Parallel simulation of time-domain acoustic wave propagation

Jure Močnik - Berljavac[1,2], Jure Slak[1,2], Gregor Kosec[1]
[1]"Jožef Stefan" Institute, Parallel and Distributed Systems Laboratory, Ljubljana, Slovenia
[2]Faculty of Mathematics and Physics, University of Ljubljana, Ljubljana, Slovenia
jure.mocnik---berljavac@student.fmf.uni-lj.si jure.slak@ijs.si, gregor.kosec@ijs.si

Abstract – Simulations of acoustic wave propagation are an important tool for reconstructing structure of Earth's subsurface. The core of such simulations is an efficient and accurate method of time-domain acoustic wave simulation in an inhomogeneous domain. Radial Basis Function-generated Finite Differences (RBF-FD) method is a popular variant of local strong form meshless methods that does not require predefined connections between nodes, making it easier to adapt node distribution to the problem under consideration. This paper explores RBF-FD as an alternative to traditionally FDM based methods for time-domain wave propagation. It is demonstrated that RBF-FD provides accurate results even in challenging cases, where conventional methods struggle to even obtain a stable solution.

I. Introduction

A reconstruction of Earth's properties from measurements at the surface can be done by analyzing the acoustic waves that propagate through the Earth. The essential parts of such an approach are a propagation model and its numerical solution. From a numerical point of view an obvious problem arises from the fact that the computational domain cannot cover the whole physical domain and therefore artificial boundaries have to be enforced, which causes nonphysical reflections.

Finite Difference Method (FDM) is often used for such simulations, as it provides an excellent compromise between computational efficiency and accuracy. FDM is also compatible with Absorbing Boundary Conditions [1], one of the simplest methods for suppressing unwanted reflections from the boundaries. Nevertheless, FDM has its shortcomings, namely it suffers from relative inflexibility regarding the considered domain, which arises from its definition on a static uniform grid, while the medium properties can drastically change on relatively small scales. Such cases force a sacrifice to be made either in terms of accuracy or computational efficiency. Limitations of other mesh based method were discussed also in [2] and [3].

In this paper we address the shortcomings of FDM using a solution procedure based on Radial Basis Function-generated Finite Differences (RBF-FD) method [4], a popular form of local strong form meshless methods, which has shown promise with several successful use cases and is still actively researched [5, 6, 7]. RBF-FD offers high flexibility regarding approximation order [6], treating complex domains [8], potential for effective parallel implementation [9] and high control over the complexity vs. accuracy/stability, which makes it an impressive tool for treating problems such as acoustic wave propagation.

The rest of the paper is organized as follows: in section II RBF-FD method and Absorbing boundary conditions in context of RBF-FD are defined, in section III the application of the described method is presented on a concrete problem, in section IV the numerical results are presented to demonstrate the advantages of RBF-FD, and in section V we present the conclusions.

II. Method description

A. Radial Basis Function-generated Finite Difference Method

The RBF-FD method is used for spatial discretization of a partial differential equation (PDE). In this paper an initial value problem will be solved, which requires discretization of derivatives in time, as well. The spatial discretization procedure will be presented first.

Radial Basis Functions are real valued functions $\Phi(r)$ only dependent on the distance from some center point $\boldsymbol{x_i}$

$$\Phi_i(\boldsymbol{x}) = \Phi(r) \text{ where } r = |\boldsymbol{x} - \boldsymbol{x}_i|. \tag{1}$$

In this paper Gaussian Radial Basis Functions are used

$$\Phi(r) = \exp(-r^2/\sigma_B^2), \tag{2}$$

where σ_B stands for the shape parameter.

Consider a differential operator $\mathcal{L}$ acting on a function u in the domain Ω. The domain Ω and its boundary $\partial\Omega$ are populated with N nodes. For each node $\boldsymbol{x}_i$ its n closest neighbors are found, forming the so-called support domain of $\boldsymbol{x}_i$. This is analogous the 5 or 9 neighboring nodes used to construct FDM approximations. Function values at support nodes are used to approximate the operator $\mathcal{L}$ as a weighted linear combination of said values

$$(\mathcal{L}u)(\boldsymbol{x}_i) \approx \sum_{\boldsymbol{x}_j \in \text{ support of } \boldsymbol{x}_i} w_j^i u(\boldsymbol{x}_j) = \boldsymbol{w}_i^\mathsf{T} \cdot \boldsymbol{u}_i, \tag{3}$$

where $u(\boldsymbol{x}_j)$ represents the value of the approximated field at position $\boldsymbol{x}_j$. The expression on the right shortens the notation by representing the sum as a dot product by packaging values inside vectors $\boldsymbol{w}_i^\mathsf{T}$ and $\boldsymbol{u}_i$. As an example, in one dimensional FDM we have the following known approximation for u'':

$$u''(x_i) \approx \begin{bmatrix} \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \end{bmatrix} \begin{bmatrix} u(x_{i-1}) \\ u(x_i) \\ u(x_{i+1}) \end{bmatrix}. \tag{4}$$

In contrast to FDM, RBF-FD uses weights that are not known beforehand as they are dependent on the positions of the nodes in the support domain. To determine the values of the weights w_j^i, equality is enforced in (3) for a set of Radial Basis Functions. In case of this paper the selected functions are Gaussians centered on all the nodes of the support domain. For the k-th function centered in support node p_k of node $\boldsymbol{x}_i$ we write

$$(\mathcal{L}\Phi_k)(\boldsymbol{x}_i) = \sum_{\boldsymbol{x}_j \in \text{ support of } \boldsymbol{x}_i} w_j^i \Phi_k(\boldsymbol{x}_j) = \boldsymbol{w}_i^\mathsf{T} \cdot \boldsymbol{\Phi}_k = \boldsymbol{\Phi}_k^\mathsf{T} \cdot \boldsymbol{w}_i. \quad (5)$$

As the number of functions Φ_k and in turn equations is equal to the number of unknown weights w_j, we have a linear system, which can be presented in matrix form by assembling rows $\boldsymbol{\Phi}_k^\mathsf{T}$ to a matrix:

$$\begin{bmatrix} \boldsymbol{\Phi}_1^\mathsf{T} \\ \vdots \\ \boldsymbol{\Phi}_k^\mathsf{T} \\ \vdots \\ \boldsymbol{\Phi}_n^\mathsf{T} \end{bmatrix} \begin{bmatrix} w_1^i \\ \vdots \\ w_j^i \\ \vdots \\ w_n^i \end{bmatrix} = \begin{bmatrix} (\mathcal{L}\Phi_1)(\boldsymbol{x}_i) \\ \vdots \\ (\mathcal{L}\Phi_k)(\boldsymbol{x}_i) \\ \vdots \\ (\mathcal{L}\Phi_n)(\boldsymbol{x}_i) \end{bmatrix} \quad (6)$$

where both j and k indices run over the nodes in the support domain of node $\boldsymbol{x}_i$. The matrix is symmetric and when Gaussian basis functions are used, it is also positive definite [5]. This guaranties non-singularity as long as all support domain nodes are distinct.

If the positions of the nodes do not change during the simulation, the weights have to be calculated only once at the beginning of the simulation and can be stored for further use. If the boundary conditions include differential operators as well (normal derivative for example), they can be discretized in a similar fashion to operator $\mathcal{L}$.

With weights computed, the operator $\mathcal{L}$ can be approximated with the discretized version, obtained from (3) as

$$(\mathcal{L}u)(\boldsymbol{x}_i) \approx \boldsymbol{w}_i^\mathsf{T} \cdot \boldsymbol{u}_i \quad (7)$$

for each node $\boldsymbol{x}_i$ inside the domain, where $\boldsymbol{u}_i$ still refers to values of nodes in support domain of node $\boldsymbol{x}_i$.

B. Time discretization

In addition to RBF-FD discretization of spatial derivatives, time derivatives must be discretized as well. As solving the wave equation is the goal of this paper only second time derivative must be discretized. The equation

$$\frac{\partial^2 u}{\partial t^2} = \mathcal{L}u \quad (8)$$

is discretized using an explicit scheme

$$\frac{u^t - 2u^{t-1} + u^{t-2}}{\Delta t^2} = \mathcal{L}u^{t-1}, \quad (9)$$

where the superscript index refers to the time step. When $\mathcal{L}u^{t-1}$ is evaluated, the values u^{t-1} and u^{t-2} are already known, either from the initial conditions or from the previous time step. Calculating the value of $(\mathcal{L}u_i^{t-1})(\boldsymbol{x}_i)$ is only a mater of a single dot product, as derived in (7). This formulation also gives rise to parallelization opportunities, since the result for each node $\boldsymbol{x}_i$ can be computed independently within a given time step.

C. Absorbing boundary conditions

In cases where reflections from the boundary are undesirable, some method for suppressing such reflections must be employed in conjunction with the RBF-FD method. Absorbing boundary conditions or ABC [1] is one of the simplest approaches available. The general idea is to introduce a damping factor for nodes inside area close to the boundary of the domain, which will be referred to as absorbing layer. Originally it was formulated in discrete form for FDM on a grid as

$$G(i) = \exp\left(-\left[0.015(i_{max} - i)\right]^2\right), \quad (10)$$

where i_{max} is the integer number of grid points which define the thickens of the absorbing layer; the index i is an integer value and runs from 0 to i_{max}. Usually i_{max} is set to 20. When RBF-DF is used, the nodes are generally not placed on an uniform grid. This necessitates a continuous form of (10),

$$G(x) = \exp\left(-\left[\frac{0.015}{a}(ai_{max} - x)\right]^2\right), \quad (11)$$

where x is the shortest distance from the boundary to the considered node and a is the expected inter nodal distance (which in general is dependent on x).

After this modification, RBF-FD implementation of ABCs continues in same way as in original formulation, i.e. every node inside the absorbing layer has its field value multiplied by $G(x)$ at every time-step.

However, despite the name, ABCs are not true boundary conditions in a sense of Dirichlet or Neumann, as they do not dictate the value or its derivative at the boundary. For this reason Dirichlet's boundary condition $u = 0$ for u on $\partial\Omega$ is used for the actual boundary nodes.

III. Problem formulation

The goal of this paper is to simulate time-domain acoustic wave propagation, where the waves are generated by a point-like source located under the surface of the Earth as depicted in Figure 1.

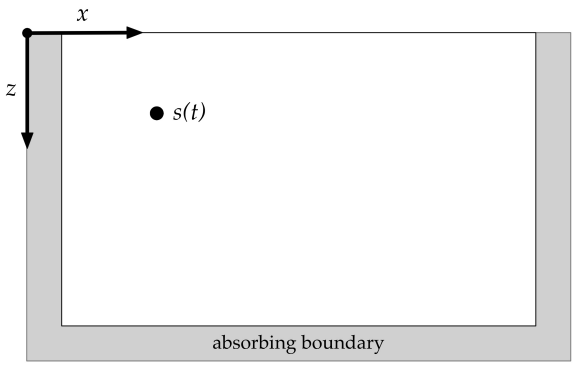

Figure 1. Shematic of the problem.

The problem of acoustic wave propagation is described by the wave equation

$$\frac{1}{v(\boldsymbol{r})}\frac{\partial^2 u(\boldsymbol{r},t)}{\partial t^2} = \nabla^2 u(\boldsymbol{r},t) + s(t)\delta(\boldsymbol{r}-\boldsymbol{r}_s) \tag{12}$$

$$u(\boldsymbol{r},t=0) = 0 \quad \text{and} \quad \frac{\partial u(\boldsymbol{r},t=0)}{\partial t} = 0 \tag{13}$$

where $\boldsymbol{r} = (x,z)$ stands for 2D position vector (z stands for the depth and is measured from the top downwards); $v(\boldsymbol{r})$ stands for position dependent wave velocity, which introduces the properties of the medium in the equation; $s(t)\delta(\boldsymbol{r}-\boldsymbol{r}_s)$ is the Dirac delta function, representing the point-like, time dependent source located at $\boldsymbol{r}_s = (x_s, z_s)$. It is assumed that the system is in equilibrium at the beginning of the simulation.

The problem is solved inside a $500\,\mathrm{m} \times 500\,\mathrm{m}$ square domain. The top boundary ($z = 0$) represents the Earth's surface. Reflections from this boundary are physical and for this reason Dirichlet's boundary conditions are used. All other edges of the domain do not represent any real physical barrier and therefore absorbing boundary conditions are employed. However as explained in Section II.C technically Dirichlet's boundary conditions are used in this case as well, allowing for the same boundary condition on the boundary

$$u = 0, \text{ on } \partial\Omega. \tag{14}$$

The source $s(t)$ is implemented as Ricker's Wavelet

$$s(t) = \frac{2}{\sqrt{3\sigma_R}\pi^{1/4}}\left(1-\left(\frac{t}{\sigma_R}\right)^2\right)e^{-\frac{t^2}{2\sigma_R^2}}, \tag{15}$$

where σ_R is parameter related to flatness of the wavelet. Ricker's wavelet is depicted in Figure 2.

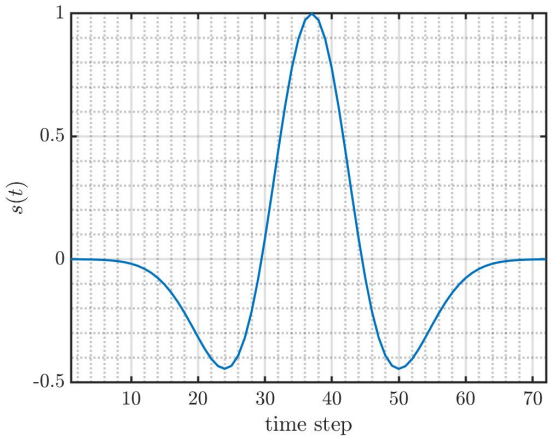

Figure 2. Ricker's wavelet; $\sigma_R = 7.5$.

The $\delta(\boldsymbol{r})$ function is implemented as its approximation

$$\delta(\boldsymbol{r}) = \frac{1}{\pi}\frac{\epsilon}{r^2+\epsilon^2}. \tag{16}$$

where ϵ is a small positive number in units of distance. Selection of ϵ has to be larger than the characteristic distance between nodes, as in this case the results were the most consistent between different discretization methods.

This set-up allow for a variety of different cases to be considered, depending on the choose of $v(\boldsymbol{r},t)$. Case of constant velocity and case of a step in velocity field are presented in next section.

IV. Numerical examples

A. Homogeneous medium

A simple example of the wave propagation problem in homogeneous medium is presented first to validate the RBF-FD solution against the one obtained by FDM. The effect of absorbing boundary conditions will also be compared. The wave velocity is constant throughout the domain and is set to $v = 3000\,\mathrm{m/s}$. A Ricker's wavelet source is located at coordinate $(150\,\mathrm{m}, 150\,\mathrm{m})$, with $\sigma_R = 0.00147\,\mathrm{s}^{-1}$. For δ function approximation parameter $\epsilon = 4\,\mathrm{m}$.

Since the velocity is constant within the domain, RBF-FD was used on uniformly distributed nodes. Poisson Disk Sampling algorithm [7] with constant distance between nodes a is used for node placement. Additionally, regularization algorithm based on simulated annealing and free charged particle simulation is used to move the nodes which might not be placed optimally. This results in randomly placed nodes, the pattern however does resemble a hexagonal grid. The expected distance between nodes is equal to $a = 1.2\,\mathrm{m}$ which corresponds to total number of node $n = 248572$. Time step is chosen as $\Delta t = 0.000196\,\mathrm{s}$. RBF-FD is used with support size of 7 (the node itself and 6 closest neighbors) and shape parameter of Gaussian basis functions is set to $\sigma_B = 70$.

FDM was used in 5-stencil formulation on a uniform grid with a comparable number of nodes $n_{\text{FDM}} = 250000$ – as given by the grid spacing of $a_{\text{FDM}} = 1\,\mathrm{m}$. The time step was the same with both methods.

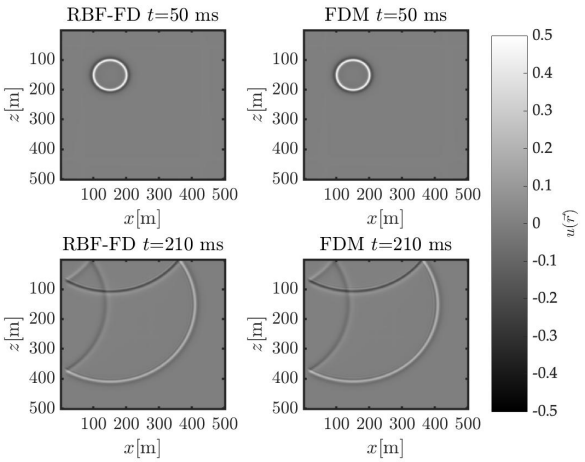

Figure 3. Values of the field u obtained by RBF-FD and FDM at various times.

The Figure 3 displays the comparison of the results from both methods. As positions of the nodes differ between methods, linear interpolation was used in post process. Snapshots of the wave field are provided at two times. We can observe the wave propagating in a perfect circle until it hits the boundary. At time $t = 210\,\mathrm{ms}$ the effect of absorbing boundary conditions can be observed

as the reflection from left boundary is reduced compared to the reflection from top boundary.

The RBF-FD solution generally agrees with the FDM in scope of max pointwise error $e \approx 10^{-2}$. Absorbing boundary conditions also appear to be just as effective in use with RBF-FD as they are in original form with FDM.

As there is special interest in state of wave field directly at the surface, the time evolution of the top row of nodes is presented in Figure 4 – the seismogram. The difference between the real wave field and the suppressed reflection from the side boundaries is more clearly visible. Again, results of both methods agree visually.

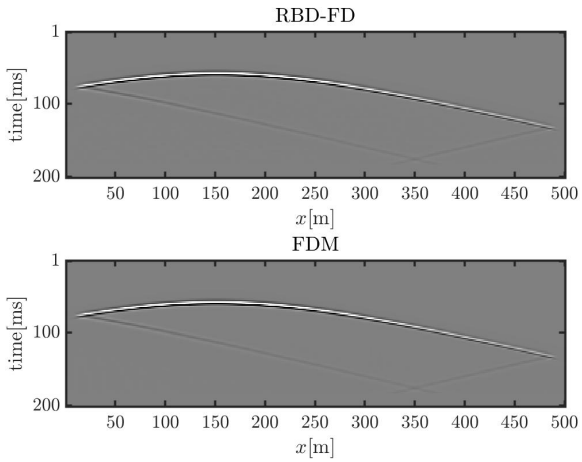

Figure 4. Comparison of seismograms computed using RBF-FD and FDM.

A) Computation time analysis

The most computationally demanding part of the code is the time loop where in each time step equation (12) is solved by performing spatial loop over all computational nodes. Due to the explicit treatment of time derivative, the iterations in spatial loop are completely independent, and can be therefore executed in parallel. The shared memory parallelization was implemented by *OpenMP omp parallel for* directive with static scheduling.

The execution time was measured using 1 to 8 threads as presented in Figure 5. The testing was done on a laptop computer with `Intel(R) Core(TM) i7-4750HQ CPU @ 2.00GHz` processor and 8 GB 1600 MHz DDR3 RAM. Code was compiled using `LLVM 5.0.1` for MacOS with `-std=c++11 -O3 -DNDEBUG` flags

Use of multiple threads had a positive impact on the calculation with speedups up to factor 3, as shown in Figure 6. One aspect holding back the effectiveness of the parallelization is the node placement process that takes place once at the beginning of the calculation and was not executed in parallel in this implementation. Parallelization of this aspect is left for future work.

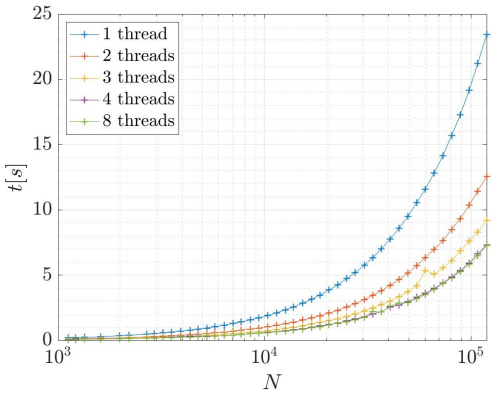

Figure 5. Time stepping – comparison of calculation time when using paralelization – log/lin scale.

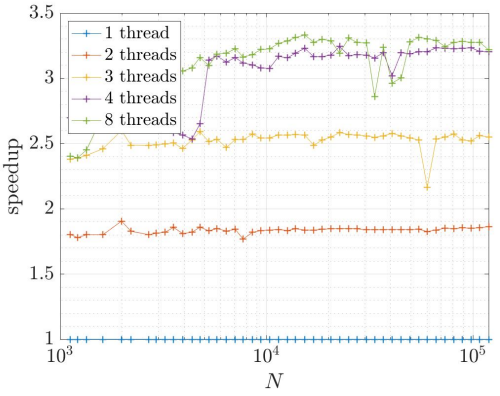

Figure 6. Time stepping – speedup when using paralelization – log/lin scale.

B. Velocity step

To demonstrate the advantages gained by using the more flexible RBF-FD method we present a second example, where the domain is divided in two regions and the boundary between them represents a step in wave velocity. The velocity field for the case is

$$v(z) = \begin{cases} 3000\,\mathrm{m/s}\,; & z < 250\,\mathrm{m} \\ 5000\,\mathrm{m/s}\,; & z \geq 250\,\mathrm{m} \end{cases} \tag{17}$$

The wavelength is linearly proportional to the wave velocity and for accurate simulation from 10 to 20 nodes per wavelength are required [10, 11]. This loosely translates to same requirement of 10 to 20 nodes per characteristic distance of the disturbance caused by the source. Ricker's wavelet was for purposes of this paper considered to be 3 characteristic distances long. In case of wave propagation in in-homogeneous media the computational nodes should therefore be distributed with density inversely proportional to the velocity field of the medium. As sudden jumps in node density tend to cause numerical errors, the node density follows the moving average of the velocity field instead of the field itself.

Again, Poisson Disk Sampling in conjunction with regularization algorithm was used for node placing. Fig-

ure 7 displays a part of the domain near where the gradual change in node density is observable.

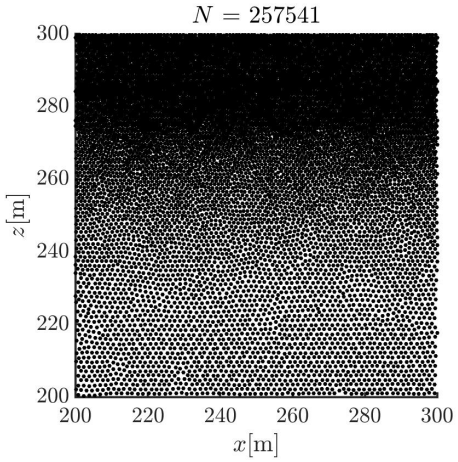

Figure 7. Node distribuition around the velocity step.

The size of the domain is kept at $500\,\mathrm{m} \times 500\,\mathrm{m}$. Roughly 250000 nodes are used with both methods. The time step is again the same with both methods, however it was reduced to $\Delta t = 0.00014142\,\mathrm{ms}$ compared to the first example to keep both methods stable. The RBF-FD setup remains the same as in the first example.

The source is located at coordinate $(250\,\mathrm{m},\ 200\,\mathrm{m})$, with $\sigma_R = 0.00106\,\mathrm{s}^{-1}$ and $\epsilon = 4\,\mathrm{m}$.

Without increasing the amount of computational nodes from the previous example, FDM with uniform node grid does not provide sufficient density everywhere inside the domain. In contrast, RBF-FD with variable node density provides 11.5 nodes per wavelength everywhere inside the domain, with roughly the same number of computational nodes.

Using the stability criterion for FDM [12, p. 205] we can compute the bound for Δt, at which the solution is still stable, which in our notation reads as

$$\Delta t \leq \sqrt{2}\frac{a}{v}. \tag{18}$$

From (18) we can observe that areas of high velocity cause instabilities when using FDM, if time step is not sufficiently reduced. However if the internodal distance a is itself proportional to velocity, the velocities in (18) cancel each other out. Explicitly writing the dependence of nodal density on the velocity field as $a = Cv$, where C s some constant, the stability criterion (18) simplifies to

$$\Delta t \leq \sqrt{2}C, \tag{19}$$

which is independent of v. In our implementation of RBF-FD, a is proportional to v and in turn problems with stability are avoided.

The results of simulation with RBF-FD method are presented in Figure 8, showing snapshots of the field at four different times. In addition to effects observed in the first numerical example, we can at time $t = 30\mathrm{ms}$ observe the internal reflection caused by the step in velocity. The superiority of RBF-FD solution can be best demonstrated by focusing on the area marked by a red rectangle. Figure 9 displays this area in greater detail, which illustrates the different behavior of RBF-FD and FDM.

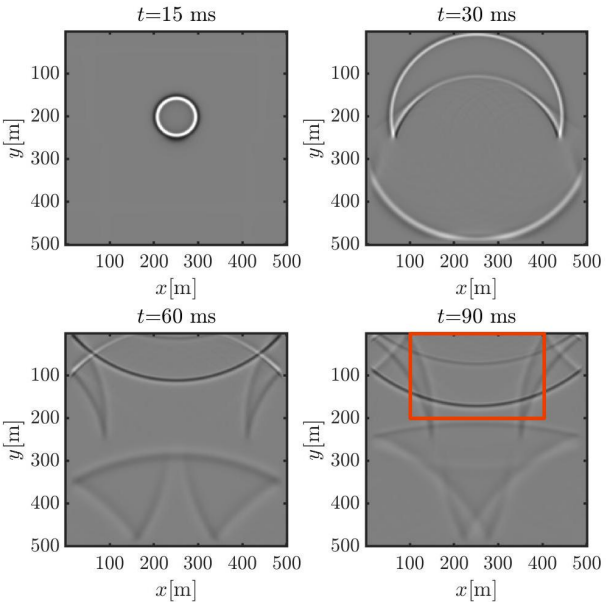

Figure 8. RBD-FD snapshots of wave field. Red rectangle depicts the area of interest.

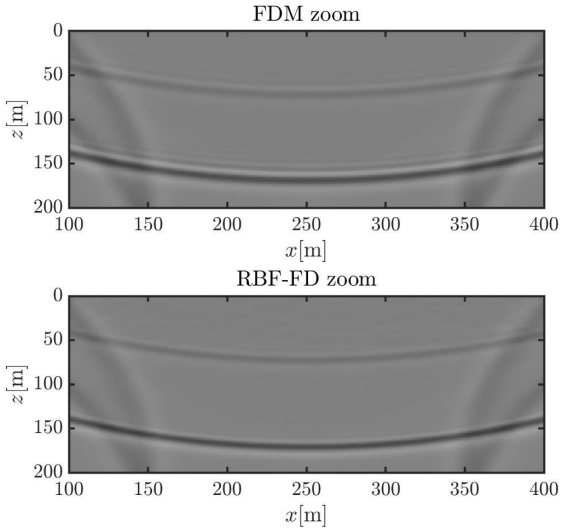

Figure 9. Comparison between RBF-FD and FDM solutions at time $t = 90$s on the part marked by red rectangle on Figure 8.

The numerical artifacts behind the main wavefront moving downwards on the FDM solution are easily visible, while the RBF-FD solution is artifact free. Artifacts that arise from variable nodal density are highly problematic in more complex use cases where they become difficult to distinguish from real reflections caused by changes in the medium.

However, the RBF-FD solution has some noise-like numerical artifacts. While much less prominent than the discussed artifacts in FDM solution, they are still undesirable. The cause is most likely the node density distribution at the discontinuity in the velocity field. One possible solution would be to slightly increase the node density in

the narrow region around the discontinuity. Unfortunately, this would require shorter time steps as a would not be proportional to v which would lead to instabilities.

V. Conclusions

This paper presents a RBF-FD solution of a time-domain acoustic wave propagation problem with absorbing boundary conditions, which was originally solved using FDM. The main advantage of the proposed RBF-FD solution is its flexibility regarding the positioning of computational nodes, which is exploited to solve the velocity step case with a refined nodal distribution. It is demonstrated that the refined solution is, as expected, superior to traditional FDM solution and that it can be relatively effectively executed in parallel on shared memory architecture.

Future research directions include further improvements to the node placement strategy, such as parallelization, and tackling more complex real world applications.

Acknowledgments

The authors would like to acknowledge the financial support of the ARRS research core funding No. P2-0095.

References

[1] C. Cerjan, D. Kosloff, R. Kosloff, and M. Reshef. A nonreflecting boundary condition for discrete acoustic and elastic wave equations. *Geophysics*, 50(4):705–708, 1985.

[2] P. Hahn and D. Negrut. On the use of meshless methods in acoustic simulations. In *ASME 2009 International Mechanical Engineering Congress and Exposition*, pages 185–199. American Society of Mechanical Engineers, 2009.

[3] Y. O. Zhang, T. Zhang, H. Ouyang, and T. Y. Li. Efficient SPH simulation of time-domain acoustic wave propagation. *Eng. Anal. Boundary Elem.*, 62:112–122, 2016.

[4] B. Fornberg. Generation of finite difference formulas on arbitrarily spaced grids. *Math. Comput.*, 51(184):699–706, 1988.

[5] B. Fornberg and N. Flyer. Solving PDEs with radial basis functions. *Acta Numerica*, 24:215–258, May 2015.

[6] V. Bayona, N. Flyer, B. Fornberg, and G. A. Barnett. On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs. *J. Comput. Phys.*, 332:257–273, 2017.

[7] J. Slak and G. Kosec. Refined meshless local strong form solution of Cauchy–Navier equation on an irregular domain. *Eng. Anal. Boundary Elem.*, 2018.

[8] J. Slak and G. Kosec. On generation of node distributions for meshless PDE discretizations. *arXiv:1812.03160 [math.NA]*, 2018. Preprint, available at `https://arxiv.org/abs/1812.03160`.

[9] J. Slak and G. Kosec. Parallel coordinate free implementation of local meshless method. In Karolj Skala, editor, *MIPRO 2018: 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, May 21–25, 2018, Opatija, Croatia*, MIPRO proceedings, pages 194–200. IEEE, Croatian Society for Information and Communication Technology, Electronics and Microelectronics, 2018.

[10] H. D. Geiger and P. F. Daley. Finite difference modelling of the full acoustic wave equation in Matlab. resreport 15, CREWES Research Report, 2003.

[11] R. M. Alford, K. R. Kelly, and D. M. Boore. Accuracy of finite-difference modeling of the acoustic wave equation. pages 834–842.

[12] A. R. Mitchell. *Computational methods in partial differential equations*. John Wiley & Sons., New York, 1969.