Parallel Numerical Simulation of Seismic Waves Propagation with Intel Math Kernel Library

Mikhail A. Belonosov[1], Clement Kostov[2], Galina V. Reshetova[3],
Sergey A. Soloviev[1], and Vladimir A. Tcheverda[1]

[1] Institute of Petroleum Geology and Geophysics SB RAS, Novosibirsk, Russia
[2] Schlumberger Moscow Research
[3] Institute of Computational Mathematics and Mathematical geophysics SB RAS, Novosibirsk, Russia

Abstract. This paper describes the implementation of parallel computing to model seismic waves in heterogeneous media based on Laguerre transform with respect to time. The main advantages of the transform are a definite sign of the spatial part of the operator and its independence of the parameter of separation. This property allows one to efficiently organize parallel computations by means of decomposition of the computational domain with successive application of the additive Schwarz method. At each step of the Schwarz alternations, a system of linear algebraic equations in each subdomain is resolved independently of all the others. A proper choice of Domain Decomposition reduces the size of matrices and ensures the use of direct solvers, in particular, the ones based on LU decomposition. Thanks to the independence of the matrix of the parameter of Laguerre transform with respect to time, LU decomposition for each subdomain is done only once, saved in the memory and used afterwards for different right-hand sides.

A software is being developed for a cluster using hybrid OpenMP and MPI parallelization. At each cluster node, a system of linear algebraic equations with different right-hand sides is solved by the direct sparse solver PARDISO from Intel Math Kernel Library (Intel MKL). The solver is extensively parallelized and optimized for the high performance on many core systems with shared memory. A high performance parallel algorithm to solve the problem has been developed. The algorithm scalability and efficiency is investigated. For a two-dimensional heterogeneous medium, describing a realistic geological structure, which is typical of the North Sea, the results of numerical modeling are presented.

1 Introduction

The large-scale numerical simulation of elastic wave propagation in realistic 3D heterogeneous media is impossible without parallel computations based on domain decomposition. So far, the most popular approach here is to use explicit finite-difference schemes based on staggered grids, despite drawbacks such as the necessity to perform data send/receive at each time step, full re-simulation of

P. Manninen and P. Öster (Eds.): PARA 2012, LNCS 7782, pp. 153–167, 2013.

the wavefield for each new source, and hard disk data storage for implementation of a reverse-time migration. In this regard, considerable attention has recently been given to the development of alternative techniques for simulation of seismic waves, especially, the ones working in the temporal frequency domain [1]. However, the use of such methods for general heterogeneous media also faces a range of significant issues. The main issue is a consequence of the indefiniteness of the impedance matrix. This property brings about a very slow rate of convergence for the iterative procedures solving the linear algebraic equations resulting from the finite-dimensional approximation of the elastic wave equations in the temporal frequency domain.

Our motivation is to overcome this difficulty and to do that, we apply the approach based on the Laguerre transform with respect to time. This leads to a uniformly elliptic system of linear equations [2] and, so, ensures the convergence of the Schwarz alternations [3], based on a suitable domain decomposition with overlapping [4]. We choose the domain decomposition providing the possibility of applying in each of elementary subdomains the LU factorization of the corresponding matrix. Once LU decomposition is performed, it is stored and later used for each component of the Laguerre decomposition and each source position. It should be stressed that it is a consequence of the main advantage of the Laguerre transform: the matrix of the corresponding system does not depend on the separation parameter and, hence, the LU factorization in each subdomain is performed only once.

2 Statement of the Problem: Separation of Time

Let us consider a 2D system of second order elastic equations for a volumetric source with zero initial conditions:

$$\left.\begin{aligned}
&\rho\tfrac{\partial^2 u_1}{\partial t^2} = \tfrac{\partial}{\partial x}\left[(\lambda+2\mu)\tfrac{\partial u_1}{\partial x} + \lambda\tfrac{\partial u_2}{\partial z}\right] + \tfrac{\partial}{\partial z}\left[\mu\tfrac{\partial u_2}{\partial x} + \mu\tfrac{\partial u_1}{\partial z}\right] + g(x,z)f'(t);\\
&\rho\tfrac{\partial^2 u_2}{\partial t^2} = \tfrac{\partial}{\partial x}\left[\mu\tfrac{\partial u_2}{\partial x} + \mu\tfrac{\partial u_1}{\partial z}\right] + \tfrac{\partial}{\partial z}\left[\lambda\tfrac{\partial u_1}{\partial x} + (\lambda+2\mu)\tfrac{\partial u_2}{\partial z}\right] + h(x,z)f'(t);\\
&u_j\big|_{t=0} = \tfrac{\partial u_j}{\partial t}\big|_{t=0} = 0, \quad j=1,2.
\end{aligned}\right\} \tag{1}$$

Here ρ is density, λ, μ are Lamet coefficients (P, S-velocities is defined in the following way: $V_p = \sqrt{\frac{\lambda+2\mu}{\rho}}$ and $V_s = \sqrt{\frac{\mu}{\rho}}$). The functions $g(x,z)$ and $h(x,z)$ reflect spacial distribution of the source, $f(t)$ is the source function. Further as a function $f(t)$ we will take the Richer impulse with dominant frequency ν_0:

$$f(t) = \left[1 - 2\pi^2\nu_0^2\left(t - \frac{t_0}{\nu_0}\right)^2\right] \cdot e^{-\pi^2\nu_0^2\left(t-\frac{t_0}{\nu_0}\right)^2}.$$

2.1 Laguerre Transform and Additive Schwarz Method

The integral Laguerre transform for the function $F(t) \in L_2(0, \infty)$ is given by the following relation:

$$F_n = \int_0^\infty F(t)(ht)^{-\frac{\alpha}{2}} l_n^\alpha(ht)dt, \tag{2}$$

with the inversion formula

$$F(t) = \sum_{n=0}^{\infty} F_n \cdot (ht)^{\frac{\alpha}{2}} l_n^\alpha(ht). \tag{3}$$

Here $l_n^\alpha(ht)$ are orthonormal Laguerre functions

$$l_n^\alpha(ht) = \sqrt{\frac{n!}{(n+\alpha)!}}(ht)^{\frac{\alpha}{2}} e^{-\frac{ht}{2}} L_n^\alpha(ht),$$

with $h \in R_+$, $\alpha \in Z_+$ and $L_n^\alpha(ht)$ being classical Laguerre polynomials [5]:

$$L_n^\alpha(y) = \frac{1}{n!} e^y y^{-\alpha} \frac{d^n}{dy^n}\left(y^{\alpha+n} e^{-y}\right). \tag{4}$$

h is the scaling parameter and it is responsible for dilation/compression of the Laguerre functions. The parameter α reflects the attenuation rate i.e. for a smaller size of α – the Laguerre function attenuate faster (see Figure 1).

Application of the integral Laguerre transform (2) to the system of elastic equations (1) transforms it to the system of eliptic second order partial differential equations with a negative definite operator:

$$\left.\begin{aligned}
&\tfrac{\partial}{\partial x}\left[(\lambda+2\mu)\tfrac{\partial u_1^n}{\partial x} + \lambda\tfrac{\partial u_2^n}{\partial z}\right] + \tfrac{\partial}{\partial z}\left[\mu\tfrac{\partial u_2^n}{\partial x} + \mu\tfrac{\partial u_1^n}{\partial z}\right] - \rho\tfrac{h^2}{4}u_1^n = (\hat{u}_1)_{n-1};\\
&\tfrac{\partial}{\partial x}\left[\mu\tfrac{\partial u_2^n}{\partial x} + \mu\tfrac{\partial u_1^n}{\partial z}\right] + \tfrac{\partial}{\partial z}\left[\lambda\tfrac{\partial u_1^n}{\partial x} + (\lambda+2\mu)\tfrac{\partial u_2^n}{\partial z}\right] - \rho\tfrac{h^2}{4}u_2^n = (\hat{u}_2)_{n-1}.
\end{aligned}\right\}$$

Its right-hand side is defined by the recurrence formulas

$$(\hat{u}_1)_{n-1} = \rho h^2 \sqrt{\tfrac{n!}{(n+\alpha)!}} \sum_{k=0}^{n-1} (n-k) \sqrt{\tfrac{(k+\alpha)!}{k!}} u_1^k - g(x,z) f_n,$$

$$(\hat{u}_2)_{n-1} = \rho h^2 \sqrt{\tfrac{n!}{(n+\alpha)!}} \sum_{k=0}^{n-1} (n-k) \sqrt{\tfrac{(k+\alpha)!}{k!}} u_2^k - h(x,z) f_n.$$

In order to define how many Laguerre functions should be used in expansion (3) an empirical criterion is applyed. It is based on the fact that the waveform of volumetric source in a 3D homogeneous medium coincides with the first derivation of the source function. That is why the number of Laguerre functions N is chosen from the condition:

$$\int_0^T \left[f'(t-T) - \sum_{n=0}^{N(T)} f_n (ht)^{-\alpha/2} l_n^\alpha (ht) \right]^2 dt \leq \varepsilon^2$$

that ensures the prescribed accuracy of the root-mean-square deviation of an initial impulse from its expansion by the Laguerre functions on the time interval $(0,T)$. An example of such a choice, but for an insufficient number of Laguerre functions is shown in Figure 2.

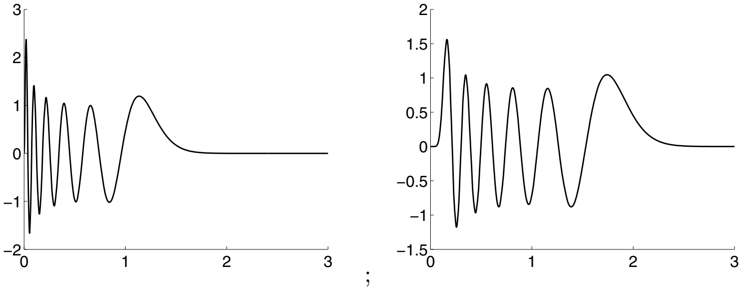

Fig. 1. The Laguerre functions $l_n^\alpha(ht)$ with $n = 10$, $h = 100$ for different values of α: $\alpha = 5$ on the left; $\alpha = 20$ on the right

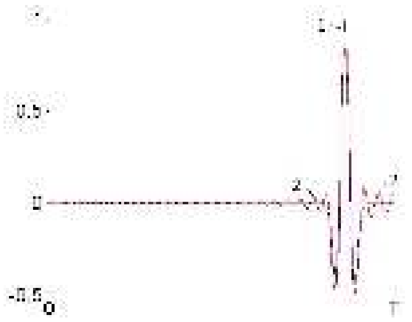

Fig. 2. Line 1 – the Richer impulse, line 2 – its reconstruction for some value of N

3 Numerical Approximation and Organization of Parallel Calculations

Parallelization of the algorithm is implemented on the base of the domain decomposition and the additive Schwarz method.

3.1 Additive Schwarz Method

A full description of the additive Schwarz method can be found in [3,6]. The basic idea of this method is to search for the solution not in the original computational domain, if it is too large, but to decompose it to elementary subdomains of an appropriate size and to resolve the problem in each of these subdomains. In particular, to resolve the boundary value problem in the domain D with the boundary S, it is decomposed to two overlapping subdomains D_1 and D_2 (Figure 3), so two new boundaries S_1 and S_2 are introduced. The Schwarz alternations start with computation of solutions within subdomains D_1 and D_2 with arbitrary boundary conditions on S_1 and S_2, respectively. For each subsequent iteration (m+1), the solution in D_1 is constructed using as boundary conditions on S_1 the trace of a solution in D_2 computed by the previous iteration (m). The same procedure is used to update the solution in D_2. The convergence of iterations for this version of the additive Schwarz method is ensured by the negative definiteness of the operator and overlapping of the neighboring subdomains [3,6].

As a stopping criterion for the Schwarz alternations, we should attain the desired level of threshold of the following value:

$$E_{rr} = \max\left(\frac{\|u_1^n - u_1^{n-1}\|_\Gamma}{\|u_1^{n-1}\|_\Gamma}, \frac{\|u_2^n - u_2^{n-1}\|_\Gamma}{\|u_2^{n-1}\|_\Gamma}\right), \tag{5}$$

where E_{rr} characterizes a relative correlation of the solution on two sequential time steps, Γ is the unification of all the boundaries introduced by domain decomposition on overlapping interfaces. In our numerical simulation, the threshold $E_{rr} \leq 10^{-5}$.

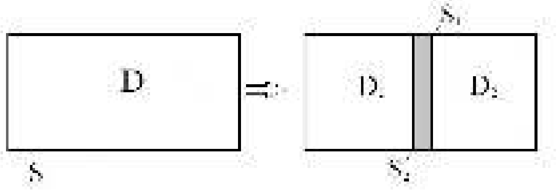

Fig. 3. The overlapping domain decomposition and Schwarz iterations

3.2 Restriction of the Computational Domain

In order to restrict the target area, we use a certain modification of the elastic Perfectly Matched Layer (PML) presented in [7]. Such a modification was proposed and implemented by G.V. Reshetova and V.A. Tcheverda [8]. The main idea is to introduce the PML for a system of first order elastic equations and then to implement the Laguerre transform. As a result, we obtain the following system of equations:

$$\left.\begin{aligned}
&\rho\left(\tfrac{h}{2}+d_x(x)\right)u^n_{1,1}=\tfrac{\partial\sigma^n_1}{\partial x}-\rho(\bar{u}_{1,1})_{n-1},\\
&\rho\left(\tfrac{h}{2}+d_z(z)\right)u^n_{1,2}=\tfrac{\partial\sigma^n_3}{\partial z}-\rho(\bar{u}_{1,2})_{n-1},\\
&\rho\left(\tfrac{h}{2}+d_x(x)\right)u^n_{2,1}=\tfrac{\partial\sigma^n_3}{\partial x}-\rho(\bar{u}_{2,1})_{n-1},\\
&\rho\left(\tfrac{h}{2}+d_z(z)\right)u^n_{2,2}=\tfrac{\partial\sigma^n_2}{\partial z}-\rho(\bar{u}_{2,2})_{n-1},\\
&\left(\tfrac{h}{2}+d_x(x)\right)\sigma^n_{1,1}=(\lambda+2\mu)\tfrac{\partial u^n_1}{\partial x}-(\bar{\sigma}_{1,1})_{n-1}+G(x,z)f_n,\\
&\left(\tfrac{h}{2}+d_z(z)\right)\sigma^n_{1,2}=\lambda\tfrac{\partial u^n_2}{\partial z}-(\bar{\sigma}_{1,2})_{n-1},\\
&\left(\tfrac{h}{2}+d_x(x)\right)\sigma^n_{2,1}=\lambda\tfrac{\partial u^n_1}{\partial x}-(\bar{\sigma}_{2,1})_{n-1}+H(x,z)f_n,\\
&\left(\tfrac{h}{2}+d_z(z)\right)\sigma^n_{2,2}=(\lambda+2\mu)\tfrac{\partial u^n_2}{\partial z}-(\bar{\sigma}_{2,2})_{n-1},\\
&\left(\tfrac{h}{2}+d_x(x)\right)\sigma^n_{3,1}=\mu\tfrac{\partial u^n_2}{\partial x}-(\bar{\sigma}_{3,1})_{n-1},\\
&\left(\tfrac{h}{2}+d_z(z)\right)\sigma^n_{3,2}=\mu\tfrac{\partial u^n_1}{\partial z}-(\bar{\sigma}_{3,2})_{n-1},
\end{aligned}\right\},\tag{6}$$

where $(\bar{u}_{1,1})_{n-1}$, $(\bar{u}_{1,2})_{n-1}$, $(\bar{u}_{2,1})_{n-1}$, $(\bar{u}_{2,2})_{n-1}$, $(\bar{\sigma}_{1,1})_{n-1}$, $(\bar{\sigma}_{1,2})_{n-1}$, $(\bar{\sigma}_{2,1})_{n-1}$, $(\bar{\sigma}_{2,2})_{n-1}$, $(\bar{\sigma}_{3,1})_{n-1}$, $(\bar{\sigma}_{3,2})_{n-1}$ are calculated by the following recurrent relation:

$$(\bar{w})_{n-1}=h\sqrt{\tfrac{n!}{(n+\alpha)!}}\sum_{k=0}^{n-1}\sqrt{\tfrac{(k+\alpha)!}{k!}}w^k.$$

Recall that unknown functions inside the PML are split to the two components

$$u_1=u_{1,1}+u_{1,2},\quad u_2=u_{2,1}+u_{2,2};$$

$$\sigma_{xx}=\sigma_{xx,1}+\sigma_{xx,2},\quad \sigma_{zz}=\sigma_{zz,1}+\sigma_{zz,2},\quad \sigma_{xz}=\sigma_{xz,1}+\sigma_{xz,2};$$

and seismic wave absorption is provided by special damping functions along the axes x and z:

$$\left.\begin{aligned}
&\begin{pmatrix}d_x(x)\\ d_z(z)\end{pmatrix}=0,\text{if }\begin{pmatrix}x\le a\\ z\le a\end{pmatrix};\\
&\begin{pmatrix}d_x(x)\\ d_z(z)\end{pmatrix}=\left.\begin{matrix}d_0\left(\frac{x-a}{\delta}\right)^4\\ d_0\left(\frac{z-a}{\delta}\right)^4\end{matrix}\right\},\text{otherwise}
\end{aligned}\right\},$$

with a numerical value

$$d_0 = \left|\ln\left(\frac{1}{R}\right)\right| \frac{2\max\limits_{x,z}(v_p(x,z))}{\delta}.$$

Here a is the beginning of the PML, δ is its width, R is the value of a desired level of artificial reflection from the PML (in our experiments, it was taken $R = 10^{-5}$).

Next let us multiply each odd equation of the system of equations (6) by $d_2 = d_z(z)$, each even equation by $d_1 = d_x(x)$ and after several substitutions we obtain the following system of second order partial differential equations:

$$\left.\begin{aligned}
&\left(\tfrac{h}{2}+d_2\right)\tfrac{\partial}{\partial x}\left[\tfrac{(\lambda+2\mu)}{\frac{h}{2}+d_1}\tfrac{\partial u_x^n}{\partial x}\right]+\tfrac{\partial}{\partial x}\left[\lambda\tfrac{\partial u_z^n}{\partial z}\right]+\tfrac{\partial}{\partial z}\left[\mu\tfrac{\partial u_z^n}{\partial x}\right]+\\
&+\left(\tfrac{h}{2}+d_1\right)\tfrac{\partial}{\partial z}\left[\tfrac{\mu}{\frac{h}{2}+d_2}\tfrac{\partial u_x^n}{\partial z}\right]-\rho\left(\tfrac{h}{2}+d_1\right)\left(\tfrac{h}{2}+d_2\right)u_x^n=\\
&=\rho\left[\left(\tfrac{h}{2}+d_2\right)(\bar{u}_{x,1})_{n-1}+\left(\tfrac{h}{2}+d_1\right)(\bar{u}_{1,2})_{n-1}\right]+\\
&+\left(\tfrac{h}{2}+d_2\right)\tfrac{\partial}{\partial x}\left[\tfrac{(\bar{\sigma}_{1,1})_{n-1}}{\frac{h}{2}+d_1}+\tfrac{(\bar{\sigma}_{1,2})_{n-1}}{\frac{h}{2}+d_2}\right]+\\
&+\left(\tfrac{h}{2}+d_1\right)\tfrac{\partial}{\partial z}\left[\tfrac{(\bar{\sigma}_{3,1})_{n-1}}{\frac{h}{2}+d_1}+\tfrac{(\bar{\sigma}_{3,2})_{n-1}}{\frac{h}{2}+d_2}\right]-\\
&-\tfrac{d_2' f_n\left(\frac{h}{2}+d_2\right)}{\frac{h}{2}+d_1}+\tfrac{d_1'\delta(x-x_0,z-z_0)f_n\left(\frac{h}{2}+d_2\right)}{\left(\frac{h}{2}+d_1\right)^2},
\end{aligned}\right\} \tag{7}$$

$$\left.\begin{aligned}
&\left(\tfrac{h}{2}+d_2\right)\tfrac{\partial}{\partial x}\left(\tfrac{\mu}{\frac{h}{2}+d_1}\tfrac{\partial u_2^n}{\partial x}\right)+\tfrac{\partial}{\partial x}\left[\mu\tfrac{\partial u_1^n}{\partial z}\right]+\tfrac{\partial}{\partial z}\left[\lambda\tfrac{\partial u_1^n}{\partial x}\right]+\\
&+\left(\tfrac{h}{2}+d_1\right)\tfrac{\partial}{\partial z}\left[\tfrac{(\lambda+2\mu)}{\frac{h}{2}+d_2}\tfrac{\partial u_2^n}{\partial z}\right]-\rho\left(\tfrac{h}{2}+d_1\right)\left(\tfrac{h}{2}+d_2\right)u_2^n=\\
&=\rho\left[\left(\tfrac{h}{2}+d_2\right)(\bar{u}_{2,1})_{n-1}+\left(\tfrac{h}{2}+d_1\right)(\bar{u}_{2,2})_{n-1}\right]+\\
&+\left(\tfrac{h}{2}+d_2\right)\tfrac{\partial}{\partial x}\left[\tfrac{(\bar{\sigma}_{3,1})_{n-1}}{\frac{h}{2}+d_1}+\tfrac{(\bar{\sigma}_{3,2})_{n-1}}{\frac{h}{2}+d_2}\right]+\\
&+\left(\tfrac{h}{2}+d_1\right)\tfrac{\partial}{\partial z}\left[\tfrac{(\bar{\sigma}_{2,1})_{n-1}}{\frac{h}{2}+d_1}+\tfrac{(\bar{\sigma}_{2,2})_{n-1}}{\frac{h}{2}+d_2}\right]-\delta_2' f_n+\tfrac{d_2'\delta(x-x_0,z-z_0)f_n}{\frac{h}{2}+d_1}.
\end{aligned}\right\} \tag{8}$$

It is worth mentioning that system (7) – (8) introduces an unsplit PML (see [8]).

3.3 Numerical Approximation

The finite-difference approximation of system (7) – (8) is done by the standard staggered grid scheme [9] that was modified for second order systems [10]. Its stencil is presented in Figure 4. Formally, we need u_1 and u_2 only, all other variables are complement. But their knowledge is necessary for simulation, and so they are computed in the corresponding nodes.

This approximation gives a system of linear algebraic equations with a sparse nine-diagonal matrix. It is worth mentioning that the matrix of this system does not depend on the separation parameter n.

For each value of the separation parameter n, we have a system of linear algebraic equations (SLAE) with the same sparse negative definite matrix but with different right-hand sides. The negative definiteness of the matrix ensures convergence of the additive Schwarz method (see [6]). Since it does not depend on the separation parameter, it is reasonable to use direct solver on the base of LU decomposition: in each subdomain it can be done only once, saved in the RAM and subsequently be used for all right-hand sides.

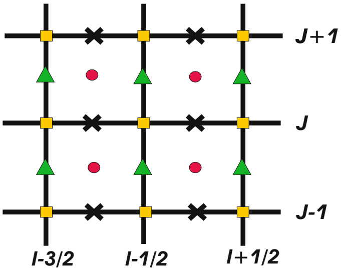

Fig. 4. Stencil of the finite-difference scheme that is used for approximation of system (7) – (8). Squares and circles are for u_1^n and u_2^n respectively, while $\overline{\sigma}$ is calculated in triangles.

3.4 LU FACTORIZATION

In order to perform the LU factorization and to solve a SLAE for a large number of right-hand sides we use Intel Math Kernel Library (Intel MKL) PARDISO direct solver (http://software.intel.com/sites/products/documentation/doclib/mkl_sa/11/mklman/index.htm) that is parallelized via OpenMP. The PARDISO package is a shared-memory multiprocessing parallel direct solver, designed to solve sparse SLAEs. It is based on row-column reordering of an initial matrix, effective parallelization of a factorization and solving steps. To perform the row-column permutation, Intel MKL PARDISO uses a nested dissection algorithm from the METIS package [11] that decreases the size of a required RAM to store LU factors. In Table 1 we present an amount of RAM required for LU decomposition for the problem we deal with. One can conclude that

1. Because of a sparse structure of a matrix we need only a few megabytes of RAM to store a matrix of finite-difference approximation.
2. The main amount of RAM is used to store the LU factorization.

Table 1. Properties of the LU factorization of a nine-diagonal matrix obtained after finite-difference approximation of system (7) – (8)

Domain size, $n_x = n_z$	Matrix size	Nonzero elements of the matrix	Nonzero elements of LU factors	LU (MB)
100	20 200	180 196	1 620 174	12
200	80 400	720 396	8 108 008	62
400	320 800	2 880 796	39 251 440	299
800	1 281 600	11 521 596	187 492 542	1 430
1 600	5 123 200	46 083 196	858 718 476	6 552

A factorization step of the Intel MKL PARDISO solver is extensively parallelized and optimized for providing a high performance on multi-core systems with shared memory. In order to improve the factorization performance, algorithms of Intel MKL PARDISO are based on a Level-3 BLAS update. Moreover, there are additional features in PARDISO, which can improve the performance, in particular, left-looking [12] and two-level [13] factorization algorithms. The first algorithm improves the scalability on a small number of threads while the second – on many threads (more than eight). The computational cost of solving SLAE with many right-hand sides (RHS) is the same or higher than those needed for factorization. A solving step in Intel MKL PARDISO is optimized both for one RHS and for many RHS. The comparison of PARDISO vs. SuperLU has been made on the cluster of Siberian Supercomputer Center (nodes based on X5675 3.00 GHz Westmere). The results obtained are presented in the chart:

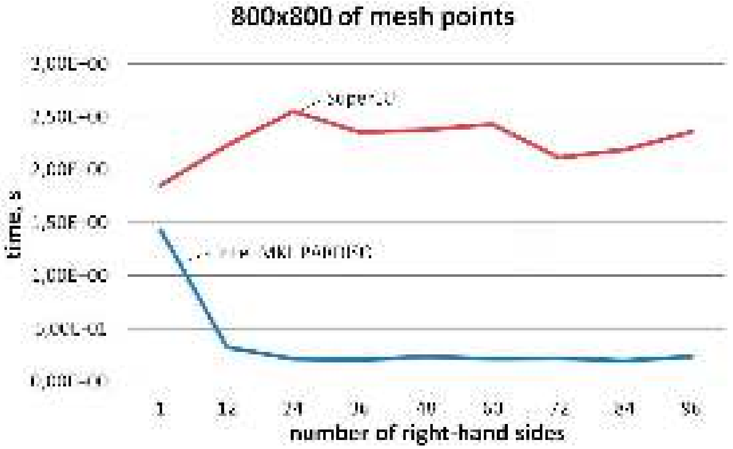

Fig. 5. Dependence the of normalized solving time on the number of RHS

3.5 Parallel Computations

High performance computations are usually performed on computational systems with distributed memory or with MPP (Massive Parallel Processing) architecture. Such systems consist of several nodes with several processors. Each of them has access to RAM of this node. Each processor is usually multi-core. That is why modern computational systems have a hybrid architecture and organized as a set of nodes with distributed memory (MPP architecture), each of them is also computational system with shared memory. Proposed numerical algorithm is orientated onto such architectures and can be effectively loaded at any cluster and consists of the following steps:

1. domain decomposition on "elementary" subdomains providing possibility to store the LU factorization in shared memory of a node (see Table 1);
2. usage of PARDISO MKL for effective parallel computations on the node with OpenMP (SMP architecture);
3. exchanges between subdomains in the process of the Schwarz iterations via MPI (MPP architecture).

The scheme of parallel computations is presented in Figure 6. Numerical experiments were carried out on computational systems with a hybrid parallel architecture. At each node, there are 8 GB RAM, so we can decompose our computational domain to squares of 800 × 800 mesh points.

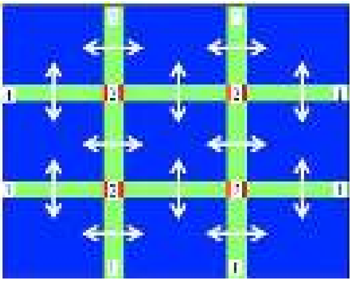

Fig. 6. Parallel computations. Strips 1 correspond to overlapping of two neighbors, while rectangles 2 match overlapping of four subdomains. Arrows correspond to the direction of exchanges between the nodes (MPI).

4 Numerical Experiments

Numerical experiments were carried out on the high performance computer of the Moscow State University with a hybrid parallel architecture: 519 nodes, each of them consists of two quad-core processors and has 8 GB RAM.

The first experiments were performed to understand the main properties of the method. We have considered a simplest situation: a homogeneous elastic medium with the wave propagation velocities $V_p = 2500$ m/s, $V_s = 2000$ m/s and the density $\rho = 2000$ kg/m^3. As the source function, we chose Ricker impulse with dominant frequency 30 Hz. Parameters of the Laguerre transform: $h = 300$, $\alpha = 5$. For the total simulation time $T = 3$ s, 550 polynomials were used. The size of the computational domain was 1000×1000 m with PML.

4.1 Dependence of the Number of Iterations on the Width of Overlapping

First of all, the dependence of the number of iterations on the width of overlapping is analyzed. These results are presented in Table 2. As one can see, the optimal overlapping is equal to 25 points.

Table 2. Dependence of number of iterations on width of the overlapping

Width of the overlapping (m)	Width of the overlapping (number of mesh points)	Number of iterations
30	15	7
40	20	6
50	25	5
70	35	5

Next both weak and strong scalabilities (see [14]) of the algorithm were studied.

4.2 Weak Scalability

Schematically, the way to estimate weak scaling is presented in Figure 7. We fix the load of a node that is equal to the size of the subdomain and enlarge the size of the global computational domain. Thereby we enlarge the number of nodes. As the measure for weak scaling, we use the following function:

$$eff_{weak}(N) = \frac{T(N)}{T(N_0)}, \tag{9}$$

where $T(N)$ is the calculation time for N nodes as long as the size of the problem N times increases (Figure 7). The ideal weak scalability corresponds to $eff_{weak}(n) \equiv 1$.

For the numerical experiment, we took 800×800 mesh points on each node. In Figure 8 the curve $T(N)/T(9)$ is presented. As can be seen, it has a low variations around the one that reflects good weak scalability of the algorithm.

4.3 Strong Scalability

Schematically the way to estimate strong scaling is presented in Figure 9. In contrast to weak scaling, here we fix the size of the global computational domain

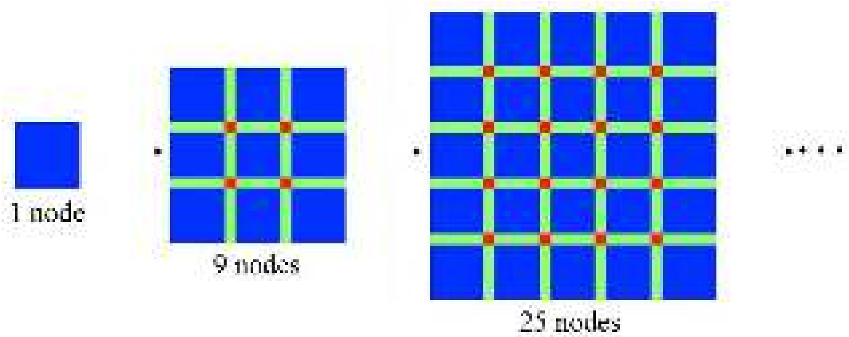

Fig. 7. Weak scaling comuputation

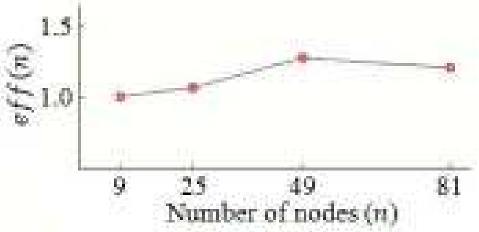

Fig. 8. Weak scalability of the algorithm

and enlarge the number of subdomains. To estimate strong scalability, we use the following function:

$$eff_{strong}(N) = \frac{T(N)}{N_0 \cdot T(N_0)}, \tag{10}$$

where N_0 is the initial number of processes. The ideal strong scalability should coincide with the linear dependence of calculation time on the number of processes, that is $T(N) = \alpha N^{-1}$, where α is certain coefficient. That is why for the ideal strong scalability $eff_{strong}(N) = N^{-1}$.

Numerical experiments to estimate strong scalability were carried out in the same conditions as previously: a homogeneous medium, the Richer impulse with dominant frequency 30 Hz, 550 Laguerre functions, a global computational domain of 2400×2400 mesh points, that is, nine initial subdomains of 800× 800 points, each of them being loaded on its own node.

The result (function 10) is presented in Figure 10 (line 2). Line 1 is the ideal scalability. Thus, one can conclude that the algorithm possesses satisfactory strong scalability as well.

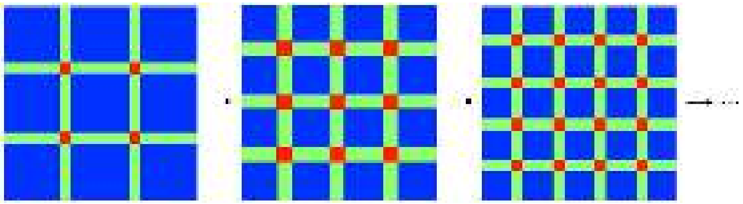

Fig. 9. Strong scaling comuputation

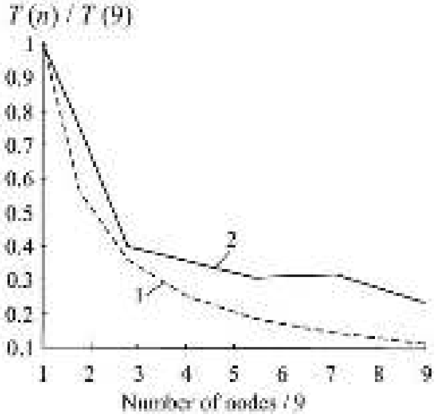

Fig. 10. Strong scalability of the algorithm: line 1 is the ideal scalability, line 2 is the scalability obtained in the numerical experiment

5 Numerical Experiment for Realistic Model

Finally, we would like to present the results of the numerical experiment for presented in Fig.11 (on the left) the realistic Gullfaks model, describing some geological area of the North Sea [15].

The volumetric point source with coordinates (1620, 20) radiates a Ricker impulse with the dominant frequency 30 Hz. The model was discretized on the uniform grid $h_x = h_z = 2$ m that corresponds to 25 mesh points on a wavelength. The integral Laguerre transform with 550 harmonics is computed with $h = 300$ and $\alpha = 3$. The total simulation time is $T = 3$ s. The computational domain is decomposed to (3×3) identical subdomains with overlapping of 50 m (25 points). To obtain solution with the residual $E_{rr} \leq 10^{-5}$ (see (5)) it took 10 Schwarz iterations. The result of this experiment is presented in Figure 11 (on the right).

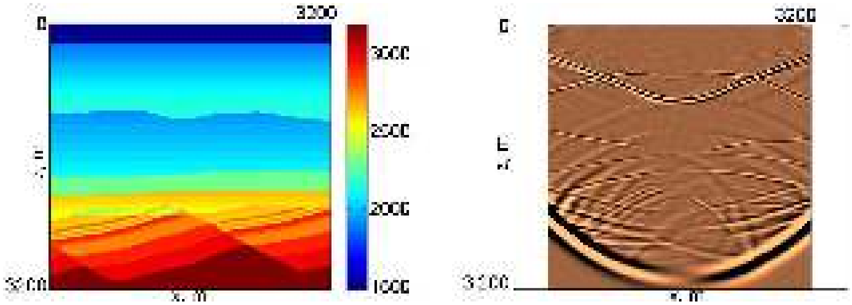

Fig. 11. The 2D Gullfaks model: P-wave velocity (left image); a snapshot for the Gullfaks model (right image)

6 Conclusions

This paper presents the algorithm for the numerical simulation of elastic waves in an inhomogeneous medium based on decomposition of a computational domain, implementation of the integral Laguerre transform and the additive Schwarz method. This algorithm is ideally suited to parallel high-performance computers with a hybrid architecture, representing a set of nodes that combine several multi-core processors with shared memory that are unified by InfiniBand to exchange the data between parallel processes at different nodes. The system of linear algebraic equations for each subdomain is solved with the use of PARDISO from Intel Math Kernel Library (Intel MKL), which is parallelized via OpenMP. The revealed scalability of the algorithm confirms the prospects of the numerical simulation on the base of this algorithm in 3D inhomogeneous media.

To conclude, let us point out that the LU factorization is not the only way to solve the system obtained after numerical approximation. In particular we can use the Cholesky factorization for interior subdomains (outside the PML) and the LU factorization for boundary subdomains. Also one can use an approximation of sparse matrices by matrices of a lower rank (see e.g. [16])).

Acknowledgements. The research described was performed in cooperation with Schlumberger Moscow Research and partially supported by RFBR grants 11-05-00947, 12-05-31008 and President of Russia grant for young scientists MK-77.2013.5. We are also grateful to the R&D site of Intel at Novosibirsk (MKL group) for assistance while application of PARDISO from Intel Math Kernel Library.

References

1. Plessix, R.E.: A Helmholtz iterative solver for 3D seismic-imaging problems. Geophysics 72(5), 185–194 (2007)
2. Mikhailenko, B.G., Mikhailov, A.A., Reshetova, G.V.: Numerical viscoelastic modeling by the spectral Laguerre method. Geophysical Prospecting 51, 37–48 (2003)

3. Chan, T., Mathew, T.P.: Domain decomposition. Acta Numerica 3, 61–143 (1994)
4. Gander, M., Halpern, L., Nataf, F.: Optimized Schwarz Methods. In: 12th International Conference on Domain Decomposition Methods, pp. 15–27 (2001)
5. Suetin, P.K.: Classical orthogonal polynomials. Nauka, M. 203–243 (1974)
6. Nepomnyashchikh, S.V.: Domain decomposition methods. Radon Series Comput. Appl. Math. 1, 81–159 (2007)
7. Collino, F., Tsogka, C.: Application of the PML absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media. Geophysics 66(1), 294–307 (2001)
8. Reshetova, G.V., Tcheverda, V.A.: Implementation of Laguerre transform to construct perfectly matched layer without splitting. Mathematical Modelling 18(1), 91–101 (2006)
9. Virieux, J.: P-SV wave propagation in heterogeneous media: Velocity - stress finite-difference method. Geophysics 51(4), 889–901 (1986)
10. Zahradnik, J., Priolo, E.: Heterogeneous formulations of elastodynamic equations and finite-difference schemes. Geophysical Journal International 120(3), 663–676 (1995)
11. Karypis, G., Kumar, V.: A fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. SIAM Journal on Scientific Computing 20(1), 359–392 (1998)
12. Schenk, O., Gartner, K., Fichtner, W.: Efficient Sparce LU Factorization with Left-right Looking Strategy on Shared Memory Multiprocessors. BIT 240(1), 158–176 (2000)
13. Schenk, O., Gartner, K.: Two-level scheduling in PARDISO: Improved Scalability on Shared Memory Multiprocessing Systems. Parallel Computing 28, 187–197 (2002)
14. Colella, P., Bell, J., Keen, N., Ligocki, T., Lijewski, M., van Straalen, B.: Performance and scaling of locally-structured grid methods for partial differential equations. Journal of Physics: Conference Series 78, 012013 (2007)
15. Fossen, H., Hesthammer, J.: Structural geology of the Gullfaks Field, northern North Sea, vol. 127, pp. 231–261. Geological Society, London (1998) (Special Publications)
16. Zhang, Z., Zha, H., Simon, H.: Low-Rank Approximation with Sparce Factors: Basic Algorithms and Error Analysis. SIAM J. of Matrix Analysis and Applications (3), 706–727 (1999)