Hindawi
International Journal of Geophysics
Volume 2018, Article ID 2913591, 8 pages
https://doi.org/10.1155/2018/2913591

Research Article

SEISGAMA: A Free C# Based Seismic Data Processing Software Platform

Theodosius Marwan Irnaka, Wahyudi Wahyudi, Eddy Hartantyo, Adien Akhmad Mufaqih, Ade Anggraini, and Wiwit Suryanto

Seismology Research Group, Physics Department, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Sekip Utara Bulaksumur, Yogyakarta 55281, Indonesia

Correspondence should be addressed to Wiwit Suryanto; ws@ugm.ac.id

Received 4 July 2017; Accepted 24 December 2017; Published 30 January 2018

Academic Editor: Filippos Vallianatos

Seismic reflection is one of the most popular methods in geophysical prospecting. Nevertheless, obtaining high resolution and accurate results requires a sophisticated processing stage. There are many open-source seismic reflection data processing software programs available; however, they often use a high-level programming language that decreases its overall performance, lacks intuitive user-interfaces, and is limited to a small set of tasks. These shortcomings reveal the need to develop new software using a programming language that is natively supported by Windows® operating systems, which uses a relatively medium-level programming language (such as C#) and can be enhanced by an intuitive user interface. SEISGAMA was designed to address this need and employs a modular concept, where each processing group is combined into one module to ensure continuous and easy development and documentation. SEISGAMA can perform basic seismic reflection processes. This ability is very useful, especially for educational purposes or during a quality control process (in the acquisition stage). Those processes can be easily carried out by users via specific menus on SEISGAMA's main user interface. SEISGAMA has been tested, and its results have been verified using available theoretical frameworks and by comparison to similar commercial software.

1. Introduction

Geophysics is a branch of physics that studies physical properties of the earth, such as mass density, seismic wave speed, electrical resistance, and magnetic properties [1–3] from surface measurements. These physical properties can be further used to image subsurface geological conditions of the earth. One of the most popular methods in geophysics is the seismic reflection method. This method is very sensitive to a rock's acoustic impedance variation, which is characterized by seismic waves reflected in each interface of the subsurface layer or structure. Acoustic impedance is denoted by the product of rock density and seismic wave speed within a certain medium. Seismic reflection is very popular, especially in the oil and gas industry, because this method yields high resolution subsurface images of geological structures [4]. The seismic reflection method is also useful in engineering, environmental, geohazard, and ground-water exploration, as discussed by Steeples and Miller [5].

Recently, a large number of open-source software programs have emerged that can be used to process seismic reflection data; however, each program has its own disadvantages. For example, Seismic Un∗x [6] has a really good ability to process seismic reflection data, but it has no intuitive user interface and depends on a command-line interface. Templeton and Gough [7] developed a web-based seismic data processing tool based on the Seismic Un∗x platform. Through this web service, users can fill-out the form provided by the server, and the input data will be queued on a parallel processing scheduler application. In this sense, the online service is a good concept, but the performance is limited to user interaction via a form-based input with limited resources on the server side. DSISoft, which was developed by Beaty et al. [8], is a vertical seismic profiling data processor based on the MATLAB® programming language. MATLAB is a popular as programming language because it has a library of common mathematical and engineering functions. Unfortunately, it has a large computational cost as data size increases and is thus inefficient compared to a lower level programming language (e.g., C, Fortran, and C#). A comparison between MATLAB and C, made by Andrews [9], shows significant speed differences; He determines that a C program is up to 546 times faster than a MATLAB program. A more comprehensive seismic reflection data processing implementation in MATLAB was developed by Mousa and Al-Shuhail, [10]. Another example is SEISPRHO [11], software developed in Lazarus Pascal Language, which can be used to interpret seismic reflection and is enhanced with simple processing tools. This software is mainly developed for interpretation purposes.

The recent popularity of Python, another high-level programming language that is free and reliable, has many features to handle numerical problems and has high quality graphics. It is increasingly being used to develop geophysical data processing software [12, 13]. One of the advantages of Python is its ability to be installed on a server so that the application can be accessed remotely (e.g., [14]). Schwehr [15] has developed an application called *seismic-py*, which provides an infrastructure for creating and managing a Python library for each seismic data format. Rizvandi et al. [16] detailed the performance of an implementation of the Prestack Kirchhoff Time Migration in Python. There is currently no complete application that is built entirely with Python (i.e., ready to use without combining the Python library to another library), as this would require specialized Python programming skills, at least the basic ability to program in Python.

To the best of our knowledge, there is no other free seismic reflection data processing software program built with the C# programming language that has a complete seismic processing toolbox, can process raw data from field recordings, and includes an intuitive and appealing user interface. The C# programming language was considered due to its native compatibility with the most popular operating system, that is, Windows. Netmarketshare states that approximately 81% of desktop users have a Windows XP or newer operating system installed on their computers [17]. The C# programming language is officially supported and periodically updated by Windows and included on a .NET (dot NET) platform [18, 19]. Developing applications natively supported on Windows operating systems ensures long-term and continuous software development. Enhanced with an intuitive user interface, SEISGAMA can easily be used by beginner or advanced users, and it is particularly useful for undergraduate geophysics students as they are learning processes. The need for reliable and free seismic processing software has long been awaited, especially since the price of oil became variable. This software may ensure that oil exploration can be resumed with lower associated costs.

2. Methods

SEISGAMA's development is divided into several development sections: (1) basic data processing, (2) intermediate data processing, and (3) advanced processing. This paper reports only the basic processing aspects of reflection seismic methods, and the advanced processing aspects will be discussed separately in another paper. The basic data processor that was developed in this research consists of amplitude correction, muting, F-x and F-K domain transform, velocity analysis, normal moveout (NMO) correction, and traces stacking. These processing steps combine into one trace with a high signal to noise ratio (SNR).

A more detailed explanation of the processing steps follows.

2.1. Amplitude Correction. Amplitude correction was performed to compensate seismic signal weakening, which decreases exponentially as a function of time due to attenuation and geometrical spreading. Campillo et al. [20] explains that this phenomenon often holds in some regional seismic phases. The phenomenon is also described by geometrical spreading [21]. Amplitude attenuation may be exploited via an exponential or oscillatory equation; therefore, the amplitude correction can be formulated as an exponential equation (1). f_{AGC} is seismic data after amplitude correction, $f(t)$ is the data before amplitude correction, and β is an exponential constant. This equation is an example of amplitude correction for a data-independent method. After amplitude correction, the signal has amplitude with no attenuation as the recording time is increased.

$$f_{\text{AGC}}(t) = f(t)\, e^{\beta t}. \tag{1}$$

2.2. Muting. The muting process removes unused parts of the seismic recording. This process is very important because in the seismic reflection method, only the reflection signal will be considered, and other signals, such as those from direct waves, ground-roll, or refraction, must be removed. The muting process is carried out by defining the area to be removed first, and then, SEISGAMA will automatically make the amplitude of that region equal to zero.

2.3. F-x and F-K Transformation. Seismic data is typically represented in time-domain versus (range or) distance (t-x). The F-x transformation is carried out by using a 1D Fourier transform and is applied to the transformation for each seismic trace in the data. This will transform the time series data into frequency domain data. The results are then displayed in terms of distance (X-axis) and frequency (Y-axis). The Fourier transformation is expressed by

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-i\omega t} dt = FT\left(f(t)\right). \tag{2}$$

The F-K transform utilizes the 2D Fourier transformation to transform a time-domain version of each seismic trace into the frequency domain in $1/s$ and transform the spatial domain of data into wave number $(1/m)$. In this transformation, seismic data in time (t) versus distance (x) will be transformed into frequency (f) versus wave number (k). The 2D Fourier transform is an extension of the 1D Fourier transformation and expressed by

$$\begin{aligned} F(\omega) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t, x)\, e^{-i\omega t} e^{-ikt} dt\, dx \\ &= \text{FT2D}\left(f(t, x)\right). \end{aligned} \tag{3}$$

2.4. Velocity Analysis. The reflection from a horizontal reflector of a seismic wave on the record for a particular seismic source will follow a parabolic equation

$$t(x) = \sqrt{t_0^2 + \left(\frac{x}{v}\right)^2}. \tag{4}$$

These characteristics are used to estimate the wave propagation speed by assuming the velocity of the medium. Parabolic response patterns can be detected using a shot-gather-based seismic semblance calculation (NE), as noted by Mousa and Al-Shuhail [10].

A seismic semblance is formulated as the quotient between stacked and prestacked energy subsequently multiplied by the number of seismic traces (7) for each time window (w). Time window parameters, with width w, are required to perform an NE calculation. Here, N is the number of samples at the time window, w; M is the number of the traces on seismic trace collections; A_{ij} is the amplitude at time i and trace j; Es is stacked energy; and Eu is prestacked energy.

$$\mathrm{Es} = \sum_{i=1}^{N}\left(\sum_{j=1}^{M} A_{ij}\right)^2, \tag{5}$$

$$\mathrm{Eu} = \sum_{i=1}^{N}\sum_{j=1}^{M} A_{ij}^2, \tag{6}$$

$$\mathrm{NE} = \frac{\mathrm{Es}}{M \times \mathrm{Eu}}. \tag{7}$$

2.5. Normal Moveout Correction. A normal moveout (NMO) correction is carried out to restore the nonzero offset seismic reflectors to zero offset. The correction is done after we obtain t_0 and v_{NMO} for each reflector during velocity analysis process. The time delay after NMO correction is defined in (9). A NMO correction is suitable for a reflector having a slope of less than 15°. When a reflector has higher slope than that threshold, another correction (Dip Move Out correction) is required.

$$\Delta t_{\mathrm{NMO}}(x) = t(x) - t_0, \tag{8}$$

$$t_{\mathrm{NMO}}(x) \approx \frac{x^2}{2t_0 v_{\mathrm{NMO}}^2}. \tag{9}$$

NMO corrections can be categorized by three classifications: (1) if the NMO velocity (v_{NMO}) is too high, then it will have a shape of a downward convex reflector, (2) if the v_{NMO} is correct, then the reflector will be flat on $t = t_0$, and (3) if v_{NMO} is too low, then the reflector will be an upward convex reflector. During this process, the NMO correction of seismic signals often causes a distortion in the frequency domain. To overcome this problem, a stretching factor (10) is a provision threshold that is required. If a signal is found to have a stretching factor that exceeds the threshold, then a muting correction will be performed.

$$S_{\mathrm{NMO}}(x) = \frac{\Delta t_{\mathrm{NMO}}(x)}{t_0}. \tag{10}$$

2.6. Stacking. Stacking is a process of merging many seismic traces that are already in a state of zero offset. The stacking process is done by quantifying the average amplitude of all signals (11). This stacking process is expected to improve SNR (signal to noise ratio) to factor $\sqrt{M}$ [10].

$$t(x) = \frac{\sum_{i=1}^{M} A_i(x)}{M}. \tag{11}$$

3. Implementation

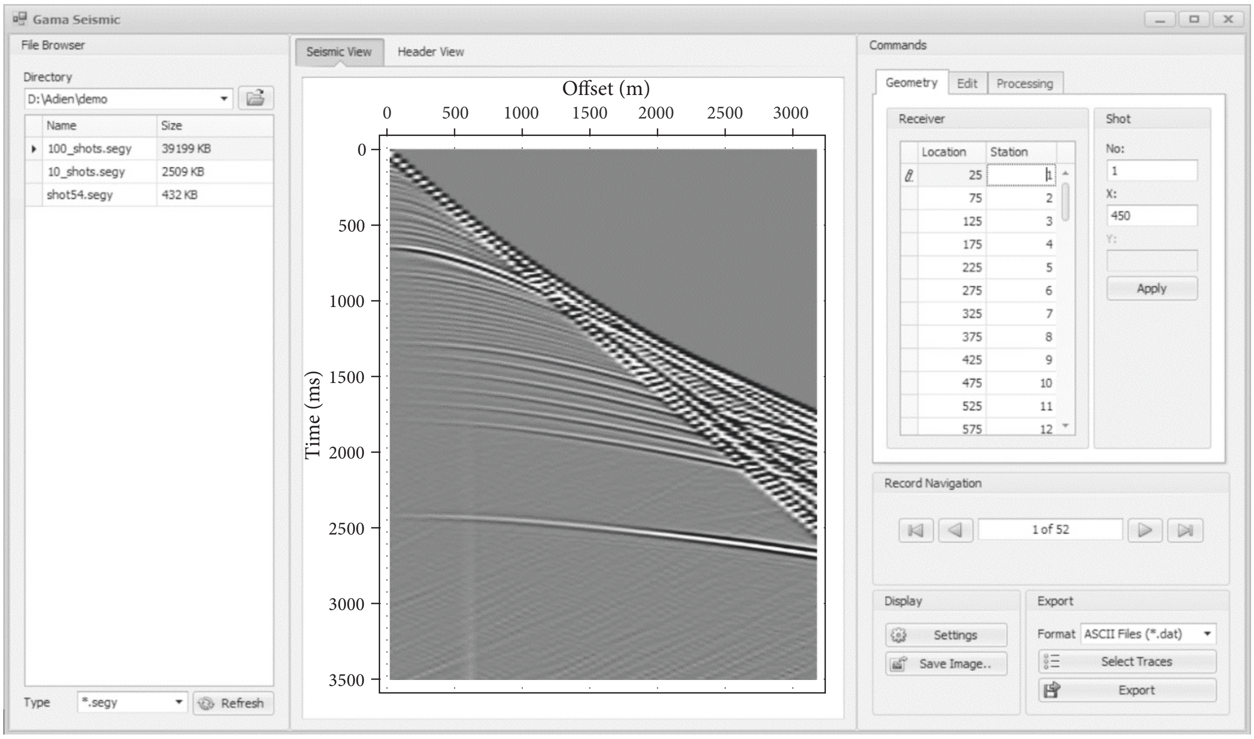
SEISGAMA is written using C# programming language provided by Microsoft Visual Studio 2015. Some scientific computing functions and the display are implemented using libraries from Ilnumerics Computing Engine and Ilnumerics Visualization Engine [22]. SEISGAMA has a modular design that aims to facilitate future development of the application. Some modules that are available in the application include the following: *GamaseisProcessing* and *GamaseisView*. *GamaseisProcessing* is on a SEISGAMA module that contains the reading and writing data processing and some physics and mathematical calculations of the seismic reflection method. *GamaseisView* is a module that controls the graphical user interface, including windows, graphs, seismic sections, labels, and buttons.

GamaseisProcessing consists of scripts to read data in a SEG-Y format, performing amplitude correction, muting, velocity analysis, normal moveout correction, and stacking. The SEG-Y seismic data storage format is the most popular to date, and it is used by almost all commercial seismic software. The reading function of the SEG-Y format is already supported in any form; however, until now, this reading function has been limited to two-dimensional seismic reflection data. Other mathematical functions that have been included in this program are implemented in accordance with existing theory. Calculation of Fourier transforms on SEISGAMA uses libraries that have been prepared and optimized by Ilnumerics [22], which implements a special compilation of C language functions to implement the algorithm called the Fastest Fourier Transform from the West (FFTW) [23]. Utilizing a premade library, which has been tested and uses a lower level programming language, is expected to improve the computation time of the Fourier transform. Thus, the calculation of the transformation to *F-x* and *F-K* domain is mostly done using FFTW algorithm.

GamaseisView consists of a set of views, views that govern the interface between user interaction with the software. Figure 1 shows the main user interface of SEISGAMA where users can perform seismic data processing easily.

4. Results and Discussion

4.1. Amplitude Correction. Amplitude correction was successfully implemented on SEISGAMA using a data-dependent exponential method. Users can change the exponential constants β, but the program provides a default value of $\beta = 16$. Figure 2 shows the results of the amplitude correction using the implemented method. The amplitude of the reflectors, which are associated with $t_0 = 2400$ ms, is



D:\Adien\demo

Name	Size
100_shots.segy	39199 KB
10_shots.segy	2509 KB
shot54.segy	432 KB

Location	Station
25	1
75	2
125	3
175	4
225	5
275	6
325	7
375	8
425	9
475	10
525	11
575	12

No: 1

X: 450

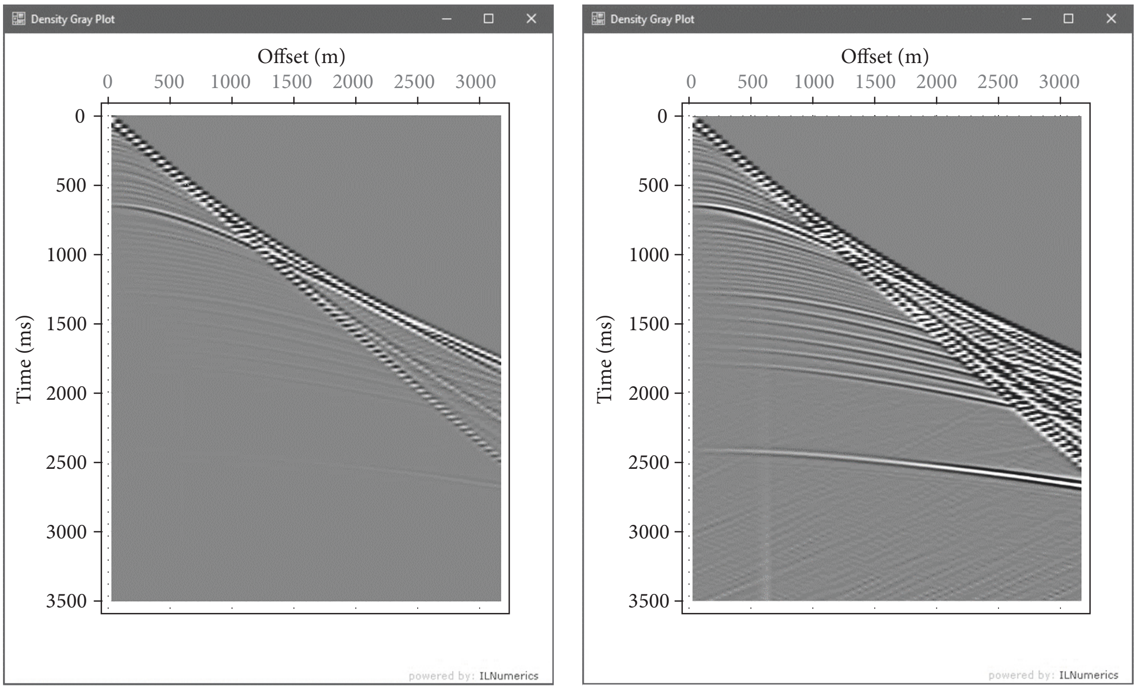
Y:

1 of 52

Format ASCII Files (*.dat)

Type *.segy

Figure 1: SEISGAMA's main graphical user interface.

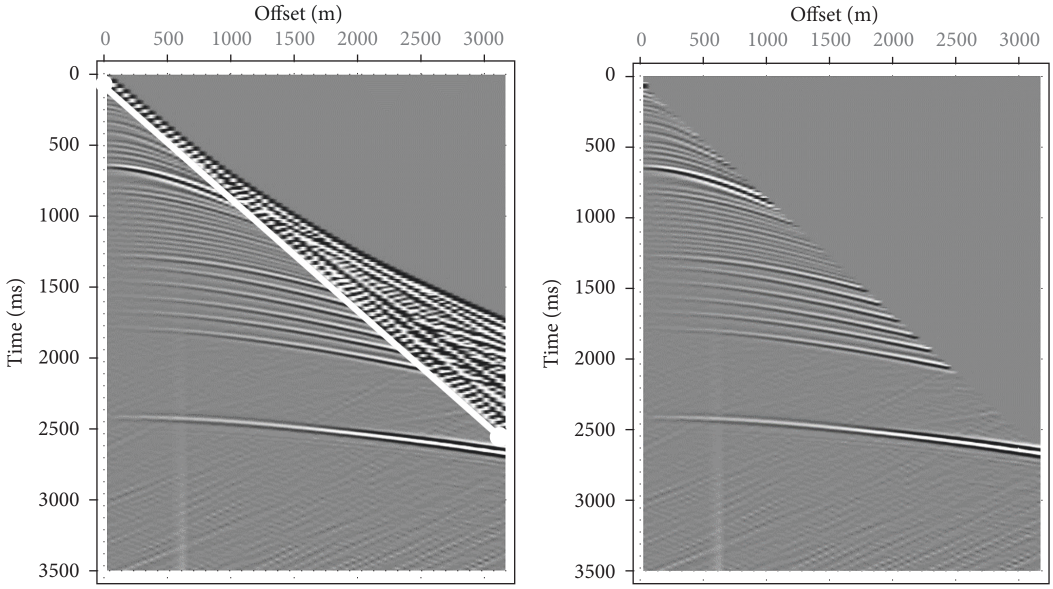


(a)

(b)

Figure 2: Before (a) and after (b) amplitude correction using SEISGAMA.

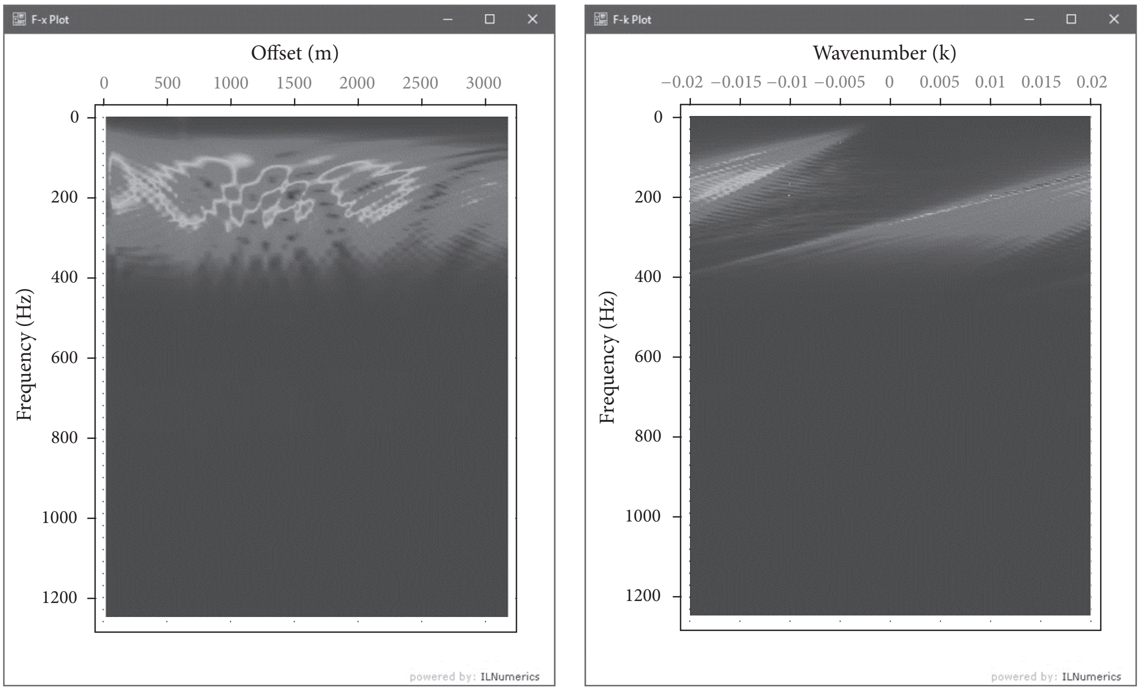
not visible (Figure 2(a)), but it became clear after amplitude correction (Figure 2(b)). Amplitude correction not only amplifies the signal but also increases the amplitude of the noise. We can see in Figure 2(b) that refracted and direct waves became increasingly dominant compared to the reflected record after amplitude correction. Furthermore, upon real seismic recording, noncoherent noise will also be amplified after amplitude correction.



(a)

(b)

Figure 3: Muting process on SEISGAMA guided by white line (a) compared with muted shot-gather (b).



(a)

(b)

Figure 4: Shot-gather in F-x domain (a) and F-K domain (b).

4.2. Muting. In general, the response of the direct and refracted wave is assertive and intrusive to seismic signals. Therefore, the muting process is necessary to remove responses that degrade or disrupt the signal. In SEISGAMA, the muting process is accomplished by picking the appropriate unwanted points in the muting boundary (Figure 3(a)). Then, the user is given a prompt to delete the top of the line which has been picked. The results of the muting process can be seen in Figure 3(b), where waves other than the reflection response have been erased.

4.3. F-X Dan F-K Transformation. The transformation of seismic data from a t-x domain (time and distance) into the F-x and F-K domains can be seen in Figure 4. In the

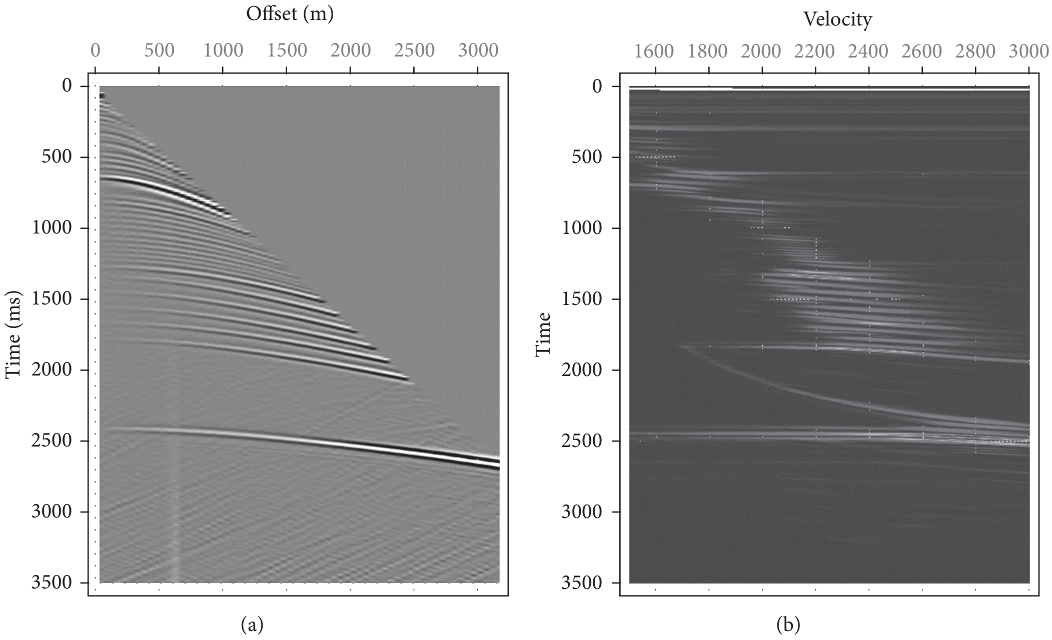


Figure 5: Velocity analysis result (b) from shot-gather on (a).

F-x transformation, the vertical axis has been transformed into frequency units. From this display, we determine all of frequencies included in each seismic trace. In the example, the frequency content included on each seismic trace ranged between 100 and 400 Hz. The maximum of the frequency axis is the Nyquist frequency. In this case, the maximum frequency is 1250 Hz; thus the sampling frequency is 2500 Hz. In the *F-K* transformation, frequency components for each seismic trace are not known directly. Many methods can be performed in the *F-K* domain to determine specific frequency of the trace; one example is to determine the apparent velocity of the incoming signal [24].

4.4. Velocity Analysis. In this section, we describe velocity analysis performed on the set of NMO uncorrected traces or on the imperfect NMO corrections. Through analysis of the velocity, we can estimate the velocity of the medium which causes the parabolic phenomenon upon a reflective seismic response. Users are prompted to enter several parameters that will be tested: that is, the width of the gate time, the minimum source wave speed, and the maximum source wave speed. Figure 5 shows the result of the velocity analysis applied to the figure on the left using the semblance seismic method. In this case, the user enters 25 ms as the time gate width, 1500 m/s as the minimum speed, and up to 3000 m/s as the maximum speed. The results of the speed analysis are picked based on the maximum values of the observed input for the NMO correction (Figure 6).

4.5. Normal Moveout Correction. NMO correction is performed using the previously chosen parameters used in the velocity analysis. Velocity from the chosen result is associated with a linear relationship; therefore, the velocity is continuous each time. Figure 7(b) shows the results of the NMO correction. A correct NMO correction, as shown, is found by flattening seismic reflectors. Figure 8(a) shows that the reflector has become flat at times over 500 ms, while at times below 500 ms the visible convex reflector is still down, which means the velocity is still too high.

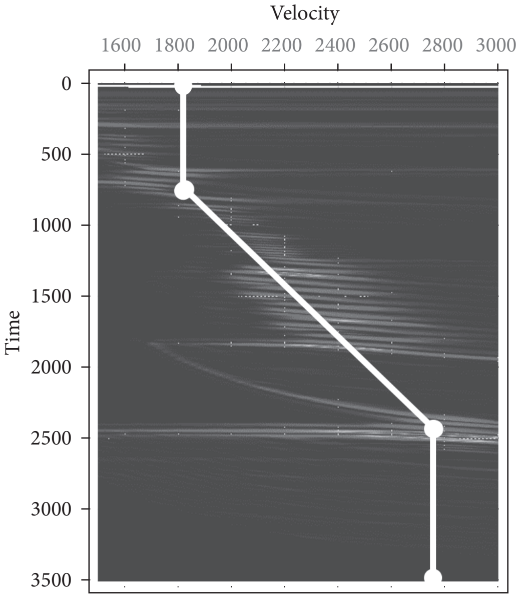


Figure 6: Velocity picking from velocity analysis.

4.6. Stacking. In principle, the stacking process sums up all the seismic traces to improve the signal to noise ratio (high S/N). The stacking process can be carried out for any group of seismic traces, but in the refraction seismic method, this

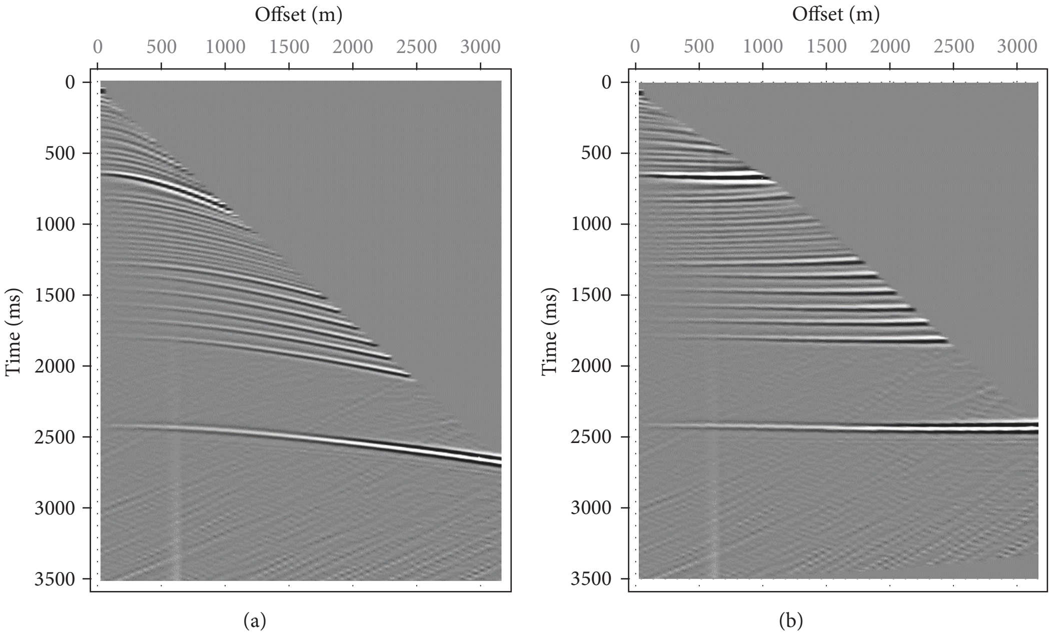


Figure 7: Uncorrected seismic traces (a) and NMO corrected seismic traces (b).

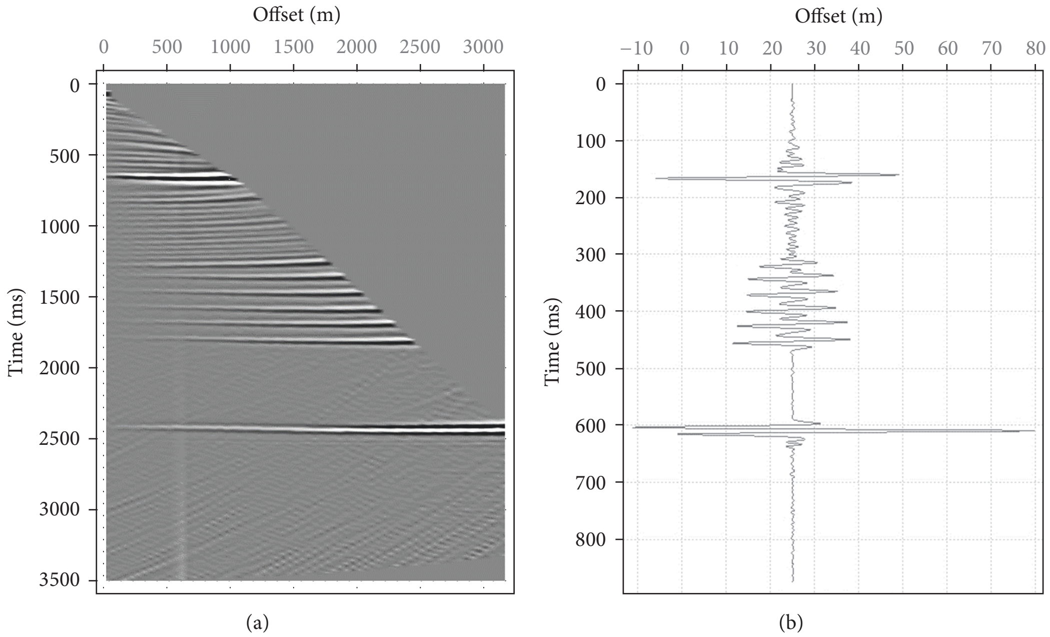


Figure 8: NMO corrected seismic traces (a) and stacked seismic trace (b).

process must be done after the NMO correction, and it should be ensured that the reflector is already flat; therefore the stacking process really can improve the signal over noise. Figure 8(b) showed the stacking result from SEISGAMA software.

5. Conclusions

SEISGAMA is a free seismic reflection data processing software program that has successfully processed 2D seismic data for amplitude correction, muting, *F*-*x* and *F*-*K* domain transform, velocity analysis, NMO correction, and trace stacking. SEISGAMA was built using C# programming language which is natively supported by a Windows operating system. A minimum .NET 4.5 framework is required to run this software; therefore, a minimum Windows 7 Service Pack 1 is implicitly required. SEISGAMA can give an alternative software choice to users interested in geophysics that has the added benefit of Windows's compatibility and an intuitive user interface. The basic procedure for the processing of seismic data of reflection has been implemented into SEISGAMA. For advanced seismic processing option, including a Prestack Depth Migration and Full-Waveform inversion, this is currently still under construction and will soon be provided in SEISGAMA. The software is freely available upon request by contacting the authors.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The authors would like to thank the Geophysics Laboratory, Faculty of Mathematics and Natural Sciences, at the Universitas Gadjah Mada, for their support and facilities to develop SEISGAMA. This research was funded by the Ministry of Research, Technology, and Higher Education of Indonesia, under Contract no. 909/UN1-PIII/LT/DIT-LIT/2016.

References

[1] W. M. Telford, L. P. Geldart, and R. E. Sheriff, *Applied Geophysics*, Cambridge University Press, Cambridge, 1990.

[2] M. A. Biot, *Acoustics, Elasticity, and Thermodynamics of Porous Media*, I. Tolstoy, Ed., Acoustical Society of America, Woodbury, New York, USA, 21th edition, 1992.

[3] Lowrie., *Fundamentals of Geophysics*, Cambridge University Press, 1997.

[4] G. V. Chilingarian and A. E. Gurevich, “The petroleum system — From source to trap,” *Journal of Petroleum Science and Engineering*, vol. 14, no. 3-4, pp. 258–260, 1996.

[5] D. W. Steeples and R. D. Miller, “Seismic Reflection Methods Applied to Engineering, Environmental, and Ground-Water Problems,” *Geotechnical and Environmental Geophysics*, vol. 1, p. 33, 1997.

[6] J. W. Stockwell Jr., “The CWP/SU: Seismic Unix package,” *Computers & Geosciences*, vol. 25, no. 4, pp. 415–419, 1999.

[7] M. E. Templeton and C. A. Gough, “Web seismic Un(*)x: Making seismic reflection processing more accessible,” *Computers & Geosciences*, vol. 25, no. 4, pp. 421–430, 1999.

[8] K. S. Beaty, G. Perron, I. Kay, and E. Adam, “DSISoft - A MATLAB VSP data processing package,” *Computers & Geosciences*, vol. 28, no. 4, pp. 501–511, 2002.

[9] T. Andrews, “Computation Time Comparison Between Matlab and C++ Using Launch Windows,” *Aerospace Engineering*, pp. 1–6, 2012.

[10] W. A. Mousa and A. A. Al-Shuhail, “Processing of seismic reflection data using MATLAB™,” *Synthesis Lectures on Signal Processing*, vol. 10, pp. 1–97, 2011.

[11] L. Gasperini and G. Stanghellini, “SeisPrho: An interactive computer program for processing and interpretation of high-resolution seismic reflection profiles,” *Computers & Geosciences*, vol. 35, no. 7, pp. 1497–1507, 2009.

[12] M. Beyreuther, R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann, “ObsPy: A python toolbox for seismology,” *Seismological Research Letters*, vol. 81, no. 3, pp. 530–533, 2010.

[13] L. Krischer, A. Fichtner, S. Zukauskaite, and H. Igel, “Large-scale seismic inversion framework,” *Seismological Research Letters*, vol. 86, no. 4, pp. 1198–1207, 2015.

[14] W. Suryanto and T. M. Irnaka, “Web-based application for inverting one-dimensional magnetotelluric data using Python,” *Computers & Geosciences*, vol. 96, pp. 77–86, 2016.

[15] K. Schwehr, “Seismic-py: Reading Seismic Data with Python,” *Center for Coastal and Ocean Mapping*, p. 472, 2008.

[16] N. B. Rizvandi, A. J. Boloori, N. Kamyabpour, and A. Y.zomaya, “MapReduce implementation of prestack kirchhoff time migration (PKTM) on seismic data,” in *Proceedings of the 2011 12th International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2011*, pp. 86–91, IEEE, Gwangju, South Korea, October 2011.

[17] Netmarketshare., “Desktop Operating System Market Share, 2016,” https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0.

[18] D. S. Platt, *Introducing Microsoft. NET*, Microsoft Press, Redmond, Washington, USA, 3rd edition, 2003.

[19] Microsoft., “Microsoft. NET Guidelines, 2012,” Microsoft Legal and Corporate Affairs, https://msdn.microsoft.com/en-us/library/ms184412(v=vs.100).aspx.

[20] M. Campillo, M. Bouchon, and B. Massinon, “Theoretical study of the excitation, spectral characteristics, and geometrical attenuation of regional seismic phases,” *Bulletin of the Seismological Society of America*, vol. 74, no. 1, pp. 79–90, 1984.

[21] X. Xu, I. Tsvankin, and A. Pech, “Geometrical spreading of P-waves in horizontally layered, azimuthally anisotropic media,” *Geophysics*, vol. 70, no. 5, pp. D43–D53, 2005.

[22] “Ilnumerics. GmbH, 2016, ILNumerics – Technical Computing,” https://ilnumerics.net.

[23] M. Frigo and S. G. Johnson, “FFTW: an adaptive software architecture for the FFT,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 1381–1384, May 1998.

[24] F. Glangeaud, J. L. Mari, and F. Coppens, “Signal processing for geologists and geophysicists,” *Editions Technip*, 1999.