ISSN: 2764-5886
e-ISSN: 2764-622X

Volume 6· Supplement 1 · January 2023

Journal of Bioengineering, Technologies and Health

An Official Publication of
SENAI CIMATEC

ICGHPC22

Advanced Technologies in HPC and Quantum Computing Applied in the Oil and Gas Industry

International Congress in Geophysics
and High-Performance Computing

JBTH

January 09 - 11, 2023
SENAI CIMATEC, Salvador, Bahia, Brazil

ISSN: 2764-5886 / e-ISSN 2764-622X

Volume 6 • Supplement 1 • January 2023

JOURNAL OF BIOENGINEERING TECHNOLOGIES AND HEALTH

An Official Publication of SENAI CIMATEC

ICGHPC22
Advanced Technologies in HPC and Quantum Computing Applied in the Oil and Gas Industry

International Congress in Geophysics
and High-Performance Computing
January 9-11, 2023

EDITOR-IN-CHIEF
Leone Peter Andrade

PUBLISHED BY SENAI CIMATEC

Sistema FIEB | SENAI CIMATEC
PELO FUTURO DA INOVAÇÃO

January 2023
Printed in Brazil

JOURNAL OF BIOENGINEERING, TECHNOLOGIES AND HEALTH

An Official Publication of SENAI CIMATEC

JOURNAL OF BIOENGINEERING AND TECHNOLOGY APPLIED TO HEALTH

Volume 6 • Supplement 1 *January 2023*

SUMMARY

Original Articles

Instructions for Authors

Statement of Editorial Policy

The Journal of Bioengineering, Technologies and Health (JBTH) is an official publication of the SENAI CIMATEC (Serviço Nacional de Aprendizagem Industrial - Centro Integrado de Manufatura e Tecnologia). It is published quarterly (March - June - September - December) in English by SENAI CIMATEC – Avenida Orlando Gomes, 1845, Piatã, Zip Code: 41650-010, Salvador-Bahia-Brazil; phone: (55 71) 3879-5501. The editorial offices are at SENAI CIMATEC.

Editorial Office

Correspondence concerning subscriptions, advertisements, claims for missing issues, changes of address, and communications to the editors should be addressed to the Deputy Editor, Dr. Roberto Badaró, SENAI CIMATEC (Journal of Bioengineering, Technologies and Health – JBTH) – Avenida Orlando Gomes, 1845, Piatã, Zip code: 41650-010, Salvador-Bahia-Brazil; phone: (55 71) 3879-5501; or sent by e-mail: jbth@fieb.org.br / jbth.cimatec@gmail.com.

Permissions

Postmaster

Send address changes to JBTH, Avenida Orlando Gomes, 1845, Piatã, Zip Code: 41650-010, Salvador-Bahia-Brazil.

Information by JBTH-SENAI CIMATEC

Address: Avenida Orlando Gomes, 1845, Piatã, Zip Code: 41650-010, Salvador-Bahia-Brazil
Home-page: www.jbth.com.br
E-mail: jbth@fieb.org.br / jbth.cimatec@gmail.com
Phone: (55 71) 3879-5501 / 3879-5500 / 3879-9500

International Congress in Geophysics and High--Performance Computing (ICGHPC22)
Advanced Technologies in HPC and Quantum Computing Applied in the Oil and Gas Industry

Sponsors:
Ministério da Ciência, Tecnologia e Inovações; SENAI CIMATEC; FINEP; PRH 27.1; UFRN; Laboratório Nacional de Computação Científica; Programa de Recursos Humanos da ANP.

10.34178/jbth.v6iSuppl. 1

Copyright

COVER: Layout from International Congress in Geophysics and High-Performance Computing 22 - Advanced Technologies in HPC and Quantum Computing Applied in the Oil and Gas Industry by SENAI CIMATEC.

Optimization of a Geophysical Application in GPU Through the Loop Tiling Technique

Gabriel Pinheiro da Costa[1*], Murilo Boratto[2], Marcelo Oliveira da Silva[3], João Henrique Speglich[4]
[1]Supercomputing Center for Industrial Innovation SENAI CIMATEC (CS2I); Salvador, Bahia, Brazil

This work aims to present the results obtained in optimizing a viscoacoustic geophysical model written with the DEVITO tool and optimized using the OpenACC tile directive for GPU execution. We compared three versions of the operator using the NVIDIA NCU profiling tool: Naive, Tiling (32,4,4), and Mixed Tiling. The Naive version does not use the loop tiling technique, the Tiling (32,4,4) version applies a tile of dimensions (32, 4, 4), and the Mixed Tiling version uses different tile sizes to other loop nests. Analyzing the experimental results, it is notable that the optimized versions substantially increase the cache hit rates and reduce the execution time by about 50%, attesting to the validity of the proposed solutions.

Keywords: HPC. Optimization. OpenACC. Loop Tiling. DEVITO.

Introduction

DEVITO is a tool for implementing computational mesh models in symbolic language. It is a Python package with automated code generation that allows portability to different platforms [1]. DEVITO is a helpful tool for building geophysical models for parallel architectures.

DEVITO allows OpenACC to offload the workload to a device with more processing power, such as a GPU. OpenACC is a programming standard for optimizing C, C++, and Fortran code. The user uses directives to inform the regions of the code that he wants to optimize in an automated way [2].

Through environmental variables, DEVITO can generate code with OpenACC directives capable of promoting GPU execution and parallelization. One such directive is the tile directive, which applies the loop tiling technique [3] in a loop nest with the dimensions defined as a parameter. The loop tiling technique modifies a loop nest, so data is no longer accessed sequentially in one dimension but in multidimensional blocks of predefined size [3]. This transformation uses better nests' spatial and temporal locality [4,5].

Materials and Methods

Developed Approaches

In the analyzed application, two kernels have a more extensive workload, responsible for a large part of the required computational effort: R and RP.

Thus, we selected three approaches for analysis in the viscoacoustic model.

Naive

The tool's default approach, without any parameter optimization or modification of the generated code. It only counts on the "advanced" default optimization level. It is the most straightforward approach.

Tiling (32,4,4)

The approach only uses the part-tile flag (DEVITO native), not requiring any transformations of the .cpp code generated by the framework. This flag applies the loop tiling technique to all loop nests restrained in the operator, using OpenACC's tile directive. The combinations of dimensions that achieved the best performance were 32 elements in x, 4 elements in y, and 4 in z [simply: (32,4,4)].

Mixed Tiling

It works with different tile sizes for loop nests.

Received on 18 December 2022; revised 5 January 2023.
Address for correspondence: Gabriel Pinheiro da Costa. Avenida Orlando Gomes, No.1845, Piatã, Salvador, Bahia, Brazil. Zipcode: 41650-010. E-mail: gabriel.pinheiro@fieb.org.br. DOI 10.34178/jbth.v6iSuppl. 1.265.

Using the OpenACC tile directive, the R kernel applies the technique of loop tiling with dimensions (32,8,4), whereas the RP kernel, through the same process, applies a tiling of sizes (32,4,4). This variation in tile dimensions in the two kernels occurs because the R kernel reached the best performance with measurements (32,8,4), while the RP kernel got its peak performance with dimensions (32,4,4).

Hardware and Experiments

NVIDIA Tesla V100 SXM2 32 GB cards performed all tests, with exclusive access to the hardware and no competition with other applications. The following environmental variables were used to enable GPU execution with OpenACC:

- DEVITO_PLATFORM=nvidiaX;
- DEVITO_ARCH=nvc;
- DEVITO_LANGUAGE=openacc.

The tests were carried out on a three-dimensional model with 701 elements in each dimension, a value that pushed the GPU memory storage capacity to the limit. Each run conducted for 1,000 iterations and applied a space order of 16 elements. All runtime results are means of three runs performed under the same conditions and parameters.

Profiling Tool - Nsight Compute (NCU)

Nsight Compute (NCU) [6] is a CUDA kernel profiler that has a graphical interface and operates by the command line. It offers a series of metrics and sections (metric grouping), which can be collected in a customized way by the user to restrict the scope of the analysis. It is the correct tool to obtain statistical and mathematical information for each application's kernel. Three sections of the NCU were used to analyze the kernels presented in this work.

GPU Speed Of Light Roofline Chart

This section brings two metrics of great value for performance analysis. Arithmetic intensity is the ratio of floating-point operations performed per second, memory transfer in bytes, and per second. This is a metric strictly related to memory traffic. Performance measures the number of floating point operations per second (FLOP/s) and indicates computational performance.

Memory Workload Analysis

Displays data-related GPU memory resources, including cache hit rates. The most relevant metrics in this section are cache hit rates on L1 and L2.

Scheduler Statistics

This section summarizes the schedulers that issue instructions. Each scheduler maintains a group of warps from which it can pull instructions. In each cycle, each scheduler checks the status of the warps allocated in its group (Active Warps), looking for warps that are not stalled (Eligible Warps) and, therefore, ready to issue its next instruction. An eligible warp is then selected, and its instructions are issued (Issued Warp). The parameter that strongly impacts the occupancy rate of a scheduler is the number of registers needed per thread. Each GPU SM has 4 sub-partitions, each one with a scheduler.

Results and Discussion

Table 1 compares the execution times in seconds for each of the three approaches. Table 2 presents the number of cycles spent on each one of the kernels.

Table 3 shows the R kernel results for this NCU GPU Speed of Light Roofline Chart section. A significant improvement in both performance and arithmetic intensity is presented in both optimized approaches. The Tiling (32,4,4) approach almost tripled the values of these two metrics concerning that obtained by the Naive version, and the Mixed Tiling approach was able to surpass three times the values obtained by the Naive version in both metrics. In the RP kernel, the improvement was also noticeable (Table 4), reaching an average of

FLOP per byte almost twice that presented by the Naive version in the two optimized approaches. The average performance also increased notably in the Tiling (32,4,4) and Mixed Tiling approaches. The results presented by the two tables indicate an increase in processing capacity over the same volume of data (arithmetic intensity) and better use of available computational resources (performance).

Tables 5 and 6 show the results of the R and RP kernels for the Memory Workload Analysis section for the three approaches. We significantly increased the cache rates achieved in L1 and L2 in the R kernel with the two optimized approaches. The Mixed Tiling approach performed better in L1, increasing the hit rate at this cache level by almost 44 percentage points compared to the Naive version and by more than 6 percentage points compared to the Tiling (32,4,4) version. In the RP kernel, the optimized approach also increased the cache hit rate at both levels. In L1, this increase is more significant, getting close to reaching twice that obtained by the Naive version, while in L2, the growth is lower but still perceptible, rising by about 7 percentage points compared to the non-optimized version. The values obtained in this section by the two optimized approaches in both kernels converge with the results presented. Increasing cache hit rates allows more efficient use of data, reducing processing bottlenecks and allowing the application to use better available processing power (improved performance and arithmetic intensity). The higher L1 cache hit rate of the Mixed Tiling approach compared to the Tiling (32,4,4) approach is one factor that explains the slightly superior performance of one strategy over the other.

Table 1. Execution times.

Approach	Time (s)
Naive	220.14
Tiling (32,4,4)	112.47
Mixed Tiling	110.01

Table 2. Cycles spent on each kernel.

Approach	R	RP
Naive	117.272.997	276.679.918
Tiling (32,4,4)	52.049.206	110.608.921
Mixed Tiling	43.834.245	110.124.618

Table 3. R - GPU speed of light roofline chart.

Approach	Performance (FLOP/s)	Arithmetic Intensity (FLOP/byte)
Naive	$0.317 \cdot 10^{12}$	0.53
Tiling (32,4,4)	$0.953 \cdot 10^{12}$	1.57
Mixed Tiling	$1.069 \cdot 10^{12}$	1.84

Table 4. RP - GPU speed of light roofline chart.

Approach	Performance (FLOP/s)	Arithmetic Intensity (FLOP/byte)
Naive	$0.370 \cdot 10^{12}$	0.63
Tiling (32,4,4)	$0.581 \cdot 10^{12}$	1.23
Mixed Tiling	$0.583 \cdot 10^{12}$	1.22

Table 5. R - Memory Workload Analysis.

Approach	L1 Cache Hit (%)	L2 Cache Hit (%)
Naive	34.37	36.38
Tiling (32,4,4)	71.42	59.95
Mixed Tiling	77.73	58.57

Table 6. RP - Memory Workload Analysis.

Approach	L1 Cache Hit (%)	L2 Cache Hit (%)
Naive	33.68	30.61
Tiling (32,4,4)	61.60	37.92
Mixed Tiling	61.59	37.54

Table 7 presents the R kernel results for the scheduler statistics section. Although there is no significant variation in the theoretical maximum amount of warps per scheduler between the Tiling(32,4,4) and Naive versions, the rates of eligible and effectively issued warps are notably accentuated in the Tiling (32,4,4) and Mixed Tiling, which reach values that exceed three times that obtained by Naive, with the Mixed Tiling approach having slightly higher values. A similar but more timid result is obtained by the optimized approaches in the RP kernel, as shown in Table 8. The theoretical maximum of warps per scheduler does not change; however, the amount of eligible warps exceeds twice the Naive version, and the average warps emitted per cycle jumps from 0.14 to 0.25. The results point to better use of the schedulers in the Tiling(32,4,4) and Mixed Tiling versions, which start to emit more warps per cycle and mitigate the possibilities of taking the computational resources to idleness. The increase in cache hit rates in the optimized approaches is the main factor that increased the average warps emitted.

Conclusion

The results reveal that both kernels are positively sensitive to loop tiling. In the R kernel, the application of the OpenACC tiling directive had the main effect of substantially increasing the cache hit rates at both levels in the two optimized approaches. This better use of cache memory allowed an increase in computational efficiency, observed in improving metrics such as arithmetic intensity, performance, and the rate of warps emitted per cycle. After applying the loop tiling technique, the RP kernel went through a process similar to that of the R kernel, which had as its main positive effect the increase in cache hit rates. This more efficient use of cache memory increased computational efficiency, increasing metrics such as arithmetic intensity, performance, and the rate of warps emitted. Therefore, despite the very similar version of the two optimized approaches, the slightly higher cache hit rate of the Mixed Tiling approach over Tiling(32,4,4) in L1 ends up giving the process that uses mixed tiling dimensions a slightly better performance.

Table 7. R - Scheduler statistics.

Approach	Theoretical Maximum	Eligible	Emitted
Naive	5	0.19	0.11
Tiling (32,4,4)	4	0.65	0.35
Mixed Tiling	8	1.16	0.42

Table 8. RP - Scheduler statistics.

Approach	Theoretical Maximum	Eligible	Emitted
Naive	4	0.19	0.14
Tiling (32,4,4)	4	0.40	0.25
Mixed Tiling	4	0.40	0.25

References

1. Devito: Symbolic Finite Difference Computation. Available at: https://www.devitoproject.org. Accessed Nov. 2022.
2. OpenACC Programming and Best Practices Guide. Available at: openacc.org. Accessed on: 20 Nov. 2022.
3. Jeffers J, Reinders J. High-performance parallelism pearls volume two: multicore and many-core programming approaches. Morgan Kaufmann 2015:410-416.
4. McKinley KS, Carr S, Tseng C-W. Improving data locality with loop transformations. ACM Transactions on Programming Languages and Systems (TOPLAS) 1996;18(4):424-453.
5. Kandemir M, Ramanujam J, Choudhary A. Improving cache locality by a combination of loop and data transformations. IEEE Transactions on Computers 1999;48(2):159-167.
6. Nsight Compute: Developer Tools Documentation. Available at: https://docs.nvidia.com/nsight-compute/NsightCompute/index.html. Accessed Nov. 2022.

Forward and Adjoint Modeling of Three Viscoacoustic Equations Based on Rheological Models Using DEVITO

Laian de M. Silva[1*], Peterson Nogueira[1,2], Laue R. de Jesus[1]
[1]SENAI/CIMATEC University Center; [2]Federal University of Bahia, INCT-GP; Salvador, Bahia, Brazil

Disregarding the effect of seismic signal attenuation in gas and hydrocarbon zones leads to a final image with low resolution. However, a range of equations called viscoacoustics overcome such limitations. We use three second-order equations based on Maxwell, Kelvin-Voigt (KV), and Standard Linear Solid (SLS) models. We analyze the dissipation and dispersion effects on each of them through seismograms. We also perform Reverse Time Migration (RTM) using the exact adjoint operators (Q-RTAM). All numerical experiments were implemented using Devito - a domain-specific language (DSL) and code generation framework to design highly optimized finite difference kernels for use in inversion methods.

Keywords: Viscoacustic Equations. Domain-Specific Language. DEVITO. Seismic Attenuation.

Introduction

The seismic signal suffers a mechanical energy loss when propagating through the rock. Such loss is reflected in the attenuation phenomenon. The attenuation of the seismic signal is present in the seismograms due to the dispersive and dissipative effect of the signal [1-3]. The dispersive effect causes the dephasing of particular frequency contents, and the dissipation reduces the signal's amplitude. Both combinations cause significant signal distortion when such effects are not considered.

Although several theoretical models, mainly in the frequency domain, have been developed to describe the effect of attenuation, the ones with the best physical meaning are those based on Rheological models composed of a series of relaxation mechanisms [2]. Of this range of rheological models, three deserve to be highlighted. They are Maxwell, Kelvin-Voigt (KV), and Standard Linear Solid (SLS) because they are easy to implement. The Maxwell model consists of a series combination of springs (responsible for the elastic behavior of the material) and dashpot (attenuation element). In this model, the force/stress applied to the elements are equivalent, changing only the response of each element, also known as strain, which is more significant in the spring than in the dashpot.

As a consequence of this behavior, its quality factor (Q) is directly proportional to the frequency, with the most outstanding damping occurring at low frequencies [4]. The Kelvin-Voigt Model represents a parallel combination of spring and dashpot. The strain in both elements is the same; however, each element's force or stress is different. Consequently, its quality factor Q is inversely proportional to the frequency, and the attenuation is more potent at higher frequency contents. Consequently, its quality factor is inversely proportional to the frequency, and the attenuation is more substantial at higher frequency contents. The stress-strain relationship obtained from this model is of the convolutional temporal type, which requires recording the wave fields in each. However, this temporal convolution can be replaced by introducing an auxiliary memory variable [4].

The present work compares wave fields, seismograms, and RTAM images for geological media of different complexities. The equations based on the SLS, Kelvin-Voigt, and Maxwell models are well-defined in the works of Carcione and colleagues [5], Deng and colleagues [6], Gardner and colleagues [7], and Carcione [4]. We implement all equations using Devito, a DSL used to solve

Received on 17 December 2022; revised 6 January 2023.
Address for correspondence: Laian de M. Silva. Avenida Orlando Gomes, S/N, Salvador, Bahia, Brazil. Zipcode: 41650-010. E-mail: laianufba@gmail.com. DOI 10.34178/jbth.v6iSuppl. 1.266.

modeling and seismic inversion problems in a high-performance computational environment [8, 9].

Forward and Adjoint Modeling Equations

The construction of mechanical models is based on two elements (springs and dashpots) connected in parallel or a combination of series and parallel. The spring represents the elastic behavior, whereas the dashpot (represented by a cylindrical piston filled with viscous liquid) denotes the dampening behavior.

The viscoacoustic equations based on rheological models originated from the stress-strain relationship.

$$\sigma = \frac{\partial \psi}{\partial t} * \varepsilon = \psi * \frac{\partial \varepsilon}{\partial t}, \tag{1}$$

σ is the stress, ε is the deformation, and ψ is the relaxation function. Moreover, we have the following relation:

$$\frac{\partial \varepsilon}{\partial t} = \nabla \cdot \mathbf{v}, \tag{2}$$

with v being obtained by the motion equation:

$$\frac{\partial \mathbf{v}}{\partial t} = \frac{1}{\rho} \nabla \sigma, \tag{3}$$

in which v is the particle velocity, and ρ is the density.

<u>Maxwell Model</u>

The relaxation function for the Maxwell model is defined as:

$$\psi = M_U e^{(-t/\tau)} H(t) \tag{4}$$

in which M_U is the elasticity constant of the unrelaxed spring, H(t) is the Heaviside function, and $\tau = \eta / M_U = \omega_0 Q$ is the relaxation time (η is the viscosity and Q is the quality factor). From Equation 1 with some operations, we obtained the equations system:

$$\begin{cases} \frac{\partial p}{\partial t} + \kappa \nabla \cdot \mathbf{v} + \frac{\omega_0}{Q} p = \int S(\mathbf{x}_s, t) \\ \frac{\partial \mathbf{v}}{\partial t} + \frac{1}{\rho} \nabla p = 0, \end{cases} \tag{5}$$

in which κ is the Bulk modulus and $\omega_0 = 2\pi f_0$ is the angular frequency (f_0 is the dominant frequency). Differentiating about time the first Equation of system 5 and substituting the second in the first, there is as follows:

$$\frac{\partial^2 p}{\partial t^2} - \kappa \nabla \cdot \frac{1}{\rho} \nabla p + \frac{\omega_0}{Q} \frac{\partial p}{\partial t} = S(\mathbf{x}_s, t) \tag{6}$$

The adjoint-state method [10] was used to computation the adjoint equation of the forward modeling operator. Applying the adjoint operation in Equation 6, we have the following:

$$\frac{\partial^2 q}{\partial t^2} - \nabla \cdot \frac{1}{\rho} \nabla \kappa q + \frac{\omega_0}{Q} \frac{\partial q}{\partial t} = -\Delta d \tag{7}$$

<u>Kelvin-Voigt Model</u>

The relaxation function to the KV model is determined as:

$$\psi = M_R H(t) + \eta \delta(t), \tag{8}$$

in which M_R is the elasticity constant of the relaxed spring, η is the viscosity, H(t) and δ(t) are the Heaviside and Dirac delta functions, respectively. Once more, starting from Equation 1, using the relaxation function (Equation 8) and following some steps:

$$\begin{cases} \frac{\partial p}{\partial t} + \kappa \nabla \cdot \mathbf{v} - \eta \nabla \cdot \frac{1}{\rho} \nabla p = \int S(\mathbf{x}_s, t), \\ \frac{\partial \mathbf{v}}{\partial t} + \frac{1}{\rho} \nabla p = 0. \end{cases} \tag{9}$$

in which $\eta = \tau\kappa$ with $\tau = (\omega_0 Q)^{-1}$, where κ, η and τ are the Bulk moduli, viscosity, and relaxation time, respectively. Differentiating regarding time, the first Equation of system 9 and substituting the equation of motion, we arrive at:

$$\frac{\partial^2 p}{\partial t^2} - \kappa \nabla \cdot \frac{1}{\rho} \nabla p - \eta \nabla \cdot \frac{1}{\rho} \nabla p = S(\mathbf{x}_s, t) \tag{10}$$

After applying the adjoint-state method in Equation 8, we obtain:

$$\frac{\partial^2 q}{\partial t^2} - \nabla \cdot \frac{1}{\rho} \nabla \kappa q + \nabla \cdot \frac{1}{\rho} \nabla \eta q = -\Delta d \tag{11}$$

SLS Model

The SLS model is the most realistic, consisting of a KV model connected in series with a spring. The relaxation function of this model is defined by

$$\psi = M_R\left[1-\left(1-\frac{\tau_\varepsilon}{\tau_\sigma}\right)e^{-t/\tau_\sigma}\right]H(t) \tag{12}$$

Thus, starting with Equation 1, using Equation 10, and following some steps, there are as follows:

$$\begin{cases} \frac{\partial^2 p}{\partial t^2} - \kappa(1+\tau)\nabla\cdot\frac{1}{\rho}\nabla p + r_p = S(\mathbf{x}_s, t) \\ \frac{\partial r_p}{\partial t} - \frac{\tau}{\tau_\sigma}\rho\nabla\cdot\frac{1}{\rho}\nabla p + \frac{1}{\tau_\sigma}r_p = 0 \end{cases} \tag{13}$$

Where ρ(x) is the density at position x, $\kappa(x)$ is the Bulk modulus, $\mathrm{v} = v(x,t)$ is the particle velocity vector, and $S = S(x_s,t)$ is the source at position x_s. The symbol $*$ represents a convolution operation, which describes the dissipation mechanism in a viscoacoustic medium in Equation 1. $\tau = \tau_\varepsilon/\tau_\sigma - 1$ represents the magnitude of Q. τ_ε and τ_σ are, respectively, the relaxation time stress and strain, given by:

$$\tau_\sigma = \frac{\sqrt{Q^2+1}-1}{2\pi f_0 Q} \quad and \quad \tau_\varepsilon = \frac{\sqrt{Q^2+1}+1}{2\pi f_0 Q} \tag{14}$$

Applying the adjoint-state method in Equation 11:

$$\begin{cases} \frac{\partial^2 q}{\partial t^2} - \nabla\cdot\frac{1}{\rho}\nabla(1+\tau)\kappa q + \nabla\cdot\frac{1}{\rho}\nabla\rho r_q = -\Delta d, \\ \frac{\partial r_q}{\partial t} + \frac{\tau}{\tau_\sigma}q - \frac{1}{\tau_\sigma}r_q = 0 \end{cases} \tag{15}$$

Numerical Experiments

Two-Layer Model

The two-layer model (Figure 1) is used to show the stability of the viscoacoustic modeling for a medium with high-velocity contrast and Q factor and to highlight the reflection and transmission phenomena in Figure 6.

Figure 1. Two-layer model parameters. Top layer: v =1800 m/s and Q = 30. Bottom layer: v = 3000 m/s and Q = 100.

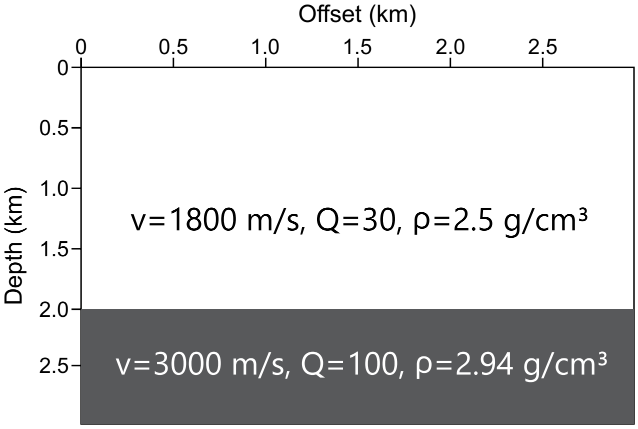
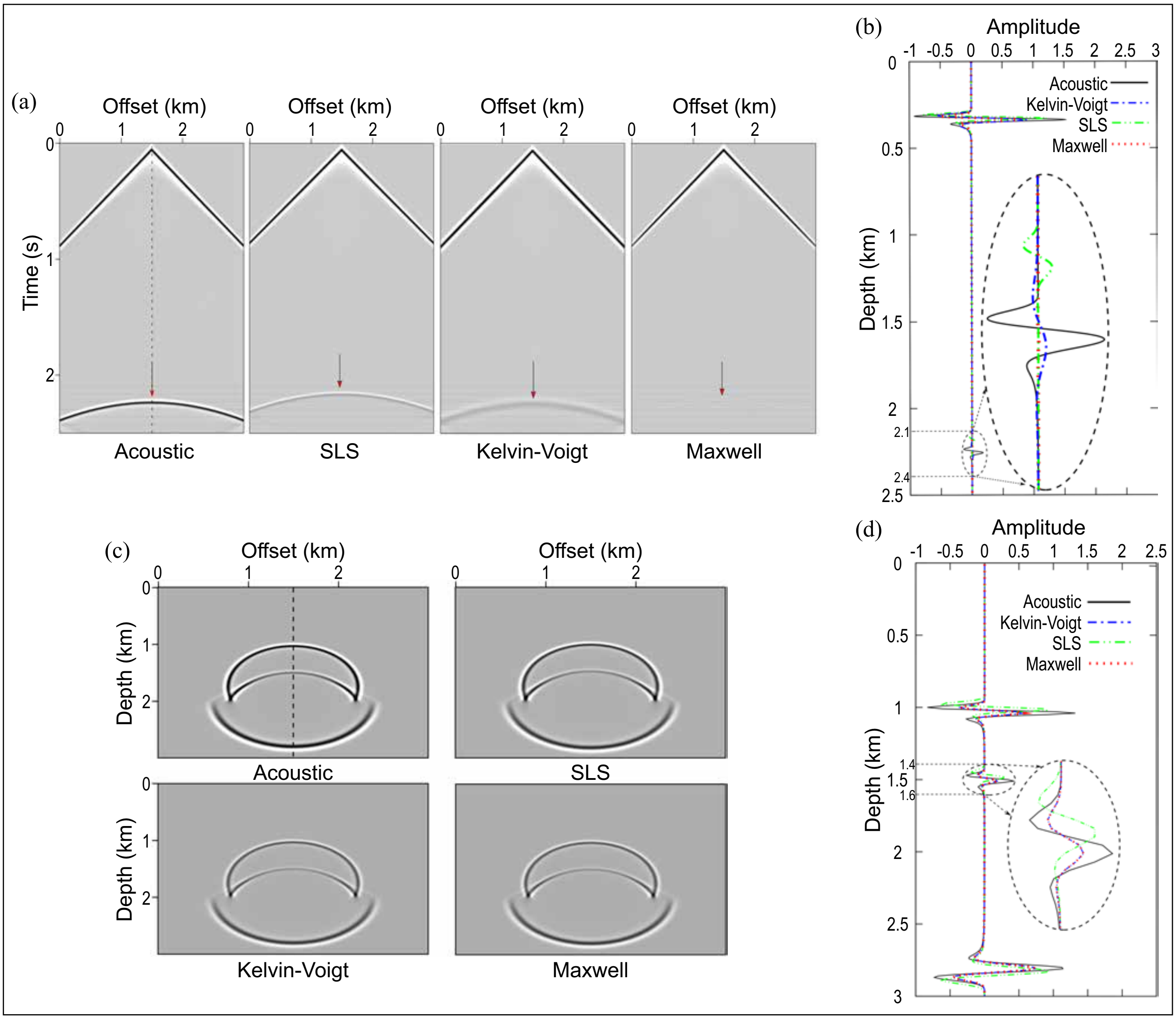

Offset (km)
0
0.5
1.0
1.5
2.0
2.5
Depth (km)
0
0.5
1.0
1.5
2.0
2.5
v=1800 m/s, Q=30, ρ=2.5 g/cm³
v=3000 m/s, Q=100, ρ=2.94 g/cm³

Figure 2a shows a modeled seismogram with a source in the center at the surface. According to the reflection indicated by the red arrow, a substantial drop in amplitude occurs in all viscoacoustic equations. However, the SLS model is the one that has the lowest loss, especially the Maxwell model, which ends up with an almost imperceptible reflection. Therefore, we compared the traces taken at the 1 km position to more accurately observe the effects observed in the seismograms (Figures 2a and 2b). In this image, the green trace has a sizeable temporal displacement concerning the others, indicating that the SLS models have a strong dispersion. Still analyzing the green trace, there is also a marked loss of amplitude concerning the reference acoustic. On the other hand, the KV and Maxwell EVAs showed only energy dissipation, with Maxwell being extremely sensitive to low Q factor values.

Figure 2c presents the time instants of a wave field propagated up to 0.5 s time. At first glance, it is worth noting that the generated wavefronts with the Maxwell and KV equations present more evident energy dissipation than those of the SLS. Therefore, we took a trace at the 1.5 km offset (Figure 2c) to the wavefront of all equations and put them together in a comparison (Figure 2d). Observing these comparisons, we concluded that the green dashed line representing the SLS model is the only one that presents a phase shift concerning the acoustic case, regardless of the order of the equation used.

Figure 2. Seismogram comparisons, for two layer model, between acoustic a viscoacoustic cases (a) and vertical traces comparison at 1 km (b) Snapshots comparison, for two-layer model, between acoustic and viscoacoustic cases at instances of 0.5s (c) and vertical traces comparison at 1.5 km (d).

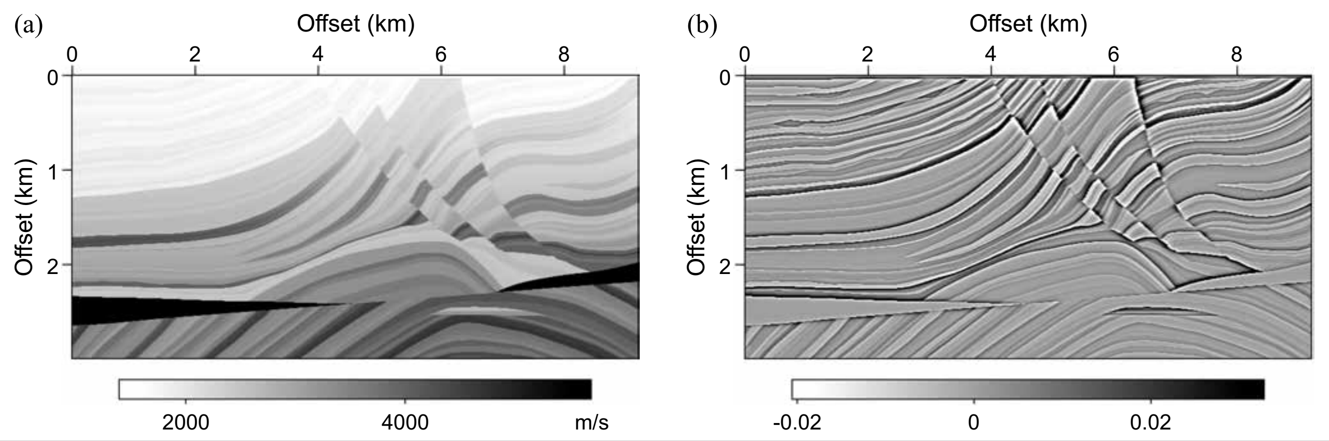


<u>Marmousi</u>

Figure 3a shows the velocity model. The values vary from 1,500 m/s to 5,500 m/s. And the Q model was calculated by empirical equation $Q = 3.516 \times v^{2.2} \times 10^{-6}$ [8]. Finally, we obtained the density model using the Gardner relation [11]. In addition, we calculated an approximation of reflectivity (Figure 2b). These models have 9.2 × 3 km 2 and were discretized with nx = 369 and nz = 375 samples in a mesh of Δx = 25 m and Δz = 8 m. The recording time was 4 s, with a frequency peak of 12 Hz, considering 5716 samples at a sampling interval Δt = 0.7 ms.

We performed numerical simulations for the seismogram with the source located at 4.7 km and receivers scattered on the surface. The arrows in Figure 4a show a decrease in the amplitudes of seismic events. Analyzing the central region of the seismograms in the time interval from 1.5 to

Figure 3. Marmousi model: velocity (a) and reflectivity (b).



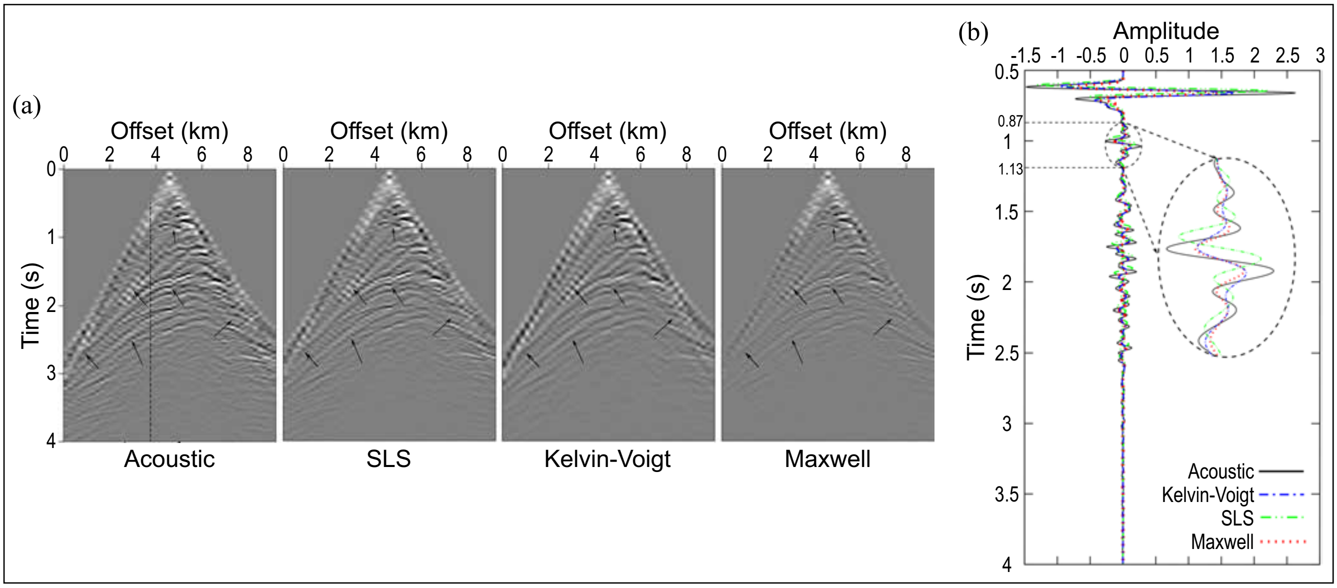
2.5 s presents a more significant attenuation than the other regions, mainly for the Maxwell and KV viscoacoustic equations. Figure 4c ahows the traces taken at 3.75 km offset. Examining the green trace in the SLS viscoacoustic equation, we observe that a temporal displacement does not occur in the remaining traces, indicating that only phase dispersion occurs for this equation.

Figure 5 presents the Q-RTAM results for the SLS, Kelvin-Voigt, and Maxwell models. Figures show that the SLS image is much superior to KV and Maxwell. This is due to equations based on SLS models better compensating for the signal dispersion, something that is less noticeable in the other equations than has already been seen. KV, on the other hand, presents a significantly better result than Maxwell due to the dissipative effect in Maxwell being stronger at lower frequencies, which is where most of the frequency content of the seismic is represented.

Gas Chimney

The Gas Chimney constitutes a minor clipping of the BP model [12]. Figure 6a and Figure 6b

Figure 4. Marmousi model: seismograms comparison among acoustic, SLS, KV, and Maxwell equations (a) and traces comparison at 3.75 km (b).



illustrate the velocity and reflectivity, respectively. The density and Q factor were calculated by the Gardner relation [13] and empirical equation [8]. These models have $9.995 \times 4km^2$ discretized with nx = 995 and nz = 402 samples.

Figure 7a shows all the seismograms for the analyzed viscoacoustic equations. The Gas Chimney model has a low Q factor anomaly in the upper central part of the model (Figure 7a). The effect is notable in seismograms, presenting significant energy loss, mainly for the Maxwell model. Analyzing the red dashed circle, we notice that the dissipative effect is not as strong as the region for the longer times indicated by the arrows.

Furthermore, the SLS equation's velocity phase dispersion effect exists beyond amplitude reduction, which causes seismic pulse distortion. The traces for the equations based on the Maxwell and KV rheological models show the dissipative effect (Figure 7). However, they are not displaced, characterizing the non-occurrence of the dispersive effect because these viscoacoustic equations do not consider the dispersion phenomena.

Figure 8 shows the result for the Q-RTAM for the gas chimney model. In this figure, the behavior seen in the Marmousi is repeated, where the SLS presents a better result, and it is possible to observe the characteristics of the model in the region of the chimney of the Q factor. The KV also presents a better resolution, mainly in the chimney region.

Figure 5. Marmousi model: Q-RTAM for SLS, Kelvin-Voigt and Maxwell rheological model.

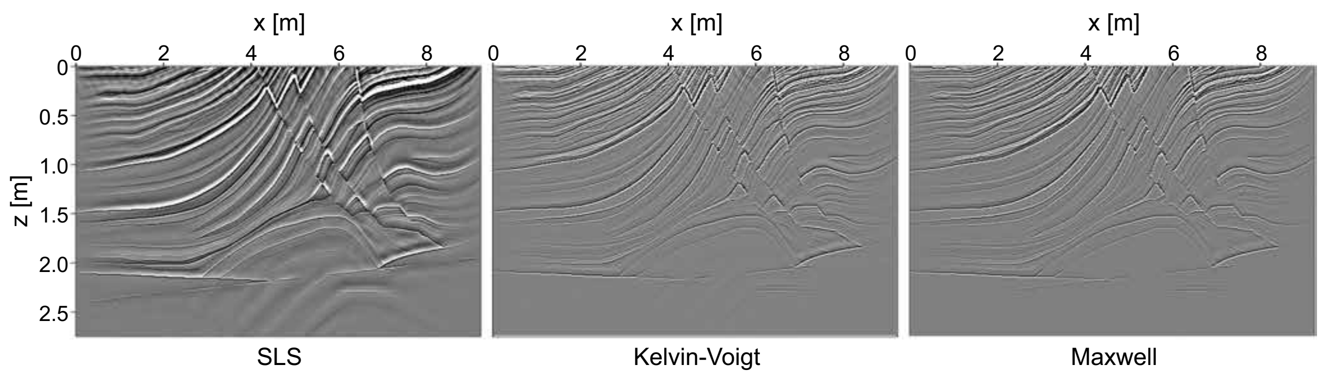
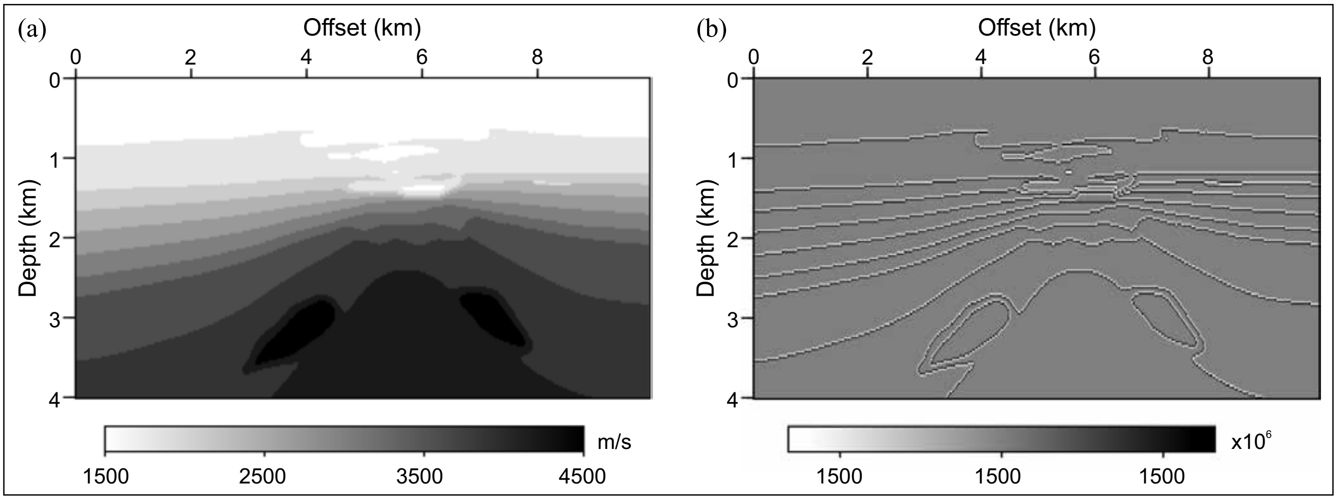


Figure 6. Gas chimney model: velocity (a); reflectivity (b).



Conclusion

We performed physical-numerical experiments considering velocity, quality factor, and density models with different complexities, explaining in detail the behavior and characteristics of each equation. Unlike the equations based on the Maxwell and KV models, the SLS equations can reasonably simulate the energy dissipation and phase dispersion phenomena in the forward modeling stage.

Acknowledgments

This study was financed by the project PIE00005/2016 of Infrastructure Edictal of FAPESB 003/2015 and was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-Brazil (CAPES)-Finance Code 001.

Figure 7. Gas chimney model: seismogram comparisons between acoustic, SLS, KV, and Maxwell equations (a), and traces comparison at 3.75 km (b).

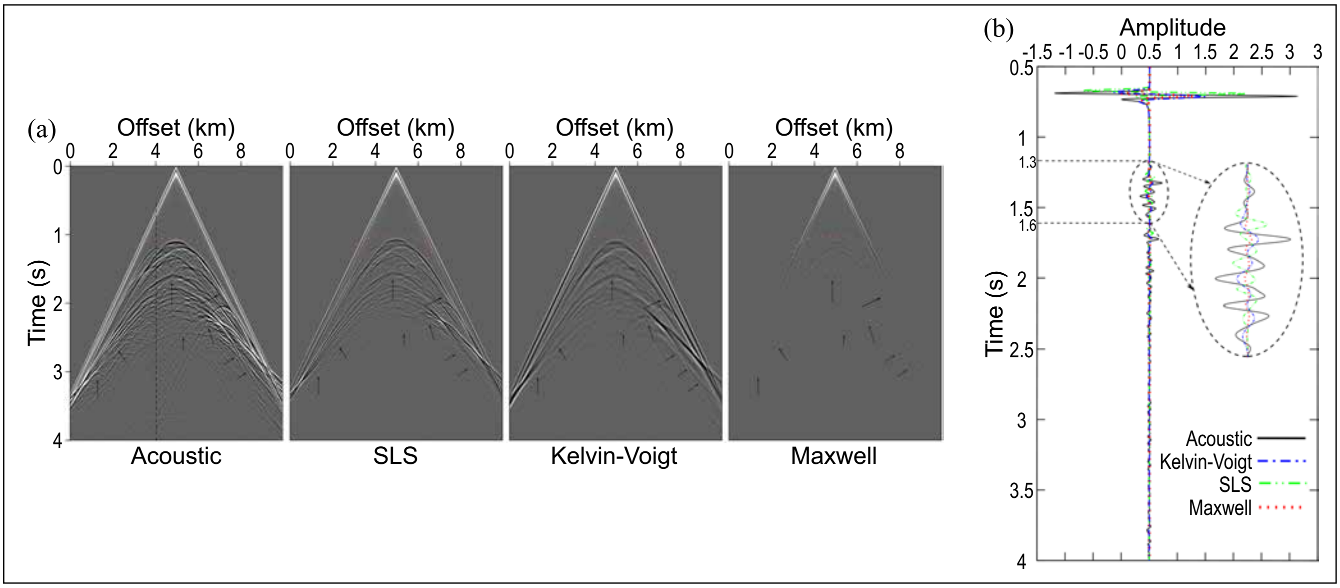
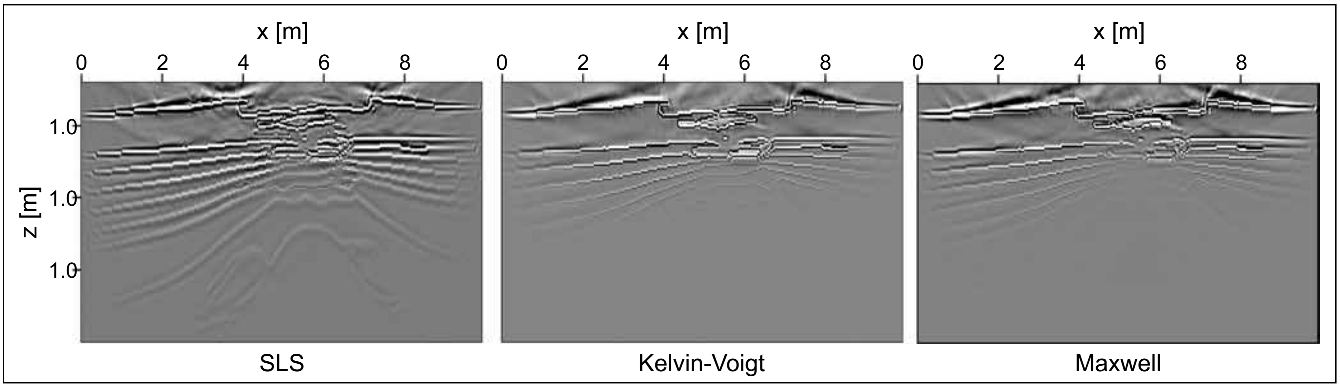


Figure 8. Gas chimney model: Q-RTAM for SLS, Kelvin-Voigt and Maxwell rheological model.



References

1. Walcott R. Flexural rigidity, thickness, and viscosity of the lithosphere, Journal of Geophysical Research. 1970;75(20):3941–3954.
2. Schiessel H, Metzler R, Blumen A, Nonnenmacher T. Generalized viscoelastic models: Their fractional equations with solutions. Journal of Physics: Mathematical and General 1995;28(23):6567.
3. Robertson JO, Blanch JO, Symes WW,. Viscoelastic finite-difference modeling, Geophysics. 1994;59(9):1444–1456.
4. Carcione JM, Kosloff D, Kosloff R. Wave propagation simulation in a linear viscoelastic medium. Geophysical Journal International 1988;95(3):597–611.
5. Carcione JM. Wave Fields in Real Media: Wave Propagation in Anisotropic, Anelastic, Porous, and Electromagnetic Media, Elsevier, 2014.
6. Bai JD, Yingst R, Bloor, Leveille J. Viscoacoustic waveform inversion of velocity structures in the time domain. Geophysics 2014;79:R103–R119.
7. Ren Z, Liu Y, Zhang Q. Multiscale viscoacoustic waveform inversion with the second generation wavelet transform and adaptive time–space domain finite-difference method. Geophysical Journal International 2014;197(2):948–974.
8. Kukreja N et al. High-level python abstractions for optimal checkpointing in inversion problems. arXiv preprint arXiv 2018. 1802.02474.
9. Louboutin M et al. Devito: An embedded domain-specific language for finite differences and geophysical exploration. arXiv 2018. preprint arXiv:1808.01995.
10. Plessix R-E. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. Geophysical Journal International 2006;167(2):495–503.
11. Gardner G, Gardner L, Gregory A. Formation velocity and density—the diagnostic basics for stratigraphic traps. Geophysics 1974;39(6):770–780.
12. Billette F, Brandsberg-Dahl S. The 2004 bp velocity benchmark. 67th EAGE Conference & Exhibition, European Association of Geoscientists & Engineers 2005;Supl–1.
13. Deng F, McMechan GA. True-amplitude prestack depth migration. Geophysics 2007;72(3):S155–S166.

Enhancing DEVITO GPU Allocator Using Unified Memory by NVIDIA

Gustavo Araujo Alvaro Coelho[1*], Atila Saraiva Quintela Soares[1], João Henrique Speglich[1], Marcelo Oliveira da Silva[1]
[1]Supercomputing Center for Industrial Innovation SENAI CIMATEC (CS2I); Salvador, Bahia, Brazil

DEVITO is a framework whose objective is to implement optimized stencil computing. Its execution can be carried out both in the CPU and in GPU. For this reason, the data must be manipulated correctly so that, in case of executions in the GPU, they are present in the memory of the GPU at the time of the execution. Natively, DEVITO transfers data every time the operator is executed from OpenACC pragmas. This approach results in performance degradation when the operator is executed repeatedly. To prevent redundant copies and alleviate this bottleneck, an allocator based on unified memory was implemented, which makes manual data transfer between CPU and GPU unnecessary, significantly reducing data transfer time in GPU applications.
Keywords: DEVITO. Unified Memory. GPU. Data Transfer.

Introduction

DEVITO [1,2] is a framework developed in Python, whose objective is to implement optimized stencil computation (e.g., finite differences, image processing). This tool uses Sympy code and automatic code generation, transforming the user's Python implementation into C code, which is lighter and faster to run computational kernels on different platforms, such as CPUs or GPUs. To perform data allocation, DEVITO implements different classes of memory allocators.

Each one with specific characteristics, from allocators based on the use of the POSIX library to allocators based on non-uniform memory access (NUMA). It is up to the user to choose an allocator that presents the best characteristics for his application. Given the available allocator options and their different characteristics, there is an essential similarity between them: they all allocate data within the CPU memory. Figure 1. Graphs illustrate the amount of data transfer between CPU and GPU before kernel execution. The blue bar represents kernel execution, and the green bar represents memory transfer between CPU and GPU.

Depending on the environment configuration, the DEVITO operator can run on both CPU and GPU. However, running on GPU requires data to be present within its memory region. For this reason, the device on which the data is initially allocated is crucial for the system's performance and the analysis of its functioning.

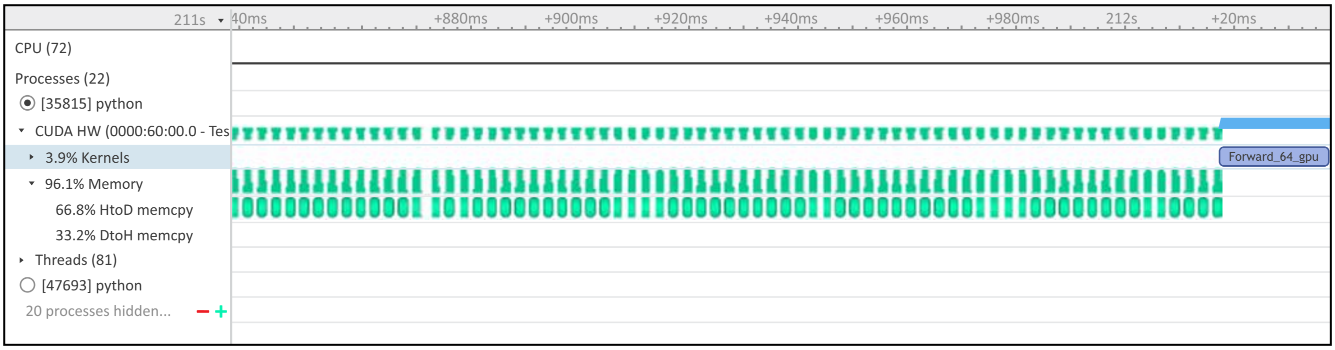
As DEVITO default allocators allocate data to the CPU, data transfer before and after operator computation is required. It uses pragma directives from the OpenACC library through the sub-directives copy in and copy out. The problem with this approach is that the data transfer process is defined within the operator, so if this operator is executed multiple times with the same arguments (e.g., execution using Checkpoint), data will also be transferred multiple times, generating redundant data stream that affects system performance. Figure 1 illustrates the amount of data transferred from CPU to GPU (Host to Device) before kernel execution.

Materials and Methods

In order to reduce the number of redundant copies mentioned above, a new allocator was created based on unified memory. Data allocated in unified memory can be accessed by CPU and GPU devices, making an exact copy of data unnecessary. Figure 2 illustrates the schematic unified memory,

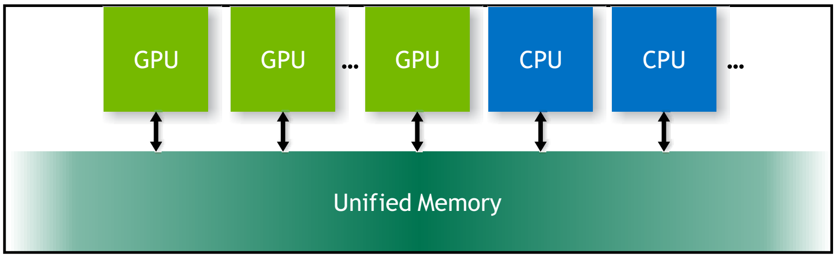
Received on 20 December 2022; revised 8 January 2023.
Address for correspondence: Gustavo Araujo Alvaro Coelho. Av. Artemia Pires Freitas, 8220, Cond. Viva Mais Master, Casa 6, Feira de Santana - BA. Zipcode: 44085-370. E-mail: gustavo.coelho@fieb.org.br.DOI 10.34178/jbth.v6iSuppl. 1.267.

Figure 1. Amount of data transfer between CPU and GPU prior to kernel execution.



Blue bar represents kernel execution and green bar represents memory transfer between CPU and GPU.

Figure 2. Schematic unified memory, which can be accessed by both CPU and GPU.



which can be accessed by both CPU and GPU [3]. The implementation of the new allocator was based on the use of CuPy [4], which, in addition to implementing unified memory in its structure, is very familiar with the NumPy library, which is widely used within DEVITO. The similarity between the two meant that implementing CuPy inside the DEVITO structure presented minimal difficulties.

The default allocator of CuPy API is set to use unified memory; any object created by it will have its data allocated in MU. Therefore, once the allocator was changed, a CuPy array of the same size as the desired data was created to allocate the data.

In order to analyze the obtained results and verify how much the unified memory improves the system performance, a test was applied. This test ran an algorithm responsible for direct propagation and adjunct calculation for a single source using the DEVITO and PyRevolve tools. This algorithm was run twice, once using the standard DEVITO allocator and once using the unified memory-based allocator.

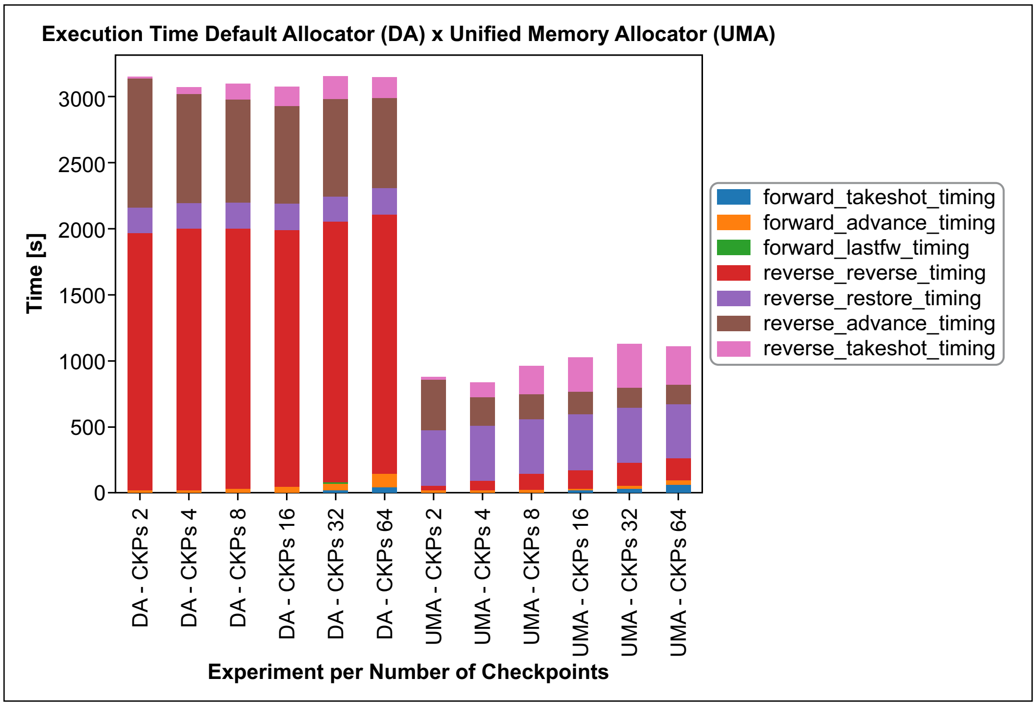
The following environmental variables were used to enable GPU execution with OpenACC:

- DEVITO_PLATFORM=nvidiaX;
- DEVITO_ARCH=nvc;
- DEVITO_LANGUAGE=openacc;

Results and Discussion

Based on the execution of the previously mentioned test, it was possible to evaluate the execution time of the algorithm. More specifically, the execution time of the forward and reverse methods, responsible for forward and reverse propagation, respectively. These are the methods that are most affected by the redundant transfer of data

Figure 3. Execution times of the overthrust script, varying the number of checkpoints, using the default DEVITO default allocator and the unified memory allocator.



when performing the checkpoint. Figure 3 shows the graph with the execution times of these functions. The unified memory allocation directly impacted the reverse propagation processing time, showing a significant decrease. This behavior is since using unified memory removes redundant copies of data made by the Devito tool. In addition, considering the entire process of calculating forward and reverse propagation, the application runtime with unified memory performs three times better than the standard implementation of the DEVITO tool.

Conclusion

DEVITO allocates data directly to the memory of CPU memory, making it necessary that, when running an application on the GPU, the data is transferred at each execution. As a result, when the operator is executed several times, data transfers become redundant, affecting application performance. A unified memory-based allocator was developed to solve this problem. Execution using this allocator makes data available to GPU and CPU, making redundant data transfers unnecessary. As a result, unified memory utilization showed significantly better results than the tool of the DEVITO default allocator.

References

1. Loubotin M. et al. DEVITO (v3.1.0): An embedded domain-specific language for finite differences and geophysical exploration. Geoscientific Model Development 2019;12(3):1165–1187.OpenACC Programming and Best Practices Guide. Available at: openacc.org. Accessed on: 20 Nov. 2022.
2. Luporini F. et al. Architecture and performance of DEVITO, a system for automated stencil computation. CoRR 2018;abs/1807.03032.
3. Harris M. Unified Memory for CUDA Beginners. Available at: <https://developer.nvidia.com/blog/unified-memory-cuda-beginners/>. 2017. Accessed on: 14 Dec. 2022.
4. Nisinho R, Loomis SHC. Cupy: A numpy-compatible library for nvidia gpu calculations. 31st Conference on Neural Information Processing Systems 2017;151(7).

A Comparison Between Different Solutions for the Marchenko Multiple Eliminations Scheme

Rodrigo S. Santos[1*], Daniel E. Revelo[1], Reynam C. Pestana[2], Marcelo S. Souza[1], Victor Koehne[1], Diego F. Barrera[1,2], Jonathas Maciel[1]

[1]SENAI CIMATEC, Supercomputing Center; [2]INCT-GP/CNPq; Salvador, Bahia Brazil

Seismic reflection data can be used to generate high-resolution in-depth images capable of facilitating, with high precision, the correct positioning of wells in hydrocarbon exploration and production. However, images produced by migrating seismic data are often contaminated by artifacts due to multiple internal reflections. Different schemes can be used to avoid problems caused by these noises and to attenuate them, such as the Marchenko multiple elimination scheme (MME). Various solutions based on the MME method have been proposed in the literature. Therefore, in this work, we explore the MME based on the least-squares schemes (LSMME), the MME as Neumann series approximation solution (NMME), and the MME based on beyond Neumann method (BNMME), and compare them in terms of effectiveness and efficiency in different numerical examples.

Keywords: Multiple Reflections. MME. LSMME. NEMME. BNMME.

Introduction

Generating in-depth images is a common step in seismic data processing flows. In this step, seismic reflection data combined with the velocity and density fields of the medium is used to build an image of the subsurface. These images would be used to make geological interpretations and the best decisions to put new wells to explore or improve the O&G production. The seismic data, represented by the set of reflections experienced by waves in the subsurface structure in the source-receiver path, contains both primary and multiple reflections. However, to image the subsurface, standard imaging methods such as reverse time migration (RTM) are based on the single-scattering assumption, i.e., the recorded seismic data do not include waves that are reflected more than once in the subsurface before reaching the receivers. Although the internal multiples generally have lower energy than the primary reflections, the single-scattering assumption can lead to the generation of false events in the seismic images, resulting in mistakes in geologic interpretation, as shown by Santos and colleagues [1]. Zhang and colleagues [2] modified the projected Marchenko equations presented by Neut and Wapenaar [3] to introduce the method known as Marchenko multiple eliminations (MME), which is a data-driven algorithm capable of removing internal multiples of all orders without velocity information or adaptive filter. Later, Zhang and Slob [4] used a set of measured laboratory data to evaluate the performance of the MME. The same authors [5] showed the first example of applying the MME on a field data set from the Norwegian North Sea, which validated the capabilities of the MME schemes and showed that it could effectively eliminate internal multiples. Aiming to explore the potential of the MME approach, many techniques went on to be developed, such as the transmission-compensated Marchenko multiple eliminations (T-MME) derived by Zhang and colleagues [6], which is a scheme that eliminates multiple internal reflections and compensates for two-way transmission losses contained in primary reflections. Afterward, Zhang and Slob [7] developed a fast implementation of the MME scheme that reduces its computational cost by an order of magnitude. The MME solution proposed by Zhang and colleagues [2] is based on the

Received on 16 December 2022; revised 7 January 2023.
Address for correspondence: Rodrigo S. Santos. Rua da Mangueira, No 60, Pero Vaz, Salvador, Bahia, Brazil. Zipcode: 40.335-755. E-mail: rodrigo.santana@fieb.org.br. DOI 10.34178/jbth.v6iSuppl. 1.268.

J Bioeng. Tech. Health 2023;6(Suppl.1):17-23.

Neumann series approximation (NEMME), so recently, Santos and colleagues [1] have proposed to formulate the MME scheme as a least-squares problem (LSMME), which averts the convergence criterion of the Neumann series approximation, and evaluates this approach in a complex 2D synthetic numerical example. Subsequently, to reduce the MME empirical scale factor dependence, Santos and colleagues [8] proposed an alternative solution based on the beyond Neumann scheme (BNMME) and showed that BNMME is more suitable in situations where it is difficult to obtain an ideal scale factor. The experiments developed by Zhang and colleagues [2] and Santos and colleagues [1,8] showed that when seismic data have previously gone through a high-quality pre-processing stage, i.e., deghosting, removal of free-surface multiples, and deconvolution with an estimated source wavelet, the schemes NEMME, LSMME, and BNMME successfully eliminates or attenuates multiple internal reflections. However, despite the experiments showing the power of the referred schemes in attenuating the internal multiples, their computational performance still needs to be evaluated. In this paper, we focus on evaluating this computational performance when the effectiveness of attenuating noise remains constant by testing NEMME, LSMME, and BNMME using a simple and another complex model.

Theory

The schemes MME, LSMME, and BNMME are based on the projected Marchenko equations for the single-sided reflection response [8-10]:

$$\overline{\mathrm{U}}^{-}(\mathbf{x}_0'',\mathbf{x}_0',\mathrm{t}) = \left(\Theta_{\mathrm{t}_2-\epsilon}^{\infty}\mathbf{R}\bar{\delta} + \Theta_{\mathrm{t}_2-\epsilon}^{\infty}\mathbf{R}\bar{\mathrm{v}}_{\mathrm{m}}^{+}\right)(\mathbf{x}_0'',\mathbf{x}_0',\mathrm{t}), \quad (1)$$

$$\begin{cases} \bar{\mathrm{v}}_{\mathrm{m}}^{+}(\mathbf{x}_0',\mathbf{x}_0'',\mathrm{t}) = \left(\Theta_{\epsilon}^{\mathrm{t}_2-\epsilon}\mathbf{R}^{*}\bar{\mathrm{v}}^{\,-}\right)(\mathbf{x}_0',\mathbf{x}_0'',\mathrm{t}) \\ \bar{\mathrm{v}}^{-}(\mathbf{x}_0',\mathbf{x}_0'',\mathrm{t}) = \left(\Theta_{\epsilon}^{\mathrm{t}_2-\epsilon}\mathbf{R}\bar{\delta} + \Theta_{\epsilon}^{\mathrm{t}_2-\epsilon}\mathbf{R}\bar{\mathrm{v}}_{\mathrm{m}}^{+}\right)(\mathbf{x}_0',\mathbf{x}_0'',\mathrm{t})' \end{cases} \quad (2)$$

in which $\mathbf{x_i}$ = ($\mathbf{x}_{\mathrm{H}}$, $\mathbf{z_i}$) and $\mathbf{x}_{\mathrm{H}}$ are the horizontal coordinates and $\mathbf{z_i}$ is the depth of an arbitrary boundary $\partial \boldsymbol{D}_i$, such that the acquisition surface $\partial \boldsymbol{D}_0$ will be defined by $\mathbf{x_0}$ = ($\mathbf{x}_{\mathrm{H}}$, $\mathbf{z_0}$) and *t* is the time. $\overline{\mathrm{U}}^{-}$ and $\bar{\mathrm{v}}^{\pm}$ represent the projected versions of the up-going Green's function and the down- and up-going focusing or filter function, respectively. The overline indicates that quantities have been convolved with the source wavelet. *t*₂ is the two-way travel time of the acquisition surface $\partial \boldsymbol{D}_0$ and a fictitious reflector at horizon $\partial \boldsymbol{D}_i$. The Θ is a truncation operator to exclude values outside the window (ε, t - ε), in which **ε** is a positive value to account for the finite bandwidth. ***R*** and ***R**** are multidimensional convolution and correlation operators [1,8]. Following Zhang and Slob [4], Equation (1) must be evaluated for each instant time *t*₂ and their value collected to be stored in a new function $\overline{\mathrm{R}}_{\mathrm{t}}$ containing only primary reflections as:

$$\overline{\mathrm{R}}_{\mathrm{t}}(\mathbf{x}_0'',\mathbf{x}_0',\mathrm{t}=\mathrm{t}_2) = \overline{\mathrm{U}}^{-}(\mathbf{x}_0'',\mathbf{x}_0',\mathrm{t}_2). \quad (3)$$

Neut and Wapenaar [3] showed that $\{\mathbf{R}\delta\}(\mathbf{x}_0'',\mathbf{x}_0',\mathrm{t})$ is equal to $\mathrm{R}(\mathbf{x}_0'',\mathbf{x}_0',\mathrm{t})$. From Equation (1), it is essential to note that to compute $\overline{\mathrm{U}}^{-}$, it is first necessary to obtain $\bar{\mathrm{v}}^{+}$.

Neumann Series Solution

As described by Santos and colleagues [1], if we organize the terms of Equation (2) and use the Neumann series expansion, we will obtain the following expression as a solution for Equation (2):

$$\bar{\mathrm{v}}_{\mathrm{m}}^{+}(\mathbf{x}_0',\mathbf{x}_0'',\mathrm{t}) = \left[\sum_{\mathrm{k}=1}^{\infty}\left(\Theta_{\epsilon}^{\mathrm{t}_2-\epsilon}\mathbf{R}^{*}\Theta_{\epsilon}^{\mathrm{t}_2-\epsilon}\mathbf{R}\right)^{\mathrm{k}}\bar{\delta}\right](\mathbf{x}_0',\mathbf{x}_0'',\mathrm{t}) \quad (4)$$

in which *k* represents the number of terms in the series that resemble the number of iterative iterations. The Marchenko multiple eliminations based on Equation (4) are conventionally named by the MME scheme. However, we must remember that the solution based on the Neumann series approximation converges only if

$$\left\|\Theta_{\epsilon}^{\mathrm{t}_2-\epsilon}\mathbf{R}^{*}\Theta_{\epsilon}^{\mathrm{t}_2-\epsilon}\mathbf{R}\right\| < 1 \quad (5)$$

LSMME

The LSMME approach treats the Marchenko multiple elimination problems as a linear system of the type $\mathbf{Ax} = \mathbf{b}$, where the solution $\mathbf{x}$ is the object of study. So, we can rewrite the down- and up-going filter functions of Equation (2) as the following linear system:

$$\begin{bmatrix} \mathbf{I} & -\Theta\mathbf{R} \\ -\Theta\mathbf{R}^{*} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{v}}^{-} \\ \bar{\mathbf{v}}_{\mathrm{m}}^{+} \end{bmatrix} = \begin{bmatrix} \Theta\bar{\mathbf{R}} \\ 0 \end{bmatrix} \tag{6}$$

The LSMME method obtains a solution for Equation (6) by formulating it as a least-squares problem (LS) and minimizing the sum of the squared residuals. To solve the linear system, we followed the approach implemented by Santos and colleagues [1] and applied the iterative method of Paige and Saunders, [11], which is based on a stable process.

BNMME

The BNMME scheme is based on solving the linear system in Equation (6) using the beyond Neumann method [12]. In this approach, the solution is obtained using the following recursive scheme:

$$\mathbf{x}_{\mathrm{n}} = \sum_{\mathrm{k}=0}^{\mathrm{n}} (\mathbf{I} - \alpha_{\mathrm{k}}\mathbf{A})^{\mathrm{k}} \alpha_{\mathrm{k}} \mathbf{b}, \tag{7}$$

in which α_k represents a relaxation parameter, which is updated according to

$$\alpha_{\mathrm{k}} = \frac{\mathbf{r}_{\mathrm{k}-1}^{\mathrm{H}} \mathbf{A} \mathbf{r}_{\mathrm{k}-1}}{\mathbf{r}_{\mathrm{k}-1}^{\mathrm{H}} \mathbf{A}^{\mathrm{H}} \mathbf{r}_{\mathrm{k}-1}} \tag{8}$$

with

$$\mathbf{r}_{\mathrm{k}} = (\mathbf{I} - \alpha_{\mathrm{k}-1}\mathbf{A})\mathbf{r}_{\mathrm{k}-1}. \tag{9}$$

Materials and Methods

In order to check the presented Marchenko multiple elimination schemes, we compared them in two numerical experiments. The objective was to evaluate the schemes in qualitative and quantitative terms in the multiple reflections attenuation process. Thus, we fixed the computational and experimental architecture for each experiment.

We first generated the acoustic impulse reflection response R with a finite-difference time-domain modeling code. We were involved with a Ricker wavelet with a 20 Hz central frequency to represent seismic data ($\bar{R}$). The R and $\bar{R}$ terms are used as input to the schemes, and the output seismic data $\bar{\mathrm{R}}_{\mathrm{t}}$ contains only primary seismic reflections.

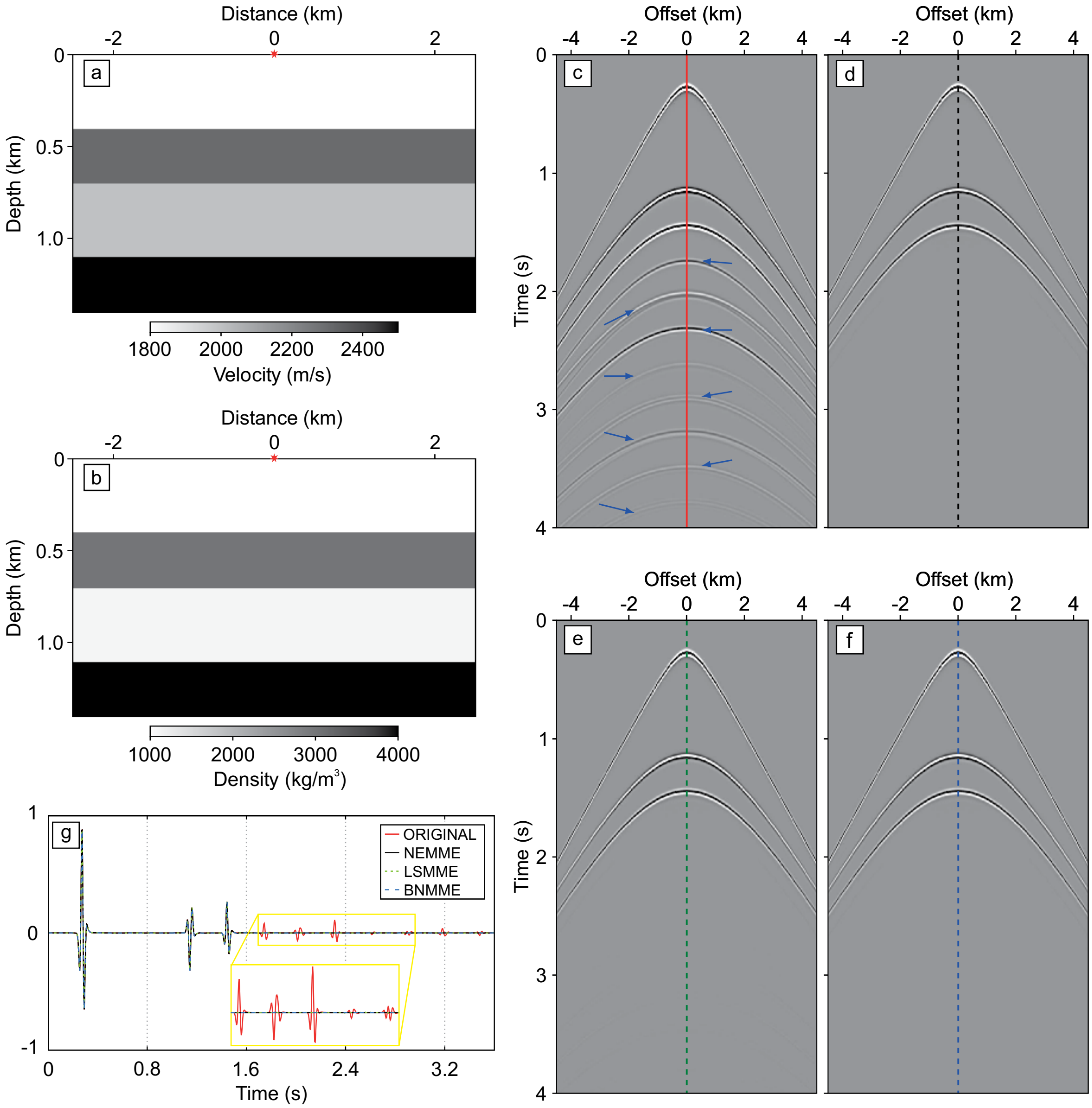
Results and Discussion

To evaluate the results, we chose the central shot gather (red star in Figures 1a, 1b, and 2a) for visualization of the effect of the multiple before and after the method application. In each situation, we compare the zero-offset trace to check if the phases and amplitudes of primary events were preserved.

Flat Layer Model

The first numerical example used is the flat layer model represented by the acoustic velocity and density values shown in Figures 1a and 1b, respectively. For this example, we have computed the reflection responses with 901 sources excited one by one and a fixed-spread array of 901 receivers with a spacing of 5 m located at the top of the model between -2.5 km and 2.5 km. The duration of each shot record is 4.0 s with a time sampling of 4 ms. In this experiment, 30 iterations were used for each scheme. Figure 1c shows the modeled synthetic data with labeled internal multiples interpretation (blue arrows). Figures 1d, 1e, and 1f show the output of NEMME, LSMME, and BNMME solutions, respectively, in which we observe that the events associated with multiple internal

Figure 1. (a) The velocity and (b) density values for the four-layer model with the red star indicates the shot position. The modeled reflection response in (c) and the retrieved data set using NEMME, LSMME, and BNMME, in (d), (e), and (f) respectively. The lines indicate the zero-offset traces selected to plot in Figure 1(g).

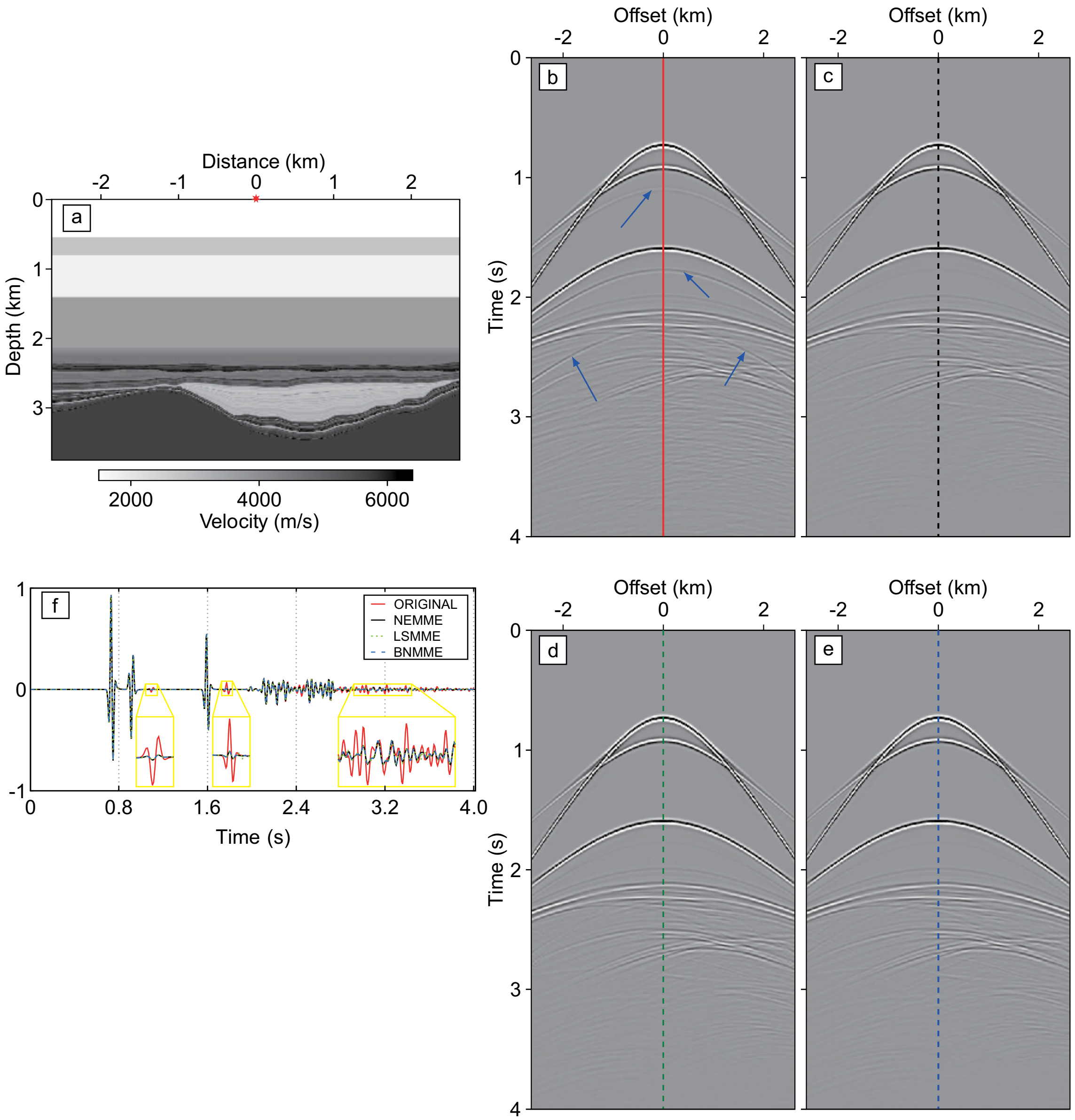


reflections signaled in Figure 1a were correctly attenuated. The dashed or solid lines in these figures indicate the selected zero-offset trace for analysis of the phases and amplitudes of the event. Figure 1g compares the zero-offset traces obtained by the three schemes, where we can see that the noise events were attenuated, preserving the amplitude and phase of the primary reflections. Figure 3 shows the computational cost for each scheme, having maintained a fixed computational structure to perform the experiments.

Santos Basin Model

The acoustic velocity model (Figure 2a) is named Santos basin model and simulates a realistic

Figure 2. (a) The velocity values for the Santos model where the red star indicates the shot position of a seismic source. The modeled reflection response in (a) and the retrieved data set using NEMME, LSMME, and BNMME, in (c), (d), and (e) respectively. The lines indicate the zero-offset traces selected to plot in Figure 2(f).



geological situation similar to those found in the sedimentary basins of the Brazilian continental shelf. For this numerical example, we computed the reflection responses with 526 sources excited one by one and a fixed-spread array of 526 receivers with a spacing of 10 m located at the model top between -2.625 km and 2.625 km. The duration of each shot record is 4.0 s with a sampling of 4 ms. In this experiment, we used 10 iterations for each scheme. Figure 2b shows the modeled synthetic data with labeled internal multiples interpretation (blue arrows), and Figures 2c, 2d, and 2e show the output of NEMME, LSMME, and BNMME solutions, respectively. The dashed or solid lines in the figures indicate the zero-offset traces selected to plot in Figure 2f. By analyzing the shot gathers of Figures 2d, 2e, and 2f and the plot of the zero-offset traces of Figure 2f, we observe that the events associated with the multiple internal reflections were correctly attenuated, similar to what happened in the previous experiment.

The computational time spent on each scheme is also shown in Figure 3.

Conclusion

This study compared the NEMME, LSMME, and BNMME approaches to treat multiple internal reflections. The presented results showed that such schemes successfully attenuate these coherent noises as long as the data is submitted to a quality pre-processing. When the results are compared, these methods have similar effectiveness in noise attenuation. By analyzing the computational cost of the three methods, we observed that NEMME and BNMME have similar efficiency, but they have shown better performance than the LSMME. These results can be used as a decision approach for choosing multiple treatment methods.

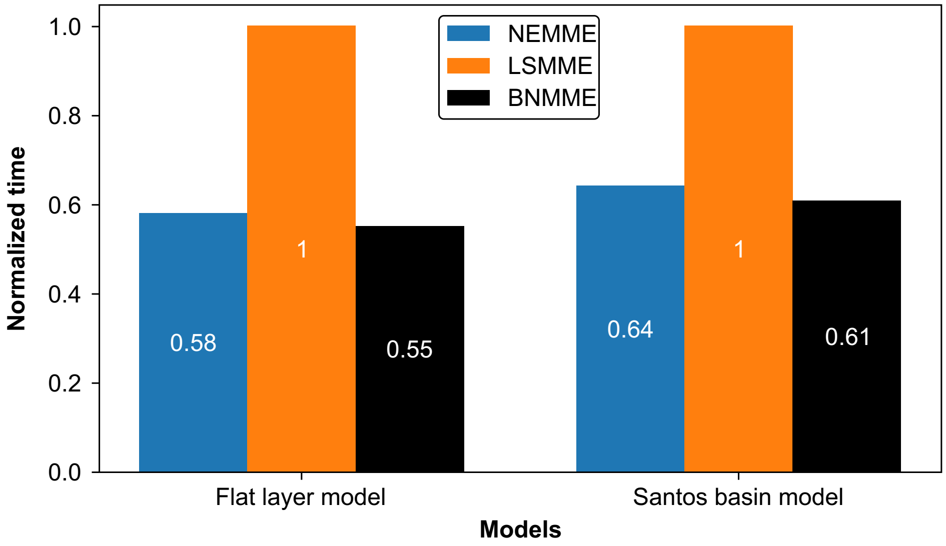
Acknowledgments

We thank the editors and reviewers for their constructive comments and suggestions. CENPES/ Petrobras and ANP supported this work through the Marchenko project at SENAI CIMATEC We want to thank FINEP for the support of the Supercomputing Center of SENAI CIMATEC.

References

1. Santos RS, Revelo DE, Pestana RC, KoehneV, Barrera DF, Souza MS, Silva A. An application of the Marchenko internal multiple elimination scheme formulated as a least-squares problem. Geophysics 2021;86(5):1–70.
2. Zhang L, Slob E, van der Neut J, Wapenaar K. Artifact-free reverse time migration. Geophysics 2018;83(5):A65–A68.
3. van der Neut J, Wapenaar K. Adaptive overburden elimination with the multidimensional Marchenko equation. Geophysics 2016;81(5):T265–T284.
4. Zhang L, Slob E. Free-surface and multiple internal eliminations in one step without adaptive subtraction. Geophysics 2019;84(1):A7–A11.
5. Zhang L, Slob E. Marchenko multiple eliminations of a laboratory example. Geophysical Journal International 2020;221(2):1138–1144.

Figure 3. Normalized time consumed to perform the NEMME, LSMME, and BNMME schemes in the flat layer and Santos basin models.



6. Zhang L, Thorbecke J, Wapenaar K, Slob E. Transmission compensated primary reflection retrieval in the data domain and consequences for imaging. Geophysics 2019;84(4):Q27–Q36.

7. Zhang L, Slob E. A fast algorithm for multiple elimination and transmission compensation in direct reflections. Geophysical Journal International 2020;221(1):371–377.

8. Santos R, Maciel J, Pestana R, Souza M, Koehne V, Revelo D, Barrera D, Paula R. A Marchenko multiple elimination solutions based on the beyond Neumann method. In: 83rd EAGE Annual Conference & Exhibition. European Association of Geoscientists & Engineers 2022:pp. 1–5.

9. Zhang L, Staring M. Marchenko scheme based internal multiple reflection elimination in acoustic wavefield. Journal of Applied Geophysics 2018;159:429–433.

10. Santos RS, Revelo DE, Pestana RC, Koehne V, Barrera DF, Souza MS. A least-squares based approach for the Marchenko internal multiple elimination scheme. In: SEG Technical Program Expanded Abstracts. Society of Exploration Geophysicists 2020:3184–3188.

11. Paige C, Saunders M. Algorithm 583. LSQR: Sparse linear equations and least squares problems. ACM Transactions on Mathematical Software 1982;8(2):195–209.

12. Maciel J, Pestana R, Barreira D. Stable solutions in Marchenko iterative scheme with beyond neumann. In: 82nd EAGE Annual Conference & Exhibition. European Association of Geoscientists & Engineers 2021:1–5.

Tuning a CPU-Based Stencil Computation in a DPC++ Multi-Device Environment

Tiago Conceição Oliveira[1*], Murilo Boratto[1], Antônio Horácio Rodrigues[1], Orlando Mota Pires[1], Leonardo Rodrigues Soares[1]
[1]SENAI CIMATEC, Supercomputing Center; Salvador, Bahia, Brazil

Reverse Time Migration (RTM) uses the finite-difference (FD) method to compute numerical approximations for the acoustic wave equation. It is a computational bottleneck for RTM applications and therefore needs to be optimized to guarantee timely results and efficiency when allocating resources for hydrocarbon exploration. This article describes our experience reengineering a migrated CUDA-based RTM code to SYCL into a multi-device RTM.

Keywords: Multi-Device RTM. OneAPI. SYCL. Heterogeneous Computing.

Introduction

RTM method, which stands for Reverse Time Migration, was first proposed by Baysal and colleagues [1] and McMechan [2] in 1983 but gained attention recently due to advances in computer capabilities [3]. It is a two-way wave equation that intends to build a high-quality and accurate image of the subsurface. Its bottleneck concerns to the need to compute two wave fields, one for the source (computed as a forward propagation) and other for the receiver (computed as a backward propagation) for each data point in the velocity model and for each shot (accurate data usually have thousands of shots) [4]. The Finite Difference Time Domain is a standard numerical solution used to model the wave propagation in RTM. The stencil data arrangement is used to compute this approximation at each grid point.

Despite the advantages intrinsic to the method, two significant computational difficulties characterize it: The high number of floating-point operations during the propagation step and the difficulty storing the wavefields in the main memory. Engineering seeks to explore both the intrinsic parallelism of tasks and the optimization of computational resources, designing solutions capable of running on different accelerated processing units, for example, to mitigate the effect of these problems. The optimization of this method represents an excellent economic advantage for exploration geophysics since it reduces the chances of errors in well-drilling.

Materials and Methods

Reengineering the DPC++ Based RTM Application for Multi-GPU Execution

In this paper, we consider a reference RTM algorithm written in C++. Algorithm 1 demonstrates the simplified execution of the RTM algorithm (Figure 1). The vector P store the state of pressure points in different time steps. The stencil computation needed for solving the finite-difference method dominates the total runtime; therefore, this is the main kernel to be accelerated on GPU devices. In this scenario, forward and backward propagation happens one after another in a serialized way. In these implementations, there is a dependence between the two propagation: during forward time steps, the source wavefields are stored and used in backward propagation to provide wavefields reconstruction.

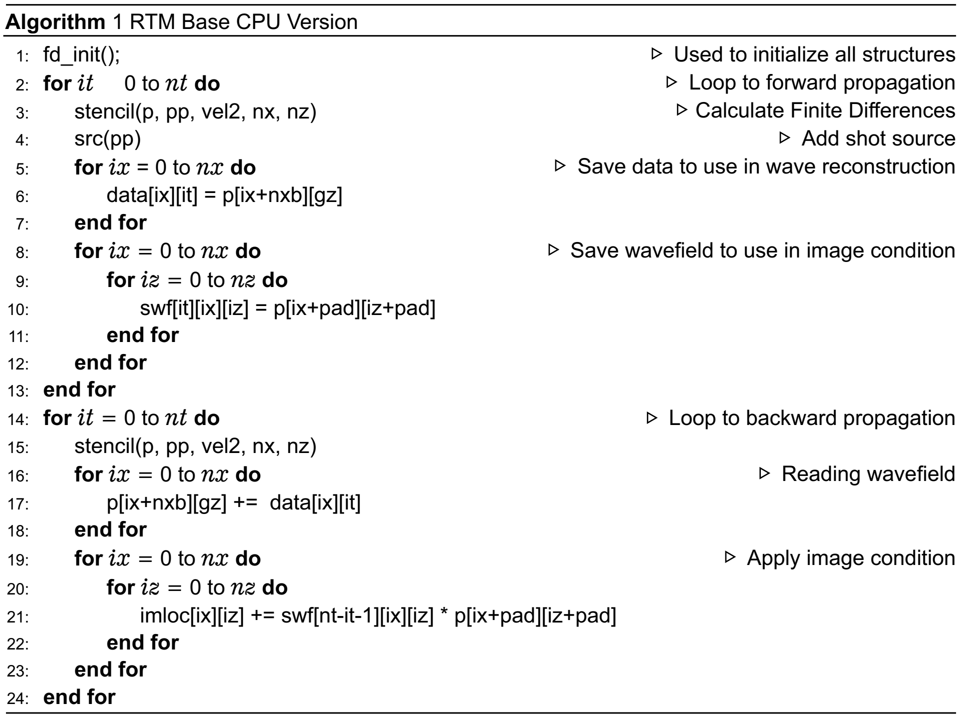
We proposed to explore a multi-GPU environment. So, we worked on the dependencies between forward and backward propagation to execute them simultaneously in two GPU

Received on 15 December 2022; revised 10 January 2023.
Address for correspondence: Tiago Conceição Oliveira. Av Comendador Franco, No. 8115, Condomínio Terra, apt 22, bloco A4. Zipcode: 81560001. Curitiba PR. E-mail: tiagocompuesc@gmail.com. DOI 10.34178/jbth.v6iSuppl. 1.269.

devices. This is only possible by building strategies to the previous model and storing the observed data to be used in the backward propagation without needing a reconstruction. Algorithm 2 presents a pseudocode describing the main steps for building RTM code without dependencies (Figure 2). The data used by RTM are generated previously and are not described in Algorithm 2. The method responsible to generates data is called modeling. Some steps were suppressed in Algorithm 2 and presented as functions, but those methods are described in Algorithm 1.

After building a code without dependencies, it is possible to propose an implementation of the RTM code using multiple devices. Figure 3 shows

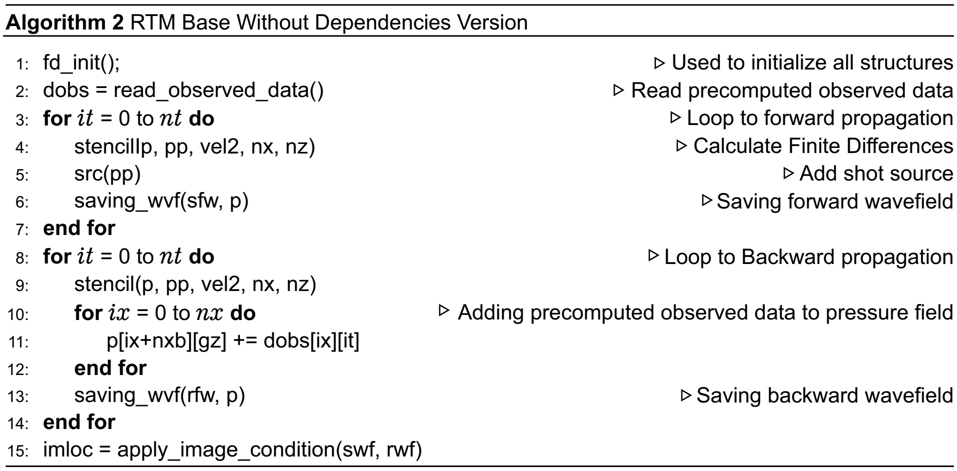
Figure 1. Algorithm 1.



Algorithm 1 RTM Base CPU Version

```
fd_init();                                        ▷ Used to initialize all structures
for $it$   0 to $nt$ do                           ▷ Loop to forward propagation
    stencil(p, pp, vel2, nx, nz)                  ▷ Calculate Finite Differences
    src(pp)                                       ▷ Add shot source
    for $ix$ = 0 to $nx$ do                       ▷ Save data to use in wave reconstruction
        data[ix][it] = p[ix+nxb][gz]
    end for
    for $ix$ = 0 to $nx$ do                       ▷ Save wavefield to use in image condition
        for $iz$ = 0 to $nz$ do
            swf[it][ix][iz] = p[ix+pad][iz+pad]
        end for
    end for
end for
for $it$ = 0 to $nt$ do                           ▷ Loop to backward propagation
    stencil(p, pp, vel2, nx, nz)
    for $ix$ = 0 to $nx$ do                       ▷ Reading wavefield
        p[ix+nxb][gz] +=  data[ix][it]
    end for
    for $ix$ = 0 to $nx$ do                       ▷ Apply image condition
        for $iz$ = 0 to $nz$ do
            imloc[ix][iz] += swf[nt-it-1][ix][iz] * p[ix+pad][iz+pad]
        end for
    end for
end for
```

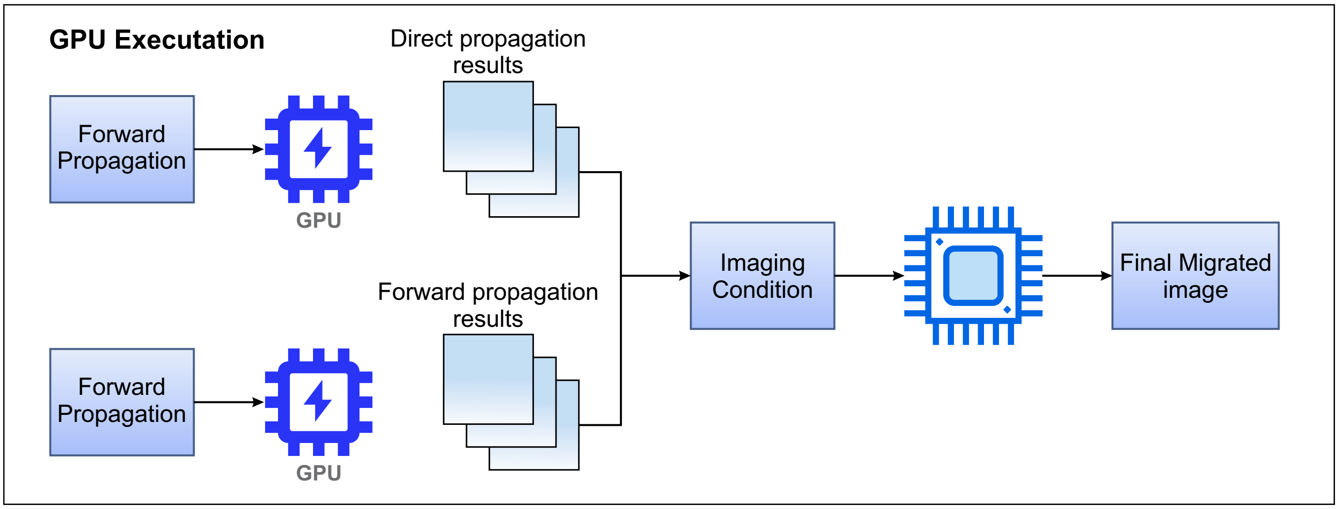
Figure 2. Algorithm 2.



Algorithm 2 RTM Base Without Dependencies Version

```
fd_init();                                        ▷ Used to initialize all structures
dobs = read_observed_data()                       ▷ Read precomputed observed data
for $it$ = 0 to $nt$ do                           ▷ Loop to forward propagation
    stencilIp, pp, vel2, nx, nz)                  ▷ Calculate Finite Differences
    src(pp)                                       ▷ Add shot source
    saving_wvf(sfw, p)                            ▷ Saving forward wavefield
end for
for $it$ = 0 to $nt$ do                           ▷ Loop to Backward propagation
    stencil(p, pp, vel2, nx, nz)
    for $ix$ = 0 to $nx$ do                       ▷ Adding precomputed observed data to pressure field
        p[ix+nxb][gz] += dobs[ix][it]
    end for
    saving_wvf(rfw, p)                            ▷ Saving backward wavefield
end for
imloc = apply_image_condition(swf, rwf)
```

Figure 3. Multi-GPU without dependence 2D-RTM flowchart.



The figure shows a structure of independent foward propagation and backward propagation computation in multiple devices. Synchronization is made only in the CPU.

a simplified version of the new code structure. The approach is that propagations are independently processed, and synchronization between devices is only necessary at the end of the operation. Finally, the image condition is applied. The result of both propagations, forward and backward, is the RTM migration image.

One way to control the flow between devices is to use threads. In this approach, we use threads to process propagations in parallel. A parallel zone starts with two threads for executing and synchronizing the RTM model. After the threading process on the GPU ends, the data return to the CPU. After parallel threads, the CPU computes the image condition, as described in Algorithm 3 (Figure 4).

The SYCL Thread Hierarchy

The thread hierarchy exploration aims to maximize the occupancy of the GPU resources. In an SYCL kernel, the programmer can affect the work distribution by structuring the kernel with proper workgroup size and sub-group size and organizing the work items for efficient vector execution (Figure 5). Writing efficient vector kernels is crucial for high performance on GPU.

Figure 4. Algorithm 3.

Algorithm 3 RTM Base CPU Version

```
fd_init();                                  ▷ Used to initialize all structures
dobs = read_observed_data()                 ▷ These data is the input for the backward propagation
init_GPUs()                                 ▷ Init all structures to use Multi-GPU
omp_set_num_threads(2)                      ▷ Each propagation runs in a different thread
#pragma omp parallel{
tid = get_thread_num();
if $tid$ == 0 then
    forward(P, PP, Vel, swf, tid)           ▷ Run in GPU 1
else
    backward(PR,PPR, Vel, rwf, tid)         ▷ Run in GPU 2
end if
}
#pragma omp barrier
imloc = apply_image_condition(swf, rfw)
```

Laplacian Kernel

Laplacian is the core of the RTM algorithm, which is highly time-consuming in the application. Figure 6 shows how the prominent part of the Laplacian was implemented using the SYCL thread hierarchy. This implementation focuses on workgroup and sub-group size selection. SYCL does not provide a mechanism to set the number of threads directly in a workgroup. However, it can use workgroup size and SIMD sub-group size to set the number of threads. Thread contexts are easy to utilize, starting with selecting the number of threads in a workgroup (Figure 6, lines 4 to 9).

Results and Discussion

Computational Results

In experiments using base serial code, execution will use only one CPU core. We use two devices in experiments using the GPU version of the code. Table 1 describes a GPU device and a CPU device. An Intel® DevCloud node was chosen for the preliminary experiments. Table 1 shows the hardware description of the node.

Reference Input Data

Since the vector P represents the pressure field, its characteristics are directly related to the characteristics of the initial velocity model. Both are represented as 2-D matrices with the exact dimensions. We begin considering three seismic velocity models illustrated by Figure 7 with *nx* = 151 and *nz* = 151, Figure 8 with *nx* = 369 and *nz* = 375, Figure 9 with *nx* = 351 and *nz* = 367, as the base models for execution and migration evaluation.

The models presented here were our reference to build matrix P, which is the input parameter of the function that performs the stencil.

Experimental Results

The results of the experiments are available in Table 2. A comparison of the serial and parallel versions is also available. As we can see, the multi-device RTM is faster for all models tested.

Figure 10 is a serial reference for a parallel approach, while Figure 11 shows an image migration using the multi-device RTM code for a 3-layer model (Figure 7). Figure 12 is a subtraction for Figures 10 and 11 using the *farith package*. Figure 14 shows an image migration using the multi-device RTM code for the *SPluto model* (Figure 9).

Figure 13 is a serial reference for a parallel approach, and Figure 15 is a subtraction for Figures 13 and 14 using the *SPluto model*. Figures 12 and 15 show that the migrated multi-device image version does not differ significantly from the CPU-migrated image version.

Figure 5. The relationship diagram among ND-Range, work-group, sub-group, and work-item.

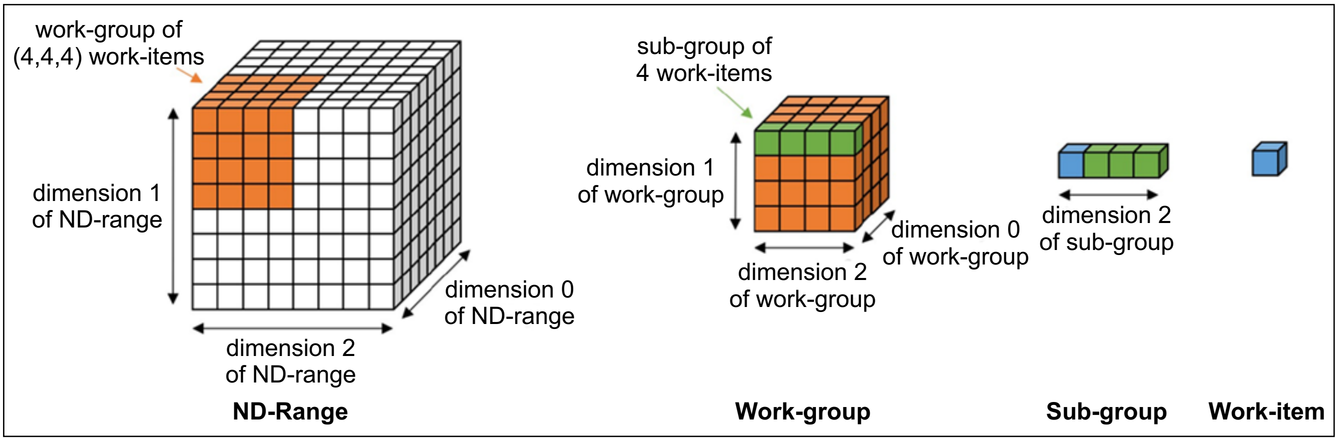
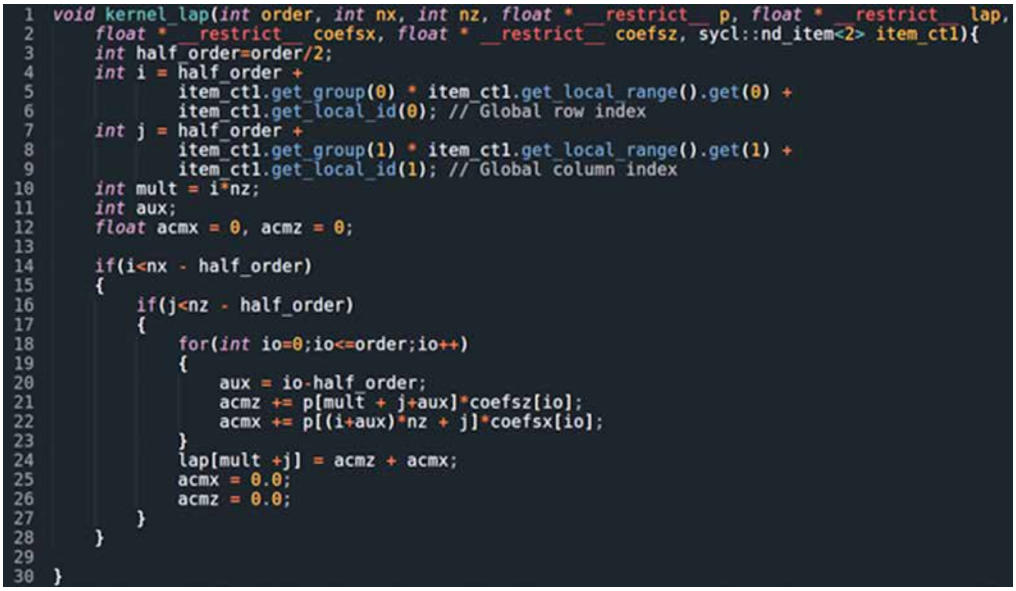


Figure 6. Kernel for laplacian on GPU using SYCL with multiple thread hierarchy.



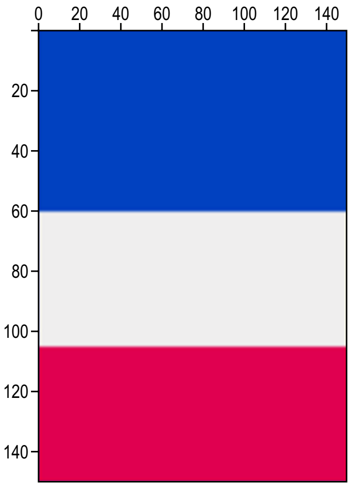
```
void kernel_lap(int order, int nx, int nz, float * __restrict__ p, float * __restrict__ lap,
    float * __restrict__ coefsx, float * __restrict__ coefsz, sycl::nd_item<2> item_ct1){
    int half_order=order/2;
    int i = half_order +
            item_ct1.get_group(0) * item_ct1.get_local_range().get(0) +
            item_ct1.get_local_id(0); // Global row index
    int j = half_order +
            item_ct1.get_group(1) * item_ct1.get_local_range().get(1) +
            item_ct1.get_local_id(1); // Global column index
    int mult = i*nz;
    int aux;
    float acmx = 0, acmz = 0;

    if(i<nx - half_order)
    {
        if(j<nz - half_order)
        {
            for(int io=0;io<=order;io++)
            {
                aux = io-half_order;
                acmz += p[mult + j+aux]*coefsz[io];
                acmx += p[(i+aux)*nz + j]*coefsx[io];
            }
            lap[mult +j] = acmz + acmx;
            acmx = 0.0;
            acmz = 0.0;
        }
    }

}
```

The figure shows a snippot for forward propagation and nackward propagation computation in multiple devices.

Table 1. DevCloud node description.

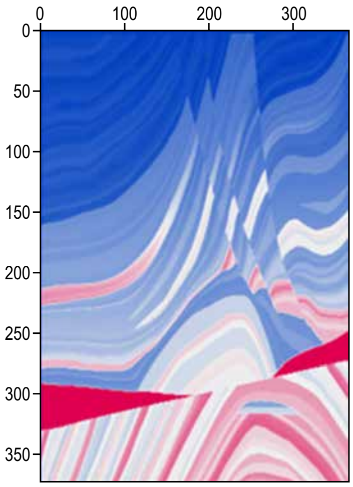
	Description
CPU	Intel(R) Xeon(R) Gold 6336Y CPU @ 2.40GHz
GPU	Intel® HD Graphics P630
Memory	128 GB
GCC	9.4.0-1ubuntu1~20.04.1
DPC++	2022.0.0 (2022.0.0.20211123)

Figure 7. seismic velocity 1.



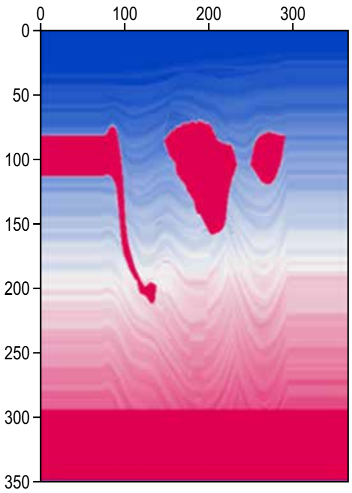
nx = 151 and *nz* = 151

Figure 8. seismic velocity 2.



nx = 369 and *nz* = 375

Figure 9. seismic velocity 3.



nx = 351 and *nz* = 367

Table 2. A CPU serial RTM and multi-device parallel RTM execution time comparison on DeCloud environment.

Velocity Model	Serial Time(s)	Parallel Time(s)
3 layer	4.76	1.08
Marmousi	42.93	3.6
SPluto	54.91	5.5

Figure 10. CPU-based RTM migration for 3 layer models.

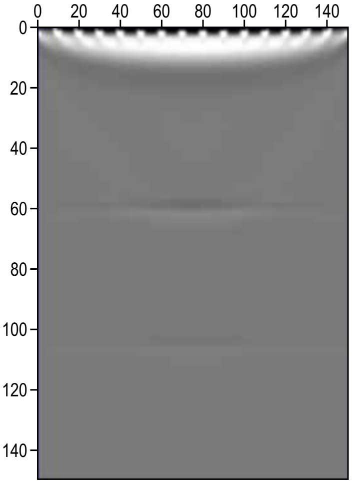


Figure 11. Muti-device-based RTM migration for 3 layer models.

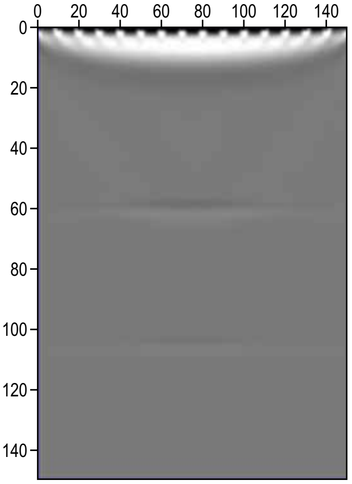


Figure 12. Difference between CPU and Multi-device RTM migration for 3 layer models.

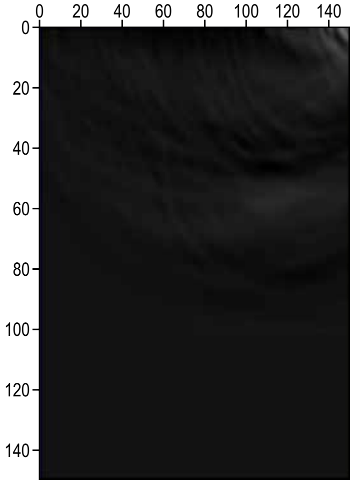


Figure 13. CPU-based RTM migration for the SPluto model.

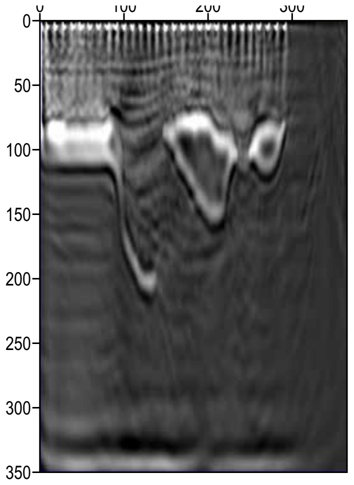


Figure 14. Multi-device RTM migration for the SPluto model.

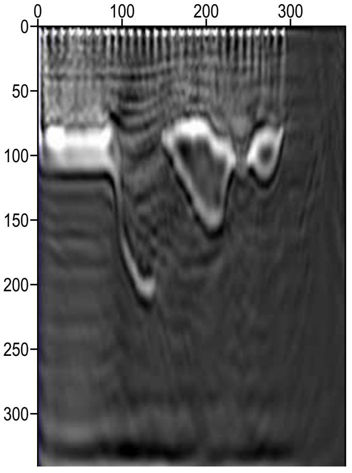
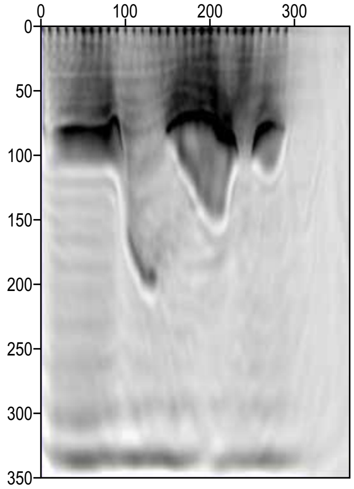


Figure 15. Difference between CPU and Multi-Device RTM migration for SPluto model.



Conclusion

The first step involves migrating a serial base RTM, already changed to provide simultaneous forward/backward propagation, to DPC++ and the improvements related to the DPC++ source code implementation. This step was achieved, as shown in the image migration in the results section. In the second step, researchers focused on possible optimizations to achieve those objectives: the application was rewritten, focusing on thread hierarchy, as shown in Figure 6.

Finally, the Multi-devices RTM application was successfully developed and tested. Intel® tools also helped to decide what resource to use and correctness check. The Multi-devices RTM code is a small workload but highly time-consuming. Further work could explore larger workloads, aiming to use the whole GPU memory and the main memory.

Acknowledgments

We thank the editors and reviewers for their constructive comments and suggestions. We would like to thank FINEP for the support of the Supercomputing Center of SENAI CIMATEC.

References

1. Baysal E, Kosloff DD, Sherwood JW. Reverse time migration. Geophysics 1983;48(11):1514–1524.
2. McMechan GA. Migration by extrapolation of time-dependent boundary values. Geophysical Prospecting 1983;31(3):413–420.
3. Yang P, GaoJ, WangB. RTM using effective boundary saving: A staggered grid GPU implementation. Computers & Geosciences 2014;68:64–72.
4. Zhang L, Slob E. Free-surface and internal multiple elimination in one step without adaptive subtraction. Geophysics 2019;84(1):A7–A11.
5. Zhang L, Slob E. Marchenko multiple elimination of a laboratory example. Geophysical Journal International 2020;221(2):1138–1144.

Instructions for Authors

The Authors must indicate in a cover letter the address, telephone number and e-mail of the corresponding author. The corresponding author will be asked to make a statement confirming that the content of the manuscript represents the views of the co-authors, that neither the corresponding author nor the co-authors have submitted duplicate or overlapping manuscripts elsewhere, and that the items indicated as personal communications in the text are supported by the referenced person. Also, the protocol letter with the number should be included in the submission article, as well as the name of sponsors (if applicable).

Manuscripts may be submitted within designated categories of communication, including:

- Original basic or clinical investigation (original articles on topics of broad interest in the field of bioengineering and biotechnology applied to health). We particularly welcome papers that discuss epidemiological aspects of international health, clinical reports, clinical trials and reports of laboratory investigations.
- Case presentation and discussion (case reports must be carefully documented and must be of importance because they illustrate or describe unusual features or have important practice implications).
- Brief reports of new methods or observations (short communications brief reports of unusual or preliminary findings).
- State-of-the-art presentations (reviews on protocols of importance to readers in diverse geographic areas. These should be comprehensive and fully referenced).
- Review articles (reviews on topics of importance with a new approach in the discussion). However, review articles only will be accepted after an invitation of the Editors.
- Letters to the editor or editorials concerning previous publications (correspondence relating to papers recently published in the Journal, or containing brief reports of unusual or preliminary findings).
- Editor's corner, containing ideas, hypotheses and comments (papers that advance a hypothesis or represent an opinion relating to a topic of current interest).
- Innovative medical products (description of new biotechnology and innovative products applied to health).
- Health innovation initiatives articles (innovative articles of technological production in Brazil and worldwide, national policies and directives related to technology applied to health in our country and abroad).

The authors should checklist comparing the text with the template of the Journal.

Supplements to the JBTH include articles under a unifying theme, such as those summarizing presentations of symposia or focusing on a specific subject. These will be added to the regular publication of the Journal as appropriate, and will be peer reviewed in the same manner as submitted manuscripts.

Statement of Editorial Policy

The editors of the Journal reserve the right to edit manuscripts for clarity, grammar and style. Authors will have an opportunity to review these changes prior to creation of galley proofs. Changes in content after galley proofs will be sent for reviewing and could be required charges to the author. The JBTH does not accept articles which duplicate or overlap publications elsewhere.

Peer-Review Process

All manuscripts are assigned to an Associate Editor by the Editor-in-Chief and Deputy Editor, and sent to outside experts for peer review. The Associate Editor, aided by the reviewers' comments, makes a recommendation to the Editor-in-Chief regarding the merits of the manuscript. The Editor-in-Chief makes a final decision to accept, reject, or request revision of the manuscript. A request for revision does not guarantee ultimate acceptance of the revised manuscript.

Manuscripts may also be sent out for statistical review ou *ad hoc* reviewers. The average time from submission to first decision is three weeks.

Revisions

Manuscripts that are sent back to authors for revision must be returned to the editorial office by 15 days after the date of the revision request. Unless the decision letter specifically indicates otherwise, it is important not to increase the text length of the manuscript in responding to the comments. The cover letter must include a point-by-point response to the reviewers and Editors comments, and should indicate any additional changes made. Any alteration in authorship, including a change in order of authors, must be agreed upon by all authors, and a statement signed by all authors must be submitted to the editorial office.

Style

Manuscripts may be submitted only in electronic form by www.jbth.com.br. Each manuscript will be assigned a registration number, and the author notified that the manuscript is complete and appropriate to begin the review process. The submission file is in OpenOffice, Microsoft

Word, or RTF document file format for texts and JPG (300dpi) for figures.

Authors must indicate in a cover letter the address, telephone number, fax number, and e-mail of the corresponding author. The corresponding author will be asked to make a statement confirming that the content of the manuscript represents the views of the co-authors, that neither the corresponding author nor the co-authors have submitted duplicate or overlapping manuscripts elsewhere, and that the items indicated as personal communications in the text are supported by the referenced person.

Manuscripts are to be typed as indicated in Guide for Authors, as well as text, tables, references, legends. All pages are to be numbered with the order of presentation as follows: title page, abstract, text, acknowledgements, references, tables, figure legends and figures. A running title of not more than 40 characters should be at the top of each page. References should be listed consecutively in the text and recorded as follows in the reference list, and must follow the format of the National Library of Medicine as in Index Medicus and "Uniform Requirements for Manuscripts Submitted to Biomedical Journals" or in "Vancouver Citation Style". Titles of journals not listed in Index Medicus should be spelled out in full.

Manuscript style will follow accepted standards. Please refer to the JBTH for guidance. The final style will be determined by the Editor-in-Chief as reviewed and accepted by the manuscript's corresponding author.

Approval of the Ethics Committee

The JBTH will only accept articles that are approved by the ethics committees of the respective institutions (protocol number and/or approval certification should be sent after the references). The protocol number should be included in the end of the Introduction section of the article.

Publication Ethics

Authors should observe high standards with respect to publication ethics as set out by the International Committee of Medical Journal Editors (ICMJE). Falsification or fabrication of data, plagiarism, including duplicate publication of the authors' own work without proper citation, and misappropriation of the work are all unacceptable practices. Any cases of ethical misconduct are treated very seriously and will be dealt with in accordance with the JBTH guidelines.

Conflicts of Interest

At the point of submission, each author should reveal any financial interests or connections, direct or indirect, or other situations that might raise the question of bias in the work reported or the conclusions, implications, or opinions stated - including pertinent commercial or other sources of funding for the individual author(s) or for the associated department(s) or organizations(s), and personal relationships. There is a potential conflict of interest when anyone involved in the publication process has a financial or other beneficial interest in the products or concepts mentioned in a submitted manuscript or in competing products that might bias his or her judgment.

Material Disclaimer

The opinions expressed in JBTH are those of the authors and contributors, and do not necessarily reflect those of the SENAI CIMATEC, the editors, the editorial board, or the organization with which the authors are affiliated.

Privacy Statement

The names and email addresses entered in this Journal site will be used exclusively for the stated purposes of this journal and will not be made available for any other purpose or to any other party.

Sponsors:

MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÕES

Finep

SENAI CIMATEC | 20 anos

PRH 27.1

UFRN
UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

Laboratório
Nacional de
Computação
Científica
UNIDADE DE PESQUISA DO MCTI

prh
Programa de
Recursos Humanos
da ANP
GESTÃO: Finep