DOI: 10.61782/fa.2023.1079

forum acusticum 2023

FINITE-DIFFERENCE TIME-DOMAIN SIMULATION OF THE ACOUSTIC SCATTERING OF UNDERWATER OBJECTS ON ADAPTIVE SEMI-REGULAR DOMAINS

Urso Giorgio [1*] **Canepa Gaetano** [1] **Tesei Alessandra** [1] **Been Robert** [1]
[1] Centre for Maritime Research and Experimentation (CMRE)

ABSTRACT

Computing the acoustic scattering response of a large submerged object insonified with a medium-frequency signal presents several difficulties. Our calculation is based on a Finite Difference Time-Domain (FDTD) method accelerated with graphics processing units. In our simulation, the size of the target, given a certain frequency range, is limited by the memory of the computing hardware. To reduce the number of points at which the solution is computed, and thus the memory required for the simulation, we restricted (adapted) the computing domain to the geometry of the target and removed from the regular domain the subset of points with vacuum-like (air) physical properties. We implemented the incident field model and absorbing boundary conditions on this adaptive semi-regular domain. A comparison with the analytical solution for the spherical target case is presented to demonstrate the accuracy of the approximation. With the adopted solution, we expect to model a 60 meter vessel using a server equipped with four NVIDIA A100s.

Keywords: *3d, efit, gpu, fdtd, underwater*

1. INTRODUCTION

This work is part of research activities on modelling tools for acoustic target scattering at low to medium frequencies. This research, aims at generating realistic sets of simulated data for training of supervised classification schemes such as Neural Networks.

Synthetic data generation is obtained using a simulator based on the tool developed by Calvo *et al.* [1], a 3D finite difference time-domain model that updates the stress tensor and velocity fields of the acoustic medium using an Elastodynamic Finite Integration Technique.

We modified and extended this model to cope with the memory occupation problem related to the simulation of as large a target as possible, fixed the frequency range.

To get an idea of the problem, consider a cylindrical target L=60 m long, with a R=3 m radius and a shell d=4 cm thick. The original algorithm works on a regular grid of points. Using a spatial step of 2 mm for the computational grid, we obtain a minimum of:

$L \cdot R^2 / s^3 = 60 \cdot 3^2 / 0.002^3 = 67$ billions of points.

Since for each point the model requires 36 bytes, the memory needed for the acoustic field storing is approximately 2.2 TB.

This memory requirement is difficult to meet and we worked to reduce the number of points needed. First, we eliminated the points representing the interior of the target, which is mostly filled with air. Second, we used an adaptive domain to reduce the outer points representing the water in which the target is immersed where the field values influence affect the scattering simulation results to a less extent.

2. TARGET SCATTERING MODELING TOOL

The numerical technique used is a Finite Difference Time-Domain scheme applied in a three-dimensional space.

The domain is covered with a regular grid of cubic cells. In each cell, the stresses and velocities of the medium are defined at specific points: the center of the cubic cell for pressures (diagonal elements of the stress tensor, Txx, Tyy, Tzz), the center of the cube faces for the velocity

[*] ***Corresponding author***: *giorgio.urso@cmre.nato.int.*

components (vx, vy, vz), and the middle of the cubes edges for the shear stress (Txy, Tyz, Txz), as in Fig. 1.
The simulation consists of alternatively updating the stress tensor and the velocity vector using equations (1).

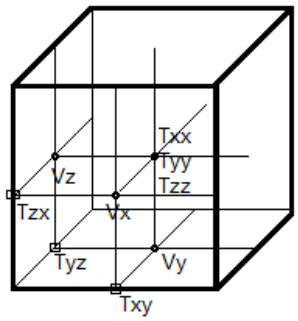

Figure 1. Computing stencil.

$$\rho\frac{\partial v_\alpha}{\partial t}=\frac{\partial T_{\alpha x}}{\partial x}+\frac{\partial T_{\alpha y}}{\partial y}+\frac{\partial T_{\alpha z}}{\partial z}$$
$$\frac{\partial T_{\alpha\alpha}}{\partial t}=\lambda\left(\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}\right)+2\mu\frac{\partial v_\alpha}{\partial a} \quad (1)$$
$$\frac{\partial T_{\alpha\beta}}{\partial t}=\mu\left(\frac{\partial v_\alpha}{\partial \beta}+\frac{\partial v_\beta}{\partial a}\right)$$
$$\lambda=\rho\left(C_p^2+C_s^2\right)\quad \mu=\rho C_s^2$$

Where T is the stress tensor, v is the velocity vector, ρ is the density of the medium, λ and μ are the Lamé elastic constants, C_p is the speed of compressional waves, C_s is the speed of shear waves, and α and β can be x, y, and z.
A cell can be occupied by a fluid or a solid, and both are treated in the same way; each cell has its own physical properties: ρ, λ and μ. For a fluid μ = 0.
Finite differences are computed as central differences using the staggered grid except for the domain boundaries described later.

As in [1], to determine the time step Δt of the explicit marching time algorithm and the geometric grid step Δx, two conditions are applied:
$\Delta t < 0.5 \cdot \Delta x / c_{max}$
$\Delta x > c_{min} / (20 * \upsilon_{max})$
Where c_{max} is the maximum and c_{min} the minimum sound speeds given by the physical properties of the medium, and the coefficient 20 is given by a heuristic rule.
For a steel made target c_{max}~6000 m/sec, in water c_{min}~1500 m/sec and, for our purposes, υ_{max}~15 kHz, giving:
Δx = 0.005 m
Δt = 4E-7 sec
A further condition can be applied on Δx being the requirement to have at least 20 points in the thickness of the target shell. This takes us to the 2 mm used in the study case of this work.

Since we have to absorb the scattered signal when it arrives at the boundary and at the same time let the signal enter the interior, we used a Higdon [3] absorbing boundary condition (ABC) applied to the difference between the boundary cell velocity and the incident signal.

The algorithm is implemented in CUDA C++ language and can use many GPUs on a single node. In the future, we will modify this algorithm to be able to run it on a multi-node cloud system.

3. FAR FIELD

To compute the scattering field at points far from the target, we used a Kirchhoff-Helmotz integral approach as in [2]:

$$p(r,f)=\frac{1}{4\pi}\int_{S_0}\left[p(r_0)\frac{\partial G(r,r_0)}{\partial n(r_0)}-\frac{\partial p(r_0)}{\partial n}G(r,r_0)\right]dS_0$$
$$\frac{\partial p(r_0,f)}{\partial n}=-j\omega\rho u(r_0)\cdot n$$
$$G(r,r_0)=\frac{e^{-jk(r-r_0)}}{\|r-r_0\|} \quad (2)$$
$$\frac{\partial G(r,r_0)}{\partial n(r_0)}=-\left(\frac{1}{\|r-r_0\|}+jk\right)G(r,r_0)\frac{(r-r_0)\cdot n}{\|r-r_0\|}$$
$$\omega=2\pi f\qquad k=\frac{\omega}{c}$$

Where the integral is extended to a closed surface S_0 around the target with infinitesimal element dS_0, $\boldsymbol{r_0}$ is the center of the element, $\boldsymbol{n}$ is the normal to the element, $p(\boldsymbol{r_0})$ and $\boldsymbol{u}(\boldsymbol{r_0})$ are pressure and velocity in $\boldsymbol{r_0}$.
The pressure p(**r**,f) is computed in **r** for a number of frequencies f in order to obtain a frequency / pressure plot.

4. EMPTY CELLS INSIDE THE TARGET

Many of the large targets we want to simulate are filled with air, which in our problem can be likened to vacuum with negligible errors. Acoustic waves do not propagate inside the target but only in the solid material, and it is not necessary to store physical quantities for these cells. Therefore, the first strategy to reduce the memory allocation was to eliminate this kind of cells.

We changed the data structure from a regular grid to a graph in which each cell had six pointers to neighboring cells. The pointers could point to NULL if the neighboring cell was filled with air. This first attempt was unsuccessful because 6 pointers required 6 x 4 = 24 extra bytes for each cell.

Therefore, we adopted a solution based on an irregular domain consisting of cubes of 8x8x8 cells, each cube having the 6 pointers to neighboring cubes. Therefore, the 24 bytes are only required by the cubes and not by the cells. The algorithm can find neighboring cells in the same cube or in connected cubes.

Considering the previous example scenario, we saved:

$L \cdot \pi(R-d)^2 / s^3 = 60 \cdot \pi (3-0.04)^2 / 0.002^3 =$

206 billions of cells that is 7.4 TB.

Of course, the savings depend on the air filled space inside the target, which may be minimal for some objects, but for vehicle containing humans it is, usually, considerable.

5. ADAPTIVE DOMAIN

The second strategy to reduce memory allocation was to eliminate as much fluid as possible outside the target.

Since the domain has a curved boundary, to simulate a plane wave orthogonal to the z dimension, we add a v_z component to the cells on the boundary considering the time it takes to travel from a virtual position outside the boundary to the boundary itself.

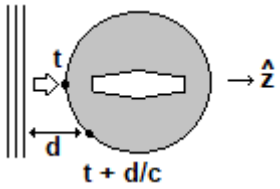

Figure 2. Scenario in which a plane wave approaches a spherical domain (in gray).

The source can be expressed as:

$S_z = S(t - d/c)$

Where *S* is the pulse shape of the source signal in time, in our case a Ricker wavelet, *t* is the time at which the source is computed, *c* is the speed of sound in the medium outside the target (water in our case) and *d* is the distance of the boundary point from the location of the source (outside the computational domain).

We tested the effectiveness of boundary absorption using a For this test, the radius of the outer surface was 15 cm and the shell thickness was 1 cm. We used a spatial step of 2 mm that, considering a Courant–Friedrichs–Lewy stability condition of 0.5, gives a time step of 16.75 µsec and 20 points inside the metallic shell. We simulated 0.5 sec of the scattering process.

Geometry is shown in Fig. 4 and comparison between simulated and analytical signals in Fig. 5.

spherical Ricker wavelet in an empty spherical domain. The source was in the center of the domain and the receiver on a diagonal. With first-order Higdon's ABC there was strong reflection from the boundary, but with second-order it was almost completely canceled (Fig. 3).

Fluid cells clipping dramatically reduces the amount of memory required, although a water shell around the target is still needed to implement the far-field calculation algorithm. Assuming we keep 10 fluid cells outside the cylinder in our scenario, the required memory will be:

$L \cdot \pi [(R+10 \cdot s)^2 - (R-d)^2] / s^3 =$

$60 \cdot \pi [(3 + 10 \cdot 0.002) - (3 - 0.04)]^2 / 0.002^3 =$

8.4 billions cells that is 302 GB.

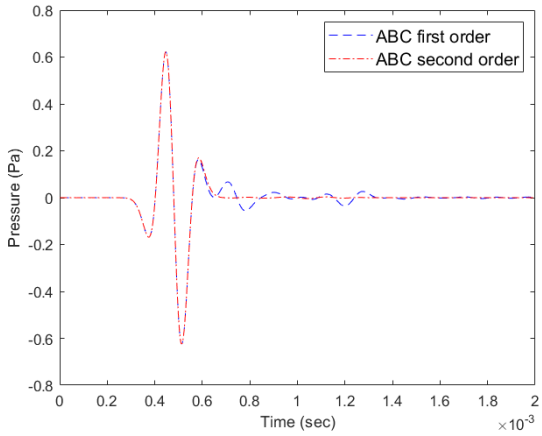

Figure 3. Boundary absorption test.

Now the simulation can run on our computer based on four NVidia A100 GPUs since each of them has 80 GB of RAM.

6. TEST WITH SPHERICAL SHELL

To test the algorithm we model a spherical steel shell, hollow inside and immersed in water, for which the analytical solution can be calculated.

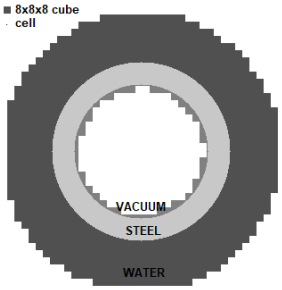

Figure 4. Spherical target within a spherical domain

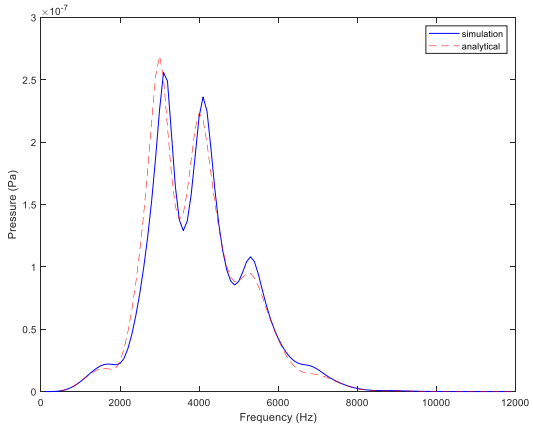

Figure 5. Backward scattering far-field

7. REALISTIC CASE

Since we are interested in the classification of targets on features as the shape and the internal structure, we considered a representative autonomous underwater vehicle in two versions: a simple one (Fig. 6) and a ribbed one (Fig. 7). The diameter of the AUV is 53 cm, the length 3 m. We report the backscattered signal from both targets insonified with a Ricker wavelet having a peek frequency of 4 kHz (Fig. 8). We used a space step of 2 mm and a time step of 13.4 µsec.

The different scattering is clearly visible.

8. CONCLUSIONS

A tool for modelling underwater acoustics scattering is under development at NATO STO-CMRE. After the first tests to verify the correctness of the FDTD - EFIT solution, we faced the memory allocation problem for large target objects. The solution based on eliminating as many computing points as possible was outlined in this paper, demonstrating the feasibility of the approach.

After the optimization of the software, we are going to test it on realistic target models with approximate dimensions of 3 x 60 meters (as used in our computational analysis).

Figure 6. Empty AUV section.

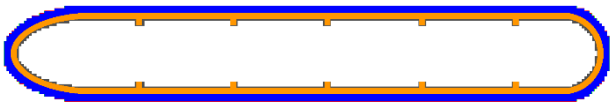

Figure 7. Ribbed AUV section.

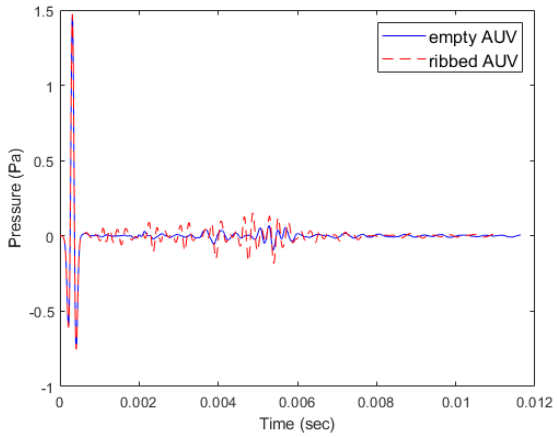

Figure 8. Far-field backscattering, input signal 4 kHz.

9. REFERENCES

[1] Calvo, D.C., Rudd, K.E., Zampolli, M., Sanders, W.M., and Bibee, L.D.: "Simulation of acoustic scattering from an aluminum cylinder near a rough interface using the elastodynamic finite integration technique", *Wave Motion 47*, pp. 616–634, 2010.

[2] Mingsian R. Bai, Jeong-Guon Ih, Jacob Benesty: "Appendix: Acoustic Boundary Element Method" *Acoustic Array Systems: Theory, Implementation, and Application*, IEEE, pp.501-511, 2013.

[3] Higdon R. L.: "Absorbing boundary conditions for elastic waves" *Geophysics*, 56(2): 231, 1991.