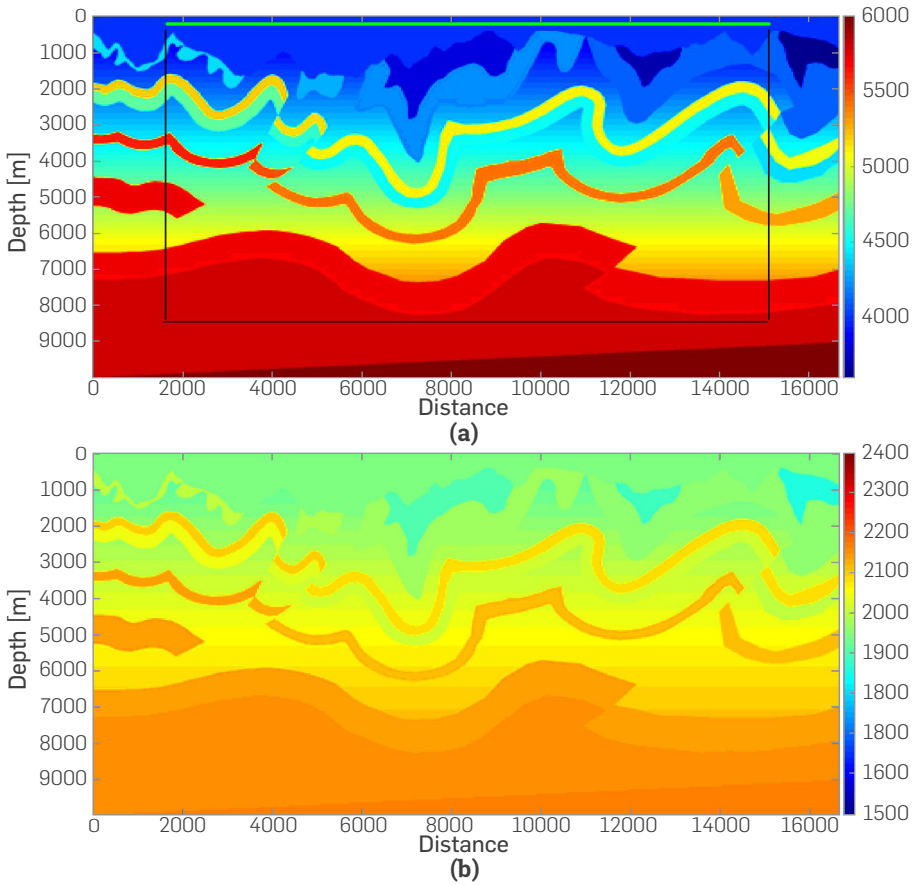


Figure 3. Canadian Foothills velocity model. (a) Original velocity model. The area outside the black lines represents the CPML zone. The top gray line represents the position of the receivers for each source used in all the experiments (each pixel is a receiver). The magenta dots represent the position for all the sources used in all the experiments. (b) Original density model used in combination with the original velocity model to generate the observed data.

If the model dimensions increase so that the RAM consumption exceeds the GPU capacity (e.g. when the implementation is used to process real seismic data volumes), the same expression can be applied to the total system memory (GPU RAM and the memory accessed by the CPU) where it is necessary to add the penalties of transferring data from GPU RAM to CPU RAM and vice versa.

The cluster used to run the experiments consists of 3 nodes with two Intel Xeon E-2620 v3 CPU, two Nvidia Tesla K40 GPU and 256 GB of ECC RAM per node. The Nvidia Cuda Compiler driver version 6.5 is used. The GNU Compiler Collection version used is 4.8.4 on Linux 3.16 - Debian Jessie.

The implementation process is focused on distributing the number of gradients per iteration (directly related to the number of sources used in the process) over all the computing nodes and improving performance, in terms of RAM consumption, per node. Thus, if more computing hardware is added to the system, the speedup factor would increase (more computing nodes will process more source wavefields at a time).

Other important parameters of the experiments are shown in **Table 2**.

Table 2. Main parameters of FWI used to run the experiments.

Parameter	Description	Value
N_x	Model Distance	417 [pt]
N_z	Model Depth	250 [pt]
Δh	Spatial step	40 [m]
Δt	Time step	1 [ms]
T_{end}	Acquisition time	5.0 [s]
N_s	Number of sources	51
N_{rec}	Number of receivers	337
W_{cpml}	Width of non-natural boundaries	40 [pt]
N_{frec}	Number of frequency steps	3 (3, 6 and 9 Hz)

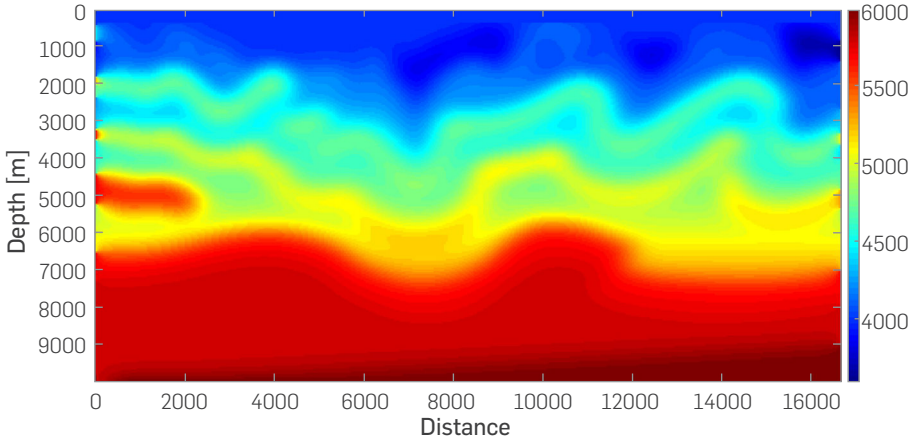


Figure 4. Initial velocity model for the multiscale process. The matrix was obtained by applying an average filter 150 times over the original velocity model.

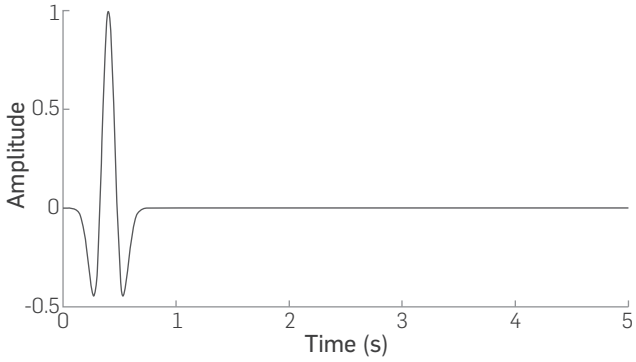


Figure 5. Ricker wavelet with a central frequency of 3 [Hz].

4. RESULTS

Thirty-five iterations were carried out per frequency step. **Figures 6, 7** and **8** show the final velocity model for f=3[Hz], f=6[Hz] and f=9[Hz], respectively.

The experiment comparison is focused on whether the wavefield reconstruction strategy is or is not used and whether the traditional way to compute the gradient or our approach is implemented. The RECO word means that the wavefield reconstruction strategy was used. The term inside the "<>" means whether the traditional way to compute the gradient is used or not.

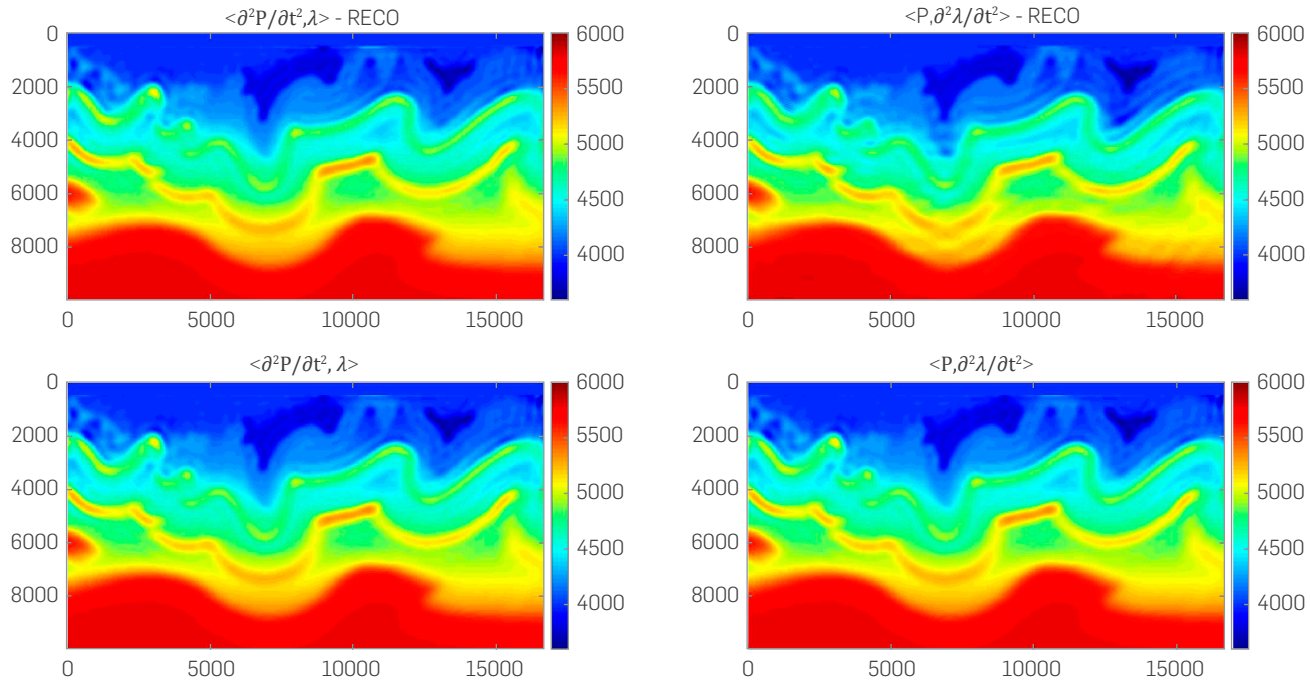


Figure 6. Final velocity model of each experiment with a central frequency of 3 [Hz].

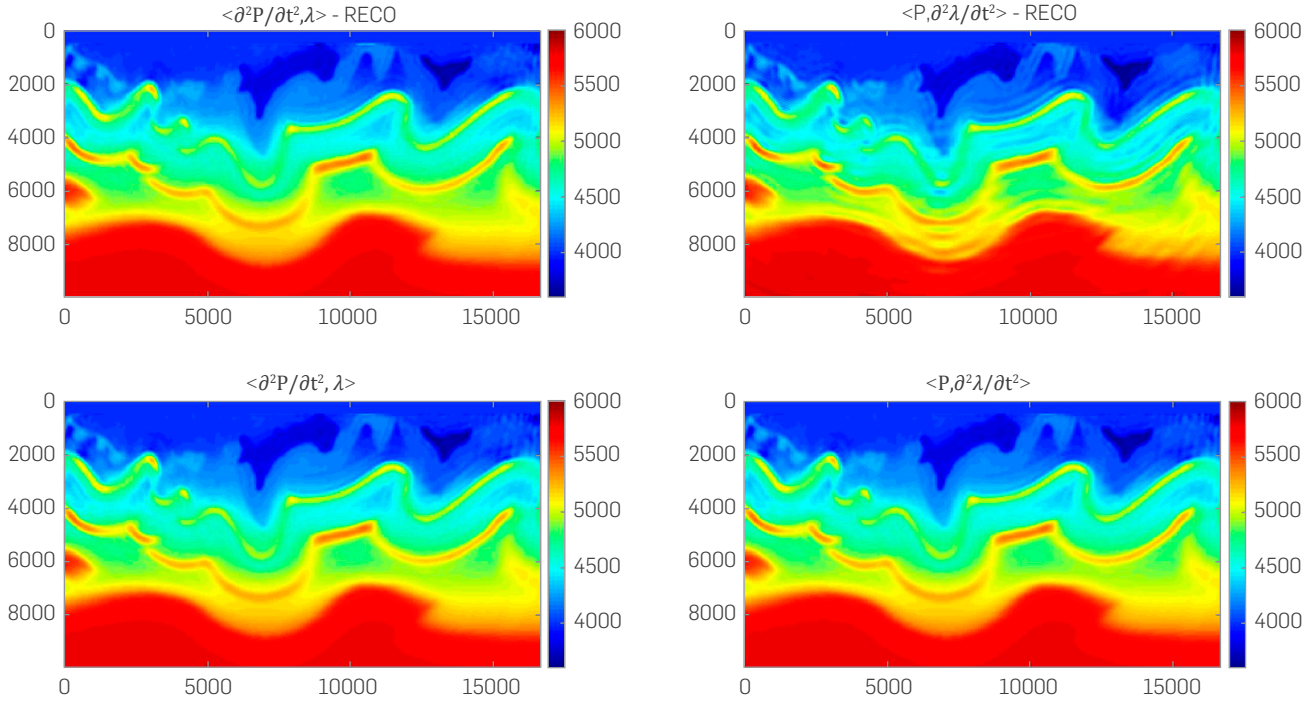


Figure 7. Final velocity model of each experiment with a central frequency of 6 [Hz].

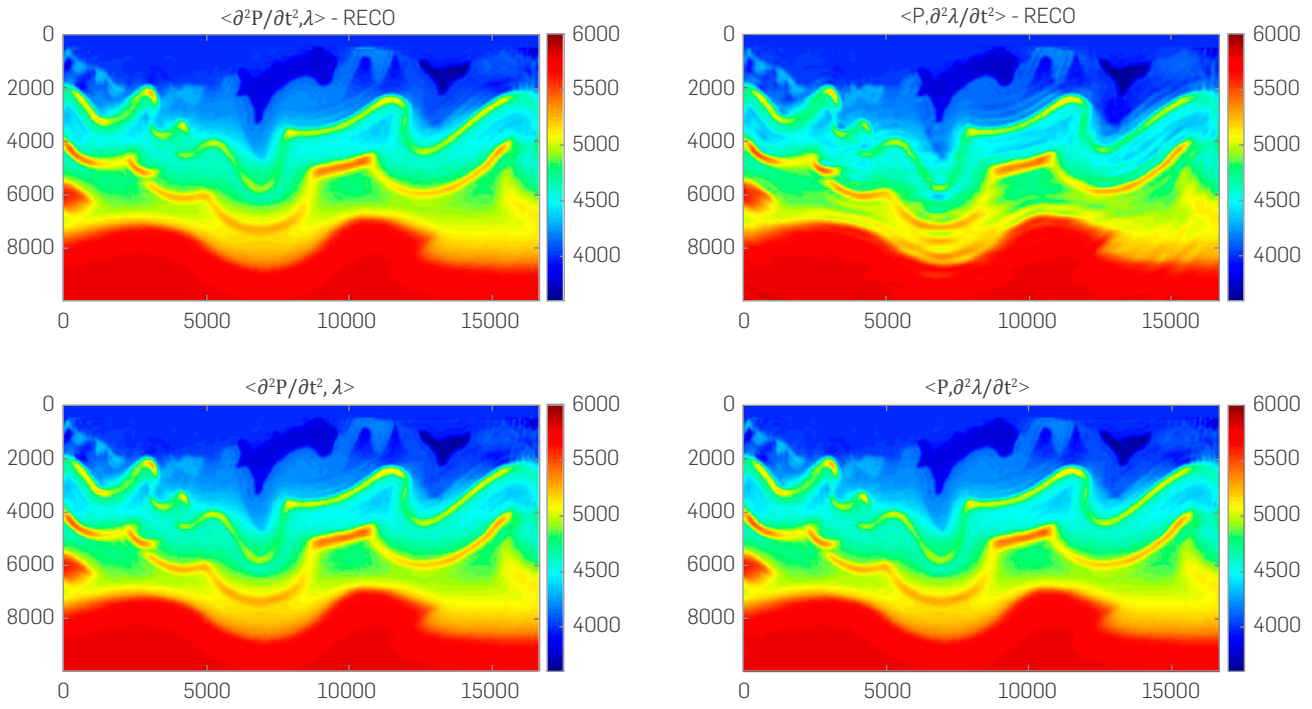


Figure 8. Final velocity model of each experiment with a central frequency of 9 [Hz].

It is to be noted that, with a central frequency of 9 [Hz], the changes over the velocity model are minimal as compared with the previous frequency value and an additional frequency step may not be necessary. Therefore, to obtain a mathematical measure of the results, the Peak Signal to Noise Ratio was calculated for each experiment. **Figures 9**, **10**, and **11** show the PSNR at each FWI iteration for all the experiments. The density data was updated with its respective gradient (Equation 11) through the inversion process, and the initial density model was obtained through the same process of the initial velocity model.

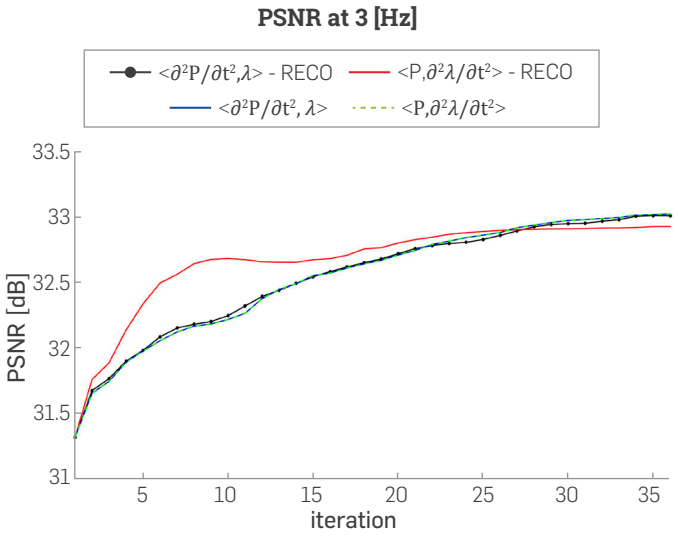


Figure 9. Peak Signal to Noise Ratio between the velocity model at each FWI iteration and the original velocity model with a ricker central frequency of 3 [Hz].

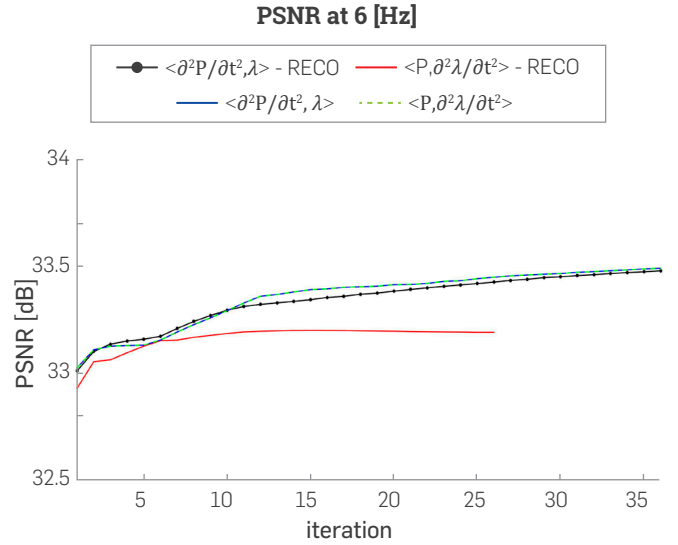


Figure 10. Peak Signal to Noise Ratio between the velocity model at each FWI iteration and the original velocity model with a ricker central frequency of 6 [Hz]. Note that the experiment that combines the wavefield reconstruction and the gradient computation proposed in this work just reached 26 iterations.

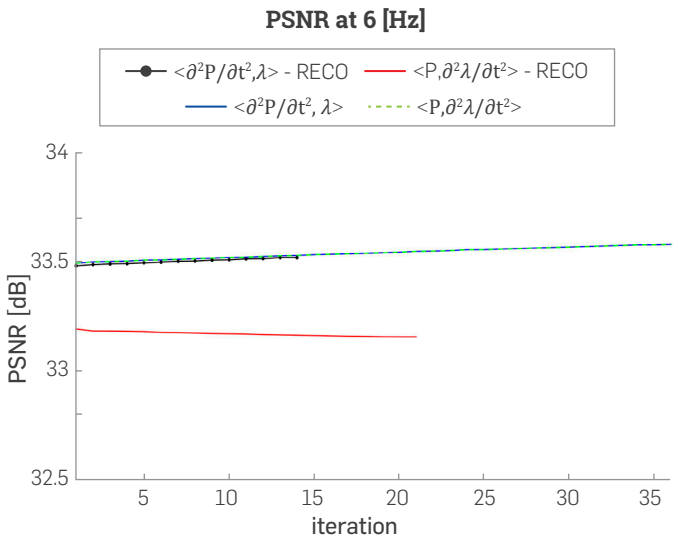


Figure 11. PSNR between the velocity model at each FWI iteration and the original velocity model with a ricker central frequency of 9 [Hz]. At this stage, the experiments that used the wavefield reconstruction strategy did not complete all the 35 iterations. The process stopped because the algorithm did not find any better model to still decreasing the cost function at these iterations.

It should be noted that, when using the wavefield reconstruction strategy, the numerical error associated with the reconstruction step decreases the quality of the result and some of the experiments did not reach all the 35 iterations but the difference in dB of the PSNR curve values are below 0.5.

There is also a measurement of the time consumed by the inversion process (**Table 3**) when using a single GPU and when using the whole computing system (to analyze the effect of using MPI for the inter-node communication).

Note that, as [3] shows, the implementation with the reconstruction strategy and the traditional way to compute the gradient (column A) is affected by a time penalty of 55%, approximately, in comparison with the implementations that do not use the reconstruction strategy (columns C and D). However, when combining this reconstruction strategy and the re-definition of the gradient by applying the inner product property (column B), the time penalty is just near 10% as compared with the results in columns C and D. The slightly differences in the execution time between columns C and D are associated to the effect of other resource utilization in the computing system, but these differences are minimal and can be ignored.

Additionally, it seems that the time penalty associated to data transfer between computing nodes is acceptable. The obtained speed up working with six GPUs (three computing nodes), instead of a single GPU, is approximately 5.6x over an expected speed up of 6x (if data transfer time between compute nodes and other devices is zero) for all the versions of the implementation.

Table 4 is obtained by applying Equations 17 and 18 and using the *nvidia-smi* command on Linux.

Table 3. Execution time for different versions of the implementation. The word RECO means that the reconstruction strategy was applied and the terms inside the angle brackets mean whether the traditional way to compute the gradient or the re-definition of the gradient computation was used. Note that, the re-definition of the gradient computation by applying the inner product property reduces the time penalty of the reconstruction strategy.

Column	A	B	C	D
Gradient	$<\frac{\partial^2 P}{\partial t^2}, \lambda>$ - RECO	$<P, \frac{\partial^2 \lambda}{\partial t^2}>$ - RECO	$<\frac{\partial^2 P}{\partial t^2}, \lambda>$	$<P, \frac{\partial^2 \lambda}{\partial t^2}>$
1 GPU	8801.19[s]	6123.61[s]	5717.05	5723.34
6 GPUs	1585.00[s]	1078.89[s]	1025.22	1017.82

Table 4. RAM used by different versions of the implementation. The word RECO means that the reconstruction strategy was applied and the terms inside the angle brackets mean whether the traditional way to compute the gradient or the re-definition of the gradient computation was used. Second row is the values obtained when applying equations 17 (without reconstruction strategy) and 18 (reconstruction strategy) to calculate the used memory. Third row is the measurement given by the nvidia-smi command of how much memory of the specific GPU is on use.

Column	A	B	C	D
Gradient	$<\frac{\partial^2 P}{\partial t^2}, \lambda>$ - RECO	$<P, \frac{\partial^2 \lambda}{\partial t^2}>$ - RECO	$<\frac{\partial^2 P}{\partial t^2}, \lambda>$	$<P, \frac{\partial^2 \lambda}{\partial t^2}>$
Equation 17/18	144.88[MiB]	144.88[MiB]	4018.42[MiB]	4018.42[MiB]
Nvidia-smi	145[MiB]	145[MiB]	4018[MiB]	4018[MiB]

It is to be noted that, the pair of implementations that use the reconstruction strategy (columns A and B) and the pair that do not use it (columns C and D) have the same memory usage measurements due to the allocated space on RAM is the same, but the applied operations are different.

CONCLUSIONS

This work proposes a different expression to compute the FWI gradient by using an inner product property.

The behaviour of the experiments when using the traditional way to compute the gradient and the expression proposed in this work, without applying the wavefield reconstruction strategy, are identical.

It is possible that the error associated with the reconstruction strategy could introduce values to the velocity models that benefits the behaviour of the PSNR curve at the beginning of the process. However, through the multiscale stages, this error will be higher and the impact on the result will be negative as can be observed in **Figure 11**.

The experiments with the wavefield reconstruction strategy implementation show a different behaviour in comparison with other tests (less FWI iterations as seen in **Figures 10** and **11** and final velocity models with some artifacts). However, the magnitudes in the PSNR curves show differences below 0.5 [dB] and the strategy is still considered useful to increase the performance of the inversion method.

The experiments showed that, when using the reconstruction strategy, at the end of the multiscale process, the numerical error introduced by the time derivative of the backward wavefield (the interaction of multiple sources) is higher than the numerical error introduced by the reconstruction technique.

If the model dimensions require the use of the reconstruction strategy for the inversion problem, the combination of the strategy with the gradient expression proposed in this work will increase the FWI performance in terms of RAM consumption without compromising the execution time.

The numerical error presented in this paper could increase as a result of adding noise to the source and the adjoint wavefield; therefore, it is necessary to evaluate the effect of noise over the reconstruction strategy as further work. Additionally, for real data volumes, it is necessary to analyze the sources and receiver's distribution along the acquisition to obtain the ratio, if it exists, between the numerical error of the reconstruction strategy as a consequence of erroneous positions.

It is necessary to analyze the effect of executing the implementation on a larger computing system (e.g. dozens or hundreds of computing nodes) in terms of speed up and other resources' utilization. Additionally, it is expected that this implementation will be useful to process larger dimension seismic data, given the reduced RAM consumption and execution time.

ACKNOWLEDGEMENTS

This work is supported by Ecopetrol S.A. and Colciencias as a part of the research project grant No. 0266-2013.

REFERENCES

[1] Tarantola, A. 'Inversion of seismic reflection data in the acoustic approximation' *Geophysics*. Volume 49, no. 8 1984, pp. 1259–1266.

[2] Plessix, R.-E. 'A review of the adjoint-state method for computing the gradient of a functional with geophysical applications' *Geophysical Journal International*. Volume 167 no. 2, 2006, pp. 495–503.

[3] Noriega, R. F., Ramirez, A. B., Abreo, S. A., and Arce, G. R. (2017). Implementation strategies of the seismic full waveform inversion. In *2017 IEEE International Conference on Acoustics, Speech and Signal. IEEE*, New Orleans, IL, USA, March 5-9.

[4] Liu, D. C. and Nocedal, J. 'On the limited memory bfgs method for large scale optimization'. *Mathematical programming*, Volume 45 no. 1-3, 1989, pp.503–528.

[5] Bleistein, N. et. al. (2001). '*Mathematics of Multidimensional Seismic Imaging*, Migration, and Inversion'. Springer.

[6] van Meel, J., Arnold, A., Frenkel, D., Portegies Zwart, S., and Belleman, R. 'Harvesting graphics power for md simulations'. *Molecular Simulation*, Volume 34, no. 5, 2008, pp. 259–266.

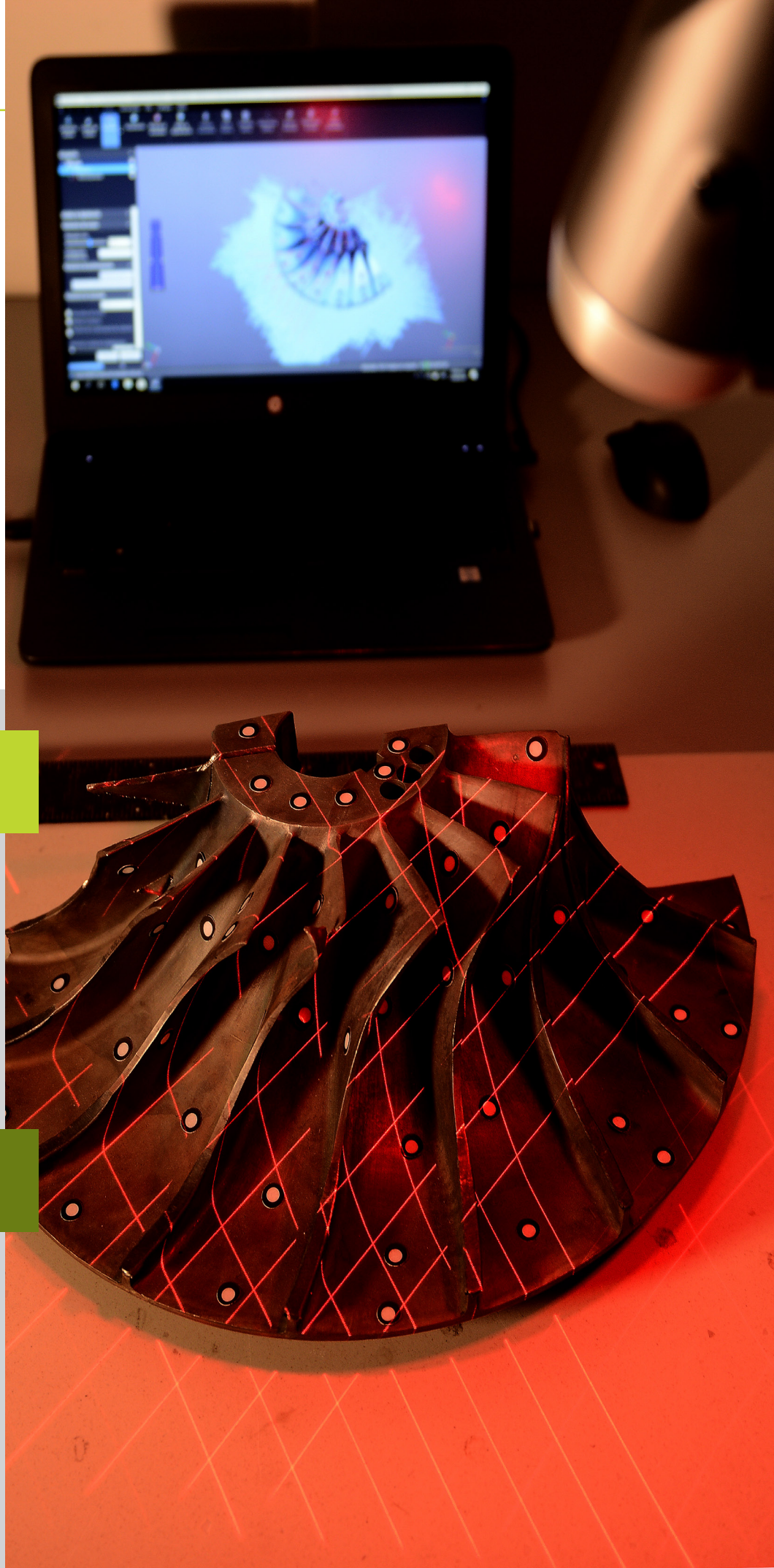
[7] Imbert, D., Imadoueddine, K., Thierry, P., Chauris, H., and Borges, L. (2011). Tips and tricks for finite difference and i/o-less fwi. In *2011 SEG Annual Meeting*. Society of Exploration Geophysicists. SEG, San Antonio, TX, USA, September 18-23.

[8] Yang, P., Brossier, R., and Virieux, J. 'Wavefield reconstruction by interpolating significantly decimated boundaries'. *Geophysics*, Volume 81 no. 5, 2016, pp. 197–209.

[9] Quijada, M. F. and Stewart, R. R. 'Density estimations using density-velocity relations and seismic inversion'. *CREWES*, Volume 19, 2007, pp.1–20.

[10] Mao, J., Wu, R.-S., Wang, B., et al. (2012). Multiscale full waveform inversion using gpu. In *2012 SEG Annual Meeting*. Society of Exploration Geophysicists. SEG, Las vegas, NV, USA, November 4-9.

[11] Pasalic, D., McGarry, R., et al. (2010). Convolutional perfectly matched layer for isotropic and anisotropic acoustic wave equations. In *2010 SEG Annual Meeting*. Society of Exploration Geophysicists. SEG, Denver, CO, USA, October 17-22.



LABORATORIO DE INGENIERÍA DE MATERIALES

En el laboratorio de Integridad y materiales del ICP desarrollamos, evaluamos y adaptamos soluciones tecnológicas orientadas a incrementar la confiabilidad operacional y preservación de la infraestructura evitando pérdidas de contención de fluidos peligrosos, garantizando que los equipos y/o sistemas se encuentren aptos para el servicio durante el ciclo de vida del activo de la operación para Ecopetrol.

ENGINEERING MATERIAL LABORATORY

In the ICP´s laboratory for integrity and materials has the capacity for to evaluate,to adapt and also to test different technological solutions focused on increasing infrastructure operational reallability. In this way it´s possible to avoid leaks or loss of hazardous fluids and also assure the service life of systems and equipments.