

Research Article

Oleg Zaikin*, Pavel Petrov, Mikhail Posypkin, Vadim Bulavintsev, and Ilya Kurochkin

A Volunteer Computing Project for Solving Geoacoustic Inversion Problems

<https://doi.org/10.1515/eng-2017-0040>

Received Oct 02, 2017; accepted Nov 15, 2017

Abstract: A volunteer computing project aimed at solving computationally hard inverse problems in underwater acoustics is described. This project was used to study the possibilities of the sound speed profile reconstruction in a shallow-water waveguide using a dispersion-based geoacoustic inversion scheme. The computational capabilities provided by the project allowed us to investigate the accuracy of the inversion for different mesh sizes of the sound speed profile discretization grid. This problem suits well for volunteer computing because it can be easily decomposed into independent simpler subproblems.

Keywords: volunteer computing, BOINC, geoacoustic inversion, underwater acoustics, global optimization, GPU

1 Introduction

Desktop Grid computing [11] uses resources of geographically distributed personal computers, connected over general-purpose network. Huge increase in personal computers' performance has made Desktop Grid computing a powerful tool for solving numerous scientific problems. Volunteer computing [16] is a special kind of desktop grid computing. Its defining characteristic is that it uses computational resources of private persons. Usually, one vol-

unteer computing project is designed to solve one or several closely related hard problems. Volunteer computing is very cheap — to maintain a project one needs only a dedicated server (connected to Internet) and several client applications. Volunteer computing suits well to solve hard problems, which can be decomposed into a set of independent subproblems. In last two decades, a lot of important results in various areas (astronomy, medicine, mathematics, etc.) were obtained in volunteer computing projects.

There exists a number of classes of computationally intense geoacoustic inversion problems, which can be solved using embarrassing parallelism. This is the main reason why we launched the volunteer computing project *Acoustics@home*, aimed at solving such problems. The project is based on the popular platform BOINC (Berkley Open Infrastructure for Network Computing [3]). The notion of geoacoustic inversion refers to a variety of techniques in underwater acoustics that can be used for the reconstruction of water column and bottom parameters from acoustic data [20]. While previously the data for the geoacoustic inversion was mostly obtained using expensive receiver arrays, recently it was shown that a single-hydrophone recording of a broadband pulse signal can be also successfully used for estimating the acoustical parameters of sea bottom [8–10]. Due to the lack of the spatial diversity in single-hydrophone measurements, the inversion procedure in this case relies on the so-called waveguide dispersion, i.e., the dependence of arrival times on frequency and mode number (see details, e.g., in [9, 23]). Some interesting recent results [25] also indicate that single-hydrophone dispersion-based inversion method outlined in [9] can be used for the estimation of a sound-speed profile in a shallow-water waveguide. The implementation of this method in practice can be thought of as a solution of a minimization problem in a (very large) discrete search space [26], and every evaluation of the cost function requires numerous solutions of an acoustic spectral problem [23, 26]. Thus, the whole computational burden can be easily divided into a large number of relatively simple independent tasks, offering many opportunities for the application of parallel computing.

The study has the following structure. In Section 2 we outline the dispersion-based geoacoustic inversion

***Corresponding Author: Oleg Zaikin:** Matrosov Institute for System Dynamics and Control Theory SB RAS, Irkutsk, Russia, E-mail: zaikin.icc@gmail.com

Pavel Petrov: V.I. Il'ichev Pacific Oceanological Institute FEB RAS, Far Eastern Federal University, Vladivostok, Russia, E-mail: petrov@poi.dvo.ru

Mikhail Posypkin: Dorodnicyn Computing Centre FRC CS RAS, Moscow, Russia, E-mail: mposypkin@gmail.com

Vadim Bulavintsev: Matrosov Institute for System Dynamics and Control Theory SB RAS, Irkutsk, Russia, E-mail: v.g.bulavintsev@gmail.com

Ilya Kurochkin: A.A. Kharkevich Institute for Information Transmission Problems RAS, Moscow, Russia, E-mail: kurochkin@gmail.com

scheme following [8, 9, 23, 26]. The synthetic acoustic data used in our work is discussed in Section 3. In Section 4 we describe the project *Acoustics@home* and show the results of computational experiment, performed within its framework. In Section 5 it is discussed in which way we plan to improve *Acoustics@home*. In the rest of the paper we discuss related work and draw conclusions.

2 Dispersion-based geoacoustic inversion

The dispersion-based technique for the geoacoustic waveguide parameters estimation [9] introduced by J. Bonnel and his colleagues several years ago is currently a very dynamic field of research in underwater acoustics [8, 10, 23, 25]. In this technique, the dispersion data is used as an input of the inversion algorithm. By the dispersion data we understand the set of functions τ_m , $m = 1, 2, \dots, M$, where $\tau_m(f)$ denotes the arrival time of m -th modal component [19, 23] of a pulse acoustical signal as a function of frequency f (M is the total number of modes that can be filtered from the data). The curves $t = \tau_m(f)$ in the two-dimensional time-frequency space are called *dispersion curves* [20].

The experimental dispersion curves can be obtained from a pulse signal recorded by a single receiver (hydrophone) [8, 10]. Usually, various approaches to the mode filtering are used to extract the dispersion curves from a time series obtained in an experiment. The rise of the dispersion-based inversion schemes popularity can be associated with the so-called warping transform [5, 8] offering a very elegant way to separate the modal components of a pulse signal. At the same time, for any fixed model of a waveguide dispersion curves can be computed theoretically by solving acoustic spectral problem [19, 26]. The mismatch between the experimental and theoretical arrival times indicates to what extent the given model of the waveguide is consistent with the observation results. Normally the waveguide model includes certain number of unknown parameters, and the geoacoustic inversion serves the purpose of estimating their values. The set of parameters corresponding to the minimal mismatch determines the theoretical model that is the most adequate to the experimental data. Hereafter this set is referred to as *media parameters estimate* based on the given experimental data. Note that probably the most important advantage of the dispersion-based inversion schemes is their ability to provide some information on the medium using single-hydrophone measurements. By contrast, more con-

ventional geoacoustic inversion methods [20] usually require the deployment of a vertical or horizontal array of receivers (which makes the experiment much more expensive and complicated). The lack of spatial diversity of the measurements in this case is compensated for by the frequency diversity.

The geoacoustic inversion problem can be transformed into a problem of minimization of a certain mismatch function [26]. The simplest natural choice for such function is the standard mean square fitness function measuring the average squared discrepancy between the theoretical arrivals $\tau_m^{\text{th}}(f, \mathbf{A})$ computed for the parameters vector $\mathbf{A}$ and experimental arrivals $\tau_m^{\text{exp}}(f)$:

$$E(\mathbf{A}) = \frac{\sum_{m=1}^M \sum_{n=1}^{N_m} \left| \tau_m^{\text{th}}(f_{nm}, \mathbf{A}) - \tau_m^{\text{exp}}(f_{nm}) \right|^2}{\sum_{m=1}^M N_m}. \quad (1)$$

The minimization is performed over a certain domain of parameters values which is typically a cuboid in a N_p -dimensional Euclidean space, where N_p is the number of the waveguide parameters being inverted. The cuboid boundaries are determined from certain physical considerations.

3 Waveguide model and the synthetic data

In the present study, the geoacoustic inversion problem is considered in the case of a homogeneous two-dimensional waveguide $\Omega = \{(x, z) | 0 \leq z \leq H\}$, where z denotes depth, and x is horizontal coordinate. The waveguide consists of the water column $0 \leq z \leq h$ and a single bottom layer $h \leq z \leq H$. The sound speed profile in the water is continuous function $c = c(z)$ (the dependence of sound speed on z).

For simplicity the sound speed profile is assumed to be independent of x . Although in real ocean this condition is never fulfilled exactly, the variation of sound speed in range x is usually much slower than in depth z . It is often reasonable to assume that the sound speed profile is the same for all x in a certain area of a shallow sea containing the source and the receiver. In our simplified model the bottom is represented by a fluid halfspace with the sound speed c_b and density ρ_b . Although this may also look somewhat non-realistic, the situation where the bottom depth h varies by few meters over a track of several kilometres is quite common in underwater acoustics. In such cases it is reasonable to assume the bottom depth to be constant. A more complicated inversion algorithm

where the variations of the latter are taken into account is described in [23].

Our goal in this study is to evaluate the possibility of the sound speed profile $c(z)$ reconstruction from the dispersion data. The results from [25] show that certain approximation of the function $c(z)$ can be obtained by the dispersion-based inversion (although in [25] it is not really perfect even for synthetic data). In the present work we show that at least in principle it can be inverted for very accurately (when inverting for the sound speed profile alone and considering all other parameters known).

For our numerical experiments we prepared a synthetic data by solving the direct propagation problem as described in [24]. A chirp signal with instantaneous frequency varying from $f_0 = 20$ Hz to $f_1 = 300$ Hz was emitted by the source at $z_s = 10$ m. The sound speed profile in the water used in simulations is shown in Figure 1, while the bottom parameters were chosen as follows: $c_b = 1700$ m/s, $\rho_b = 1.7$ g/cm³. The time series of the signal was computed at $R = 7$ km from the source.

We applied the warping-based method of the extraction of the dispersion curves from the simulated data. As a result, the dispersion curves $\tau_m(f)$ of the five modes were obtained (i.e., in our case $M = 5$).

4 Computational experiments in Acoustics@home

In order to solve computationally hard geoacoustic inversion problems, we launched the volunteer computing project Acoustics@home [2] on 28 March 2017. It is based on the BOINC platform [3]. In Acoustics@home all daemons (that operate on the server) and client application (it operates on volunteers' PCs) are based on the CMBALA MPI-program [26]. Work generator daemon decomposes an original problem into independent subproblems by varying several parameters of a search space. The rest of the parameters are varied by the client application. For each set of the parameters $\mathbf{A}$ the value of the mismatch function (1) is calculated by the client application. Program sources of the project are available online [1].

As it was mentioned above, the goal of our computational experiments is the estimation of the sound-speed profile (SSP) $c = c(z)$ in the water column (see Section 3). Hence, the set of the inversion parameters $\mathbf{A}$ consists of the sound speed values $c_1, c_2, \dots, c_{N_c}$ on a uniform grid of the depth values $z_0, z_1, z_2, \dots, z_{N_c}$. The grid is introduced in the water column in such a way that $z_0 = 0$, $z_{N_c} = h$ ($z_{i+1} - z_i = \Delta z = \text{const}$). We also assume that $c_{N_c} = c_{N_c-1}$,

and therefore N_c parameters should be inverted. For each z_j corresponding c_j in our experiment can take the values in a discrete set of even integers from 1450 m/s to 1510 m/s (31 possible values in total). Although it means that the estimation error can be ~ 2 m/s just due to the search space discretization, such accuracy is sufficient for many applications in underwater acoustics.

Clearly, the resolution of the SSP $c(z)$ depends on the number of nodes N_c . The more nodes we can afford, the better is the possible accuracy of the SSP estimation. We launched our dispersion-based geoacoustic inversion algorithm for $N_c = 1, 2, \dots, 6$. The inversion procedure for $N_c = 1, 2, \dots, 4$ was successfully completed via CMBALA on the computational cluster "Academician V.M. Matrosov" of Irkutsk Supercomputer Center of SB RAS [21]. In our experiments we used 1 node of the cluster, which is equipped with two 18-core CPUs Intel Xeon E5-2695 (36 CPU cores in total) and 128 gigabytes of RAM.

The search spaces for $N_c = 5$ and $N_c = 6$ turned out to be too large for a cluster. That is why we launched them in Acoustics@home on 12 September 2017. For $N_c = 5$ the search space was divided into 29 791 workunits, which were made by varying the values of c_1, c_2, c_3 . The client application had to vary the values of c_4, c_5 for each workunit, so 961 points had to be processed within it. For $N_c = 6$ the search space was divided into 923 521 workunits (by varying the values of c_1, c_2, c_3, c_4), each of them consisted of 961 points (the values of c_5, c_6 were varied). On average it took about 1 hour to process one workunit on 1 CPU core. Both experiments were successfully completed, it took about 2 months to perform the hardest of them.

Recently Acoustics@home has reached the average performance of 7 teraflops, 2544 PCs of 1547 volunteers have already participated in the project. Let us remind, that during our experiments for $N_c = 1, 2, \dots, 4$ we used 36 cluster's CPU cores. In the considered experiments for $N_c = 5$ and $N_c = 6$, the project's performance was comparable to that of a computational cluster equipped with 500 CPU cores of the mentioned kind. Details of the conducted experiments are presented in Table 1. The estimated sound speed profiles are presented in Figure 1.

Let us briefly discuss the inversion results from the acoustical point of view. It should have been expected that in the case $N_c = 1$ we obtained some value of the sound speed that is close to its average over the whole water column. In the case $N_c = 2$ the approximation is still very inaccurate, yet it already catches the main features of the true profile. Indeed, to certain extent it reproduces the thermocline near the surface and the homogeneous layer near the bottom.

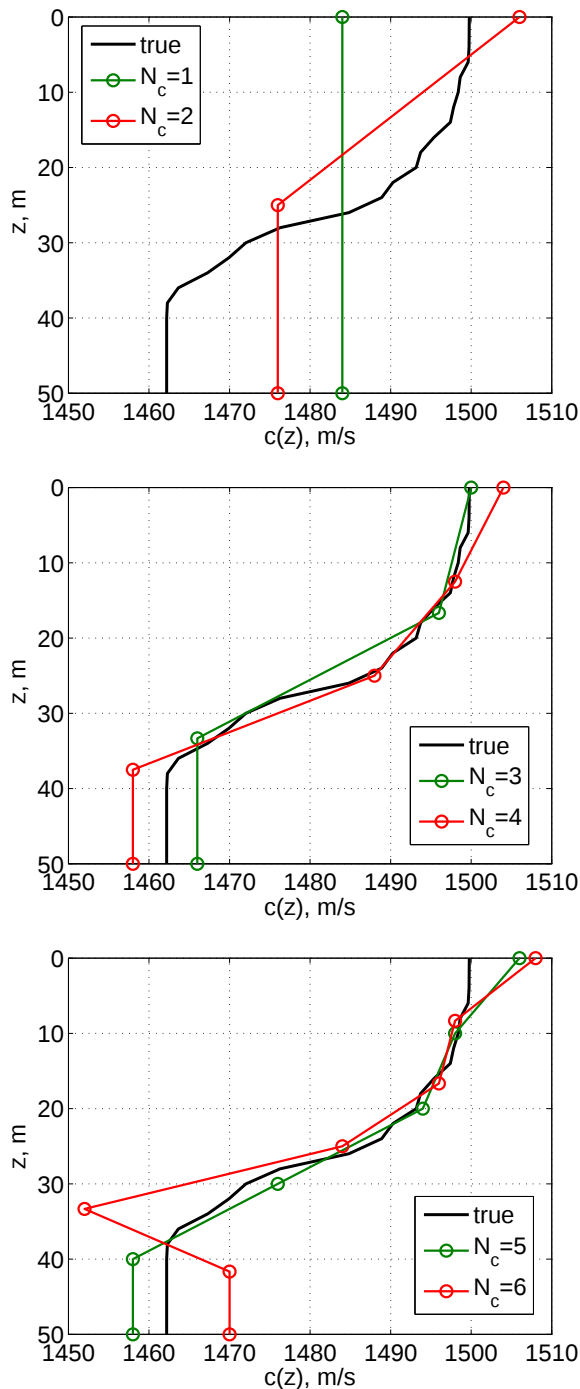


Figure 1: Estimated sound speed profiles $c(z)$ for $N_c = 1, 2, \dots, 5$ and the true one (black line on all panels). **Top panel:** $N_c = 1$ (green line) and $N_c = 2$ (red line). **Middle panel:** $N_c = 3$ (green line) and $N_c = 4$ (red line). **Bottom panel:** $N_c = 5$ (green line) and $N_c = 6$ (red line).

Table 1: Results of geoacoustic inversion for different values of N_c .

N_c	Residue	Points	Time	Platform
1	0.0127314	31	10 seconds	cluster
2	0.00924369	961	2 minutes	cluster
3	0.00500185	29 791	45 minutes	cluster
4	0.00416296	923 521	23 hours	cluster
5	0.00428371	28 629 151	10 days	BOINC
6	0.00384831	887 503 681	58 days	BOINC

From the middle and the bottom panels of Figure 1 it is clear that starting from $N_c = 3$ we already obtain a relatively accurate estimate of the sound speed profile. On average the approximations corresponding to $N_c = 4$ and $N_c = 5$ are almost identically accurate. Thus, the grid of a few points is in fact sufficient to estimate the hydrological properties of the shallow-water waveguide via geoacoustic inversion. The results for $N_c = 6$ also indicate that the increase of the number of nodes does not necessarily improves the inversion accuracy. This can be considered an additional justification for using trans-D inversion technique [25], where the preference is given to the sound speed profile approximations with fewer nodes.

One of the most efficient ways to boost the performance of a BOINC-based project is to organize a competition on the BOINCstats site [7]. In the course of such competitions teams try to earn as many project's credits as possible. In September 2017 the "Ocean of fantasy" competition was launched to help Acoustics@home perform the aforementioned experiments faster. As a result, 26% of workunits for $N_c = 6$ were processed within a week (let us remind, that the whole experiment took 58 days). At the end the competition, the project's has reached the record performance of 18 teraflops. It should be noted, that competitions of the mentioned type can lead to high load of a project's server. In order to minimize the server load, an improved task scheduling can be used (e.g., [22]). However, in our case the default task scheduling coped with the competition.

5 Preliminary experiments on GPU

In the future, we are planning to improve Acoustics@home in two respects. Firstly, it is intended to employ advanced black-box optimization algorithms. Secondly, a GPU-based version of the computing application will be implemented. In the following subsection, a new black-box optimization algorithm is proposed, then we describe

its GPU implementation, and present the experimental results obtained by this application on one geoacoustic inversion problem.

5.1 New Black-box Optimization Algorithm

The problem (1) belongs to the class of black-box optimization problems which are very common in practice. In such problems derivatives are either unavailable or very hard to compute. Thus, solution methods should not rely on derivatives. In this paper, in order to minimize the objective function (1) we use uniform mesh to obtain a finite search space. This search space is processed by exhaustive search (see Section 4). In [27] instead of exhaustive search we applied iterative hill climbing algorithm. In the future, we are planning to apply the combination of global and local search techniques. The global search techniques will be used to diversify the search, by starting the local search from several initial points. The local search method will be used to minimize the value of the objective function. We plan to consider several local search techniques: Hooke-Jeeves method [17], pseudo-gradient approach [15] and a variety of coordinate descent techniques.

In black-box optimization problems derivatives are either unavailable or very hard to compute. Thus, solution methods should not rely on derivatives. There are a lot of local search techniques which can be used for this purpose: Hooke-Jeeves method [17], pseudo-gradient approaches [15] and a variety of coordinate descent techniques. We developed a simple modification of the coordinate descent method presented in Algorithm 1

The proposed algorithm (ASN search) uses asymmetric neighborhood adaptation. It tries to improve the objective function $f(\mathbf{x})$ by shifting by some value δ_i along each coordinate vector $\mathbf{e}_i$ in both directions. In the case of success (failure) the respective step size is increased (decreased) by multiplying it on $\alpha > 1$ ($\beta < 1$). When approaching minimum values of δ_i usually decrease. The method terminates when the maximal component of the vector δ becomes less than the predefined $\varepsilon > 0$.

The choice of initial δ and ε significantly affects the performance. For the problem under consideration the best performance were obtained with $\delta_i = 10^{-1}$, $i = 1 \dots, n$, $\varepsilon = 10^{-4}$.

```

1 while  $\max_{i=1,\dots,n} \delta_i > \varepsilon$  do
2   for  $i = 1$  to  $n$  do
3      $\mathbf{x}' = \mathbf{x} + \delta_i \mathbf{e}_i$ 
4     if  $f(\mathbf{x}') < f(\mathbf{x})$  then
5        $\mathbf{x} := \mathbf{x}'$ 
6        $\delta_i := \alpha \delta_i$ 
7     else
8        $\mathbf{x}' = \mathbf{x} - \delta_i \mathbf{e}_i$ 
9       if  $f(\mathbf{x}') < f(\mathbf{x})$  then
10         $\mathbf{x} := \mathbf{x}'$ 
11         $\delta_i := \alpha \delta_i$ 
12      else
13         $\delta_i := \beta \delta_i$ 
14      end
15    end
16  end
17 end

```

Algorithm 1: ASN Search

5.2 A GPU implementation of the Black-box algorithm

For many classes of computational problems, graphics processing units (GPUs) show significant speedup over central processing units (CPUs). In particular, GPUs can help to solve global optimization problems (e.g., see [4]). However, to actually obtain this speed-up, the implementation of an algorithm should be thoroughly adapted for GPU architecture. Fortunately, in our approach to geoacoustic inversion problem the base algorithms are well fit for a GPU. The complete process of solving the geoacoustic inversion problem could be represented as a hierarchy of the subproblems solved by the corresponding subalgorithms (see Table 2).

The GPU programming model is based on data parallelism. To achieve peak efficiency, modern GPUs should simultaneously run about 10 000 computational threads [12]. In the GPU-based implementation of our search procedure, we exploit data parallelism found in the parallel calculation of modal group velocities for a single point. Level 1 sub-algorithm executes on the CPU, while level 2-4 sub-algorithms execute on the GPU (Table 2). A typical residue calculation for a single search space point requires calculation of modal group velocities for thousands of frequencies. Thus, data parallelism exposed in this way should be enough to efficiently exploit modern GPUs.

Traditionally, geoacoustics computations are performed in double precision floating point arithmetics (FP64). However, modern consumer-grade GPUs suffer a great (for some devices, up to 32 times [12]) drop in per-

Table 2: Hierarchy of algorithms used in the process of solving the geoacoustic inversion problem.

Level	Problem	Algorithm
1	Search for lowest residue point	ASN Search
2	Calc. residue for a point	Calc. residue over individual frequencies
3	Calc. modal group velocities for a single frequency	Numerical differentiation
4	Sturm-Liouville problem on a mesh	Calc. eigenvalues of a tridiagonal symmetric matrix

formance when using FP64 instructions instead of their single-precision (FP32) counterparts. This fact led us to investigate the consequences of changing the algorithm to FP32.

In our computational experiments the ASN search (see Subsection 5.1) was applied for minimizing the objective function (1). We considered the model problem of geoacoustic inversion with 3 parameters (R, ρ_b, c_b). The sound speed profile in the water column and the pulse signal emitted by the source was the same as in [27]. The true values and the intervals of the unknown parameters are presented in Table 3.

Table 3: Intervals and true values of the unknown parameters.

Parameter	True value	Min value	Max value
c_b , m/s	1700	1550	1850
ρ_b , g/cm ³	1.7	1.1	2
R , m	7000	6850	7150

The results are presented in Table 4. All experiments were conducted on the Intel Core i7 930 CPU and the Nvidia GTX 1050 GPU, with the use of CUDA 8.0 SDK and GCC 5.4.0 compiler. The NVCC compiler was forced to produce native Compute Capability 6.1 code. Both NVCC and GCC compilers were run with -O3 optimization flag.

Table 4: Algorithm performance for various platforms and floating-point precisions formats.

Platform	Precis.	Residue	Time, s.	points/s.
CPU, AlgLib	FP64	0.0128206	4678	0.1710
CPU	FP64	0.0132451	1212	0.5272
	FP32	0.0209403	487	1.0616
GPU	FP64	0.0131659	109	6.5779
	FP32	0.0215154	46	15.2391

The most computationally intensive part of our algorithm is the bisection sub-algorithm at level no. 4 in Table 2. We implemented it as described in [13] and [12], producing single- and double-precision versions, algorithmi-

cally identical on both GPU and CPU platforms. For reference, we provide the double-precision version based on the AlgLib library [6]. This library includes a state-of-the-art implementation of the bisection algorithm, thoroughly tuned to produce the most accurate results possible with the modern CPUs floating point units (FPUs). Our CPU-based implementation of bisection algorithm can not boast such accuracy. However, it is notably faster than AlgLib due to its simplicity. That is why, our and AlgLib-based algorithm's results are different. The discrepancy between the outputs (residue) of the same algorithm on the CPU and GPU is the result of the different implementations of floating-point units on these platforms.

Table 4 clearly indicates the superiority of the GPU performance, even in FP64. The FP32 mode of the GPU is 150 times faster than the FP64 mode, though the former's final residue (0.0215154) is much worse than the latter one's (0.0131284). However, recomputation of this final point in FP64 gives the result (0.0134315, see Table 5) that is on par with that of the FP64 CPU implementation.

This leads us to suggest the mixed-mode, two-stage algorithm, that uses the "coarse" FP32 computation mode during Stage I, and then refines its result in Stage II by resuming the search in the "fine" FP64 computation mode. The performance of this algorithm is presented in Table 5. For reference, the "Initial residue" column of Table 5 shows residue of the best point found by Stage I recomputed by various Stage II algorithms. It is easy to see that the GPU-based FP32 algorithm for Stage I is best complemented for Stage II by the GPU-based FP64 algorithm. The total runtime of this mixed-mode algorithm is 127 seconds. This is faster than runtime of the pure GPU-based FP64 algorithm (777 seconds), and its final residue (0.0130635) is almost equal to that of the pure CPU-based FP64 (alglib [6]) algorithm (0.0130471).

The FP64 performance of our GPU-based algorithm is only 2.5 times lower than the performance of our FP32 algorithm on the same GPU (see Table 4). However, according to [12] this performance drop should be at least 4x times for the GTX1050 GPU used in our experiments. This means that the GPU performance is likely to be bottlenecked by suboptimal multiprocessor occupancy [12]

Table 5: Mixed-mode algorithm performance.

Stage I			Stage II				Total time, s.
Platform	Final residue	Time, s.	Platform	Initial residue	Final residue	Time, s.	
GPU (FP32)	0.0215154	46	CPU (alglib)	0.0137885	0.0130473	1446	1492
			CPU	0.0133772	0.0131325	1099	1145
			GPU	0.0134315	0.0130635	81	127

and/or excessive memory transfers, rather than the FP64 FPU's of the device. The higher GPU performance we perceived in our previous experiments with the parallelization scheme based on the simultaneous computing of residues for many search space points supports this theory. This means that a thorough optimization of the GPU code could increase the GPU performance of our geoacoustic inversion algorithm at least 2-3 times.

6 Related work

The computing clusters are often used in the practical applications of geoacoustic inversion algorithms (see, e.g., [14] and references therein). While sometimes one can compromise by using heuristic optimization techniques in end-user applications, the problems of development and validation of inversion algorithms anyway set very strong demand for the high-performance computational tools.

In the previous works [26, 27] we solved on a computing cluster the problems similar to that, considered in the present paper. The restrictions on available computational resources forced us to launch *Acoustics@home*. But we will continue to use the cluster in order to solve simple scenarios, or to test new versions of the computing application.

In [18] general purpose GPUs were used to accelerate underwater acoustic propagation modeling.

7 Conclusions

In the present study we describe the volunteer computing project *Acoustics@home* that was set up to handle computationally intense geoacoustic inversion problems. With the help of our volunteers we conducted a series of experiments aiming to investigate the capabilities of the dispersion-based inversion method in producing the estimate of the sound speed profile in shallow-water environment. In our view, it seems interesting to find out, how many grid points are required to resolve the function $c(z)$

and how accurate the inversion result can be. In [25] a similar question was studied by using a sophisticated trans-D Bayesian inversion approach. Our results are obtained in a much more simple and straightforward way, and they are also probably easier to understand. The accuracy of the sound speed profile inversion is quite encouraging, and it seems that in principle function $c(z)$ can be estimated from the data almost exactly. At least, it is clear that there is no limitation neither on the side of the warping-based mode filtering algorithm, nor on the side of the global minimization problem (i.e., it features only one global extremum of the fitness function near the true set of parameters values). It is also important to note that increasing the number of nodes in sound speed profile approximation does not necessarily improve the inversion accuracy. At some point adding new dimensions of the search space results in the larger deviation of the inverted sound speed profile from the true one.

The present study highlights great opportunities that volunteer computing can bring into the field of computational underwater acoustics. In the future, we are aiming at developing a powerful computational framework that can be used on demand by any member of ocean acoustics research community who needs to conduct some very intense calculations in order to solve certain direct or inverse problems. To this end, we are planning to expand our code by adding some modules capable of solving propagation problems in inhomogeneous 2D and 3D waveguides and also implement other inversion algorithm (not necessarily relying on single-hydrophone data).

Acknowledgement: The authors are grateful to Dr Julien Bonnel (ENSTA Bretagne, France) and to Prof Stanley Dosso (University of Victoria, Canada) for sharing their code and more importantly their experience in the geoacoustic inversion.

This study was partially supported by the Council for Grants of the President of the Russian Federation (grants No. MK-2262.2017.5, No. NSh-8081.2016.9, No. NSh-8860.2016.1), the Russian Foundation for Basic research (grants No. 16-05-01074_a, No. 16-07-00155_a, No. 15-29-07095_ofi-m, No. 17-07-00510_a and No. 16-07-00659_a),

Presidium of RAS programs I.33, I.5, and the POI FEB RAS research Program “Mathematical modeling and analysis of dynamical processes in the ocean”. We thank all *Acoustics@home* volunteers, whose computers took part in the experiment.

References

- [1] Program sources of the Acoustics@home project, <https://github.com/nauchnik/acoustics-at-home>.
- [2] Acoustics@home, <http://www.acousticsathome.ru/boinc/>.
- [3] David P. Anderson. BOINC: A system for public-resource computing and storage. In Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, GRID'04, pages 4–10, Washington, DC, USA, 2004. IEEE Computer Society.
- [4] Konstantin Barkalov and Victor Gergel. Parallel global optimization on GPU. *Journal of Global Optimization*, 66(1):3–20, Sep 2016.
- [5] B. Boashash. *Time-Frequency Signal Analysis and Processing (Second Edition)*. Academic Press, Oxford, 2016.
- [6] Sergey Bochkhanov and Vladimir Bystritsky. Alglib - a cross-platform numerical analysis and data processing library. ALGLIB Project. Novgorod, Russia, 2016.
- [7] Boincstats: Boinc statistics, <https://boincstats.com/>.
- [8] J. Bonnel, C. Gervaise, B. Nicolas, and J. I. Mars. Single-receiver geoacoustic inversion using modal reversal. *The Journal of the Acoustical Society of America*, 131(1):119–128, 2012.
- [9] Julien Bonnel and N. Ross Chapman. Geoacoustic inversion in a dispersive waveguide using warping operators. *The Journal of the Acoustical Society of America*, 130(2):EL101–EL107, 2011.
- [10] Julien Bonnel, Stan E. Dosso, and N. Ross Chapman. Bayesian geoacoustic inversion of single hydrophone light bulb data using warping dispersion analysis. *The Journal of the Acoustical Society of America*, 134(1):120–130, 2013.
- [11] Christophe Cerin and Gilles Fedak. *Desktop Grid Computing*. Chapman & Hall/CRC, 1st edition, 2012.
- [12] Nvidia Corporation. *Cuda c programming guide*, 2017.
- [13] James W Demmel. *Applied numerical linear algebra*. SIAM, 1997.
- [14] Jan Dettmer, Stan Dosso, and Charles W. Holland. Automated geoacoustic inversion and uncertainty: Meso-scale seabed variability in shallow water environments. Report, The Office of Naval Research, 2012.
- [15] Yu. G. Evtushenko, S. A. Lurie, M. A. Posypkin, and Yu. O. Solyaev. Application of optimization methods for finding equilibrium states of two-dimensional crystals. *Computational Mathematics and Mathematical Physics*, 56(12):2001–2010, 2016.
- [16] A. Holohan. *Community, Competition and Citizen Science: Voluntary Distributed Computing in a Globalized World*. Global Connections. Taylor & Francis, 2016.
- [17] Robert Hooke and T. A. Jeeves. “direct search” solution of numerical and statistical problems. *J. ACM*, 8(2):212–229, April 1961.
- [18] P. Hursky and M. B. Porter. Accelerating underwater acoustic propagation modeling using general purpose graphic processing units. In OCEANS'11 MTS/IEEE KONA, pages 1–6, Sept 2011.
- [19] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt. *Computational ocean acoustics*. Springer, New-York et al, 2011.
- [20] B. G. Katsnelson, V. G. Petnikov, and J. Lynch. *Fundamentals of Shallow Water Acoustics*. Springer US, New-York et al, 2012.
- [21] Irkutsk Supercomputer Center of SB RAS, <http://hpc.icc.ru>.
- [22] Vladimir V. Mazalov, Natalia N. Nikitina, and Evgeny E. Ivashko. Task scheduling in a desktop grid to minimize the server load. In Proceedings of the 13th International Conference on Parallel Computing Technologies - Volume 9251, pages 273–278, New York, NY, USA, 2015. Springer-Verlag New York, Inc.
- [23] P. S. Petrov. A method for single-hydrophone geoacoustic inversion based on the modal group velocities estimation: Application to a waveguide with inhomogeneous bottom relief. In Proceedings of the International Conference Days on Diffraction 2014, pages 186–191, May 2014.
- [24] P. S. Petrov, A. D. Zakharenko, and M. Yu. Trofimov. The wave equation with viscoelastic attenuation and its application in problems of shallow-sea acoustics. *Acoustical Physics*, 58(6):700–707, 2012.
- [25] Graham A. Warner, Stan E. Dosso, Jan Dettmer, and David E. Hannay. Bayesian environmental inversion of airgun modal dispersion using a single hydrophone in the chukchi sea. *The Journal of the Acoustical Society of America*, 137(6):3009–3023, 2015.
- [26] Oleg Zaikin and Pavel Petrov. Algorithm of reconstruction of the sound speed profile in a shallow-water geoacoustic waveguide from modal dispersion data. *Optoelectronics, Instrumentation and Data Processing*, 52(3):259–265, 2016.
- [27] Oleg Zaikin and Pavel Petrov. Application of iterative hill climbing to the sound speed profile inversion in underwater acoustics. In Proc. of The Fifth International Workshop on Mathematical Models and their Application, volume 173 of IOP Conf. Series: Materials Science and Engineering, pages 1–8, 2017.