Virtual Testbed: Simulation of Ocean Wave Reflection from the Ship Hull

Ivan Petriakov, Alexander Degtyarev, Denis Egorov, Ivan Gankevich(✉), Anton Gavrikov, Artemii Grigorev, and Vasily Khramushin

Saint Petersburg State University, 7-9 Universitetskaya Emb., St. Petersburg 199034, Russia
{st049350,st047824,st047437,st016177}@student.spbu.ru, {a.degtyarev,i.gankevich,v.khramushin}@spbu.ru
https://spbu.ru/

Abstract. Diffraction and radiation forces result from the interaction between the ship hull and the moving fluid. These forces are typically simulated using added masses, a method that uses mass to compensate for not computing these forces directly. In this paper we propose simple mathematical model to compute diffraction force. The model is based on Lagrangian description of the flow and uses law of reflection to include diffraction term in the solution. The solution satisfies continuity equation and equation of motion, but is restricted to the boundary of the ship hull. The solution was implemented in velocity potential solver in Virtual testbed—a programme for workstations that simulates ship motions in extreme conditions. Performance benchmarks of the solver showed that it is particularly efficient on graphical accelerators.

Keywords: Ocean wave diffraction · Ocean wave radiation · Fluid velocity field · Law of reflection · OpenCL · OpenMP · GPGPU

1 Introduction

There are two mathematical models that describe rigid body motion and fluid particle motion: equations of translational and angular motion of the rigid body (Newton's Second Law) and Gerstner equations for ocean waves (which are solutions to linearised equations of motions for fluid particles). Usually, we use these models independently to generate incident ocean waves and then compute body motions caused by these waves. To measure the effect of still fluid on an oscillating rigid body (radiation forces) and the effect of fluid particles hitting the body (diffraction forces), we use added masses and damping coefficients—simplified formulae derived for small-amplitude oscillatory motion.

Supported by Saint Petersburg State University (grants no. 51129371 and 51129725) and Council for grants of the President of the Russian Federation (grant no. MK-383.2020.9).

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12254, pp. 29–39, 2020.
https://doi.org/10.1007/978-3-030-58817-5_3

But, what if we want to simulate large-amplitude rigid body motion with greater accuracy? There are two possible ways. First, we may use numerical methods such as Reynolds-Averaged Navier Stokes (RANS) method [5]. This method is accurate, can be used for viscous fluid, but not the most computationally efficient. Second, we may solve Gerstner equations with appropriate boundary condition and use the solution to compute both rigid body and fluid particle motion around it. This paper explores this second option. Similar approach was followed by Fenton in [1–3], but the distinctive feature of our approach is the use of law of reflection to derive analytic expressions for reflected waves and fluid particles.

2 Methods

2.1 Equations of Motions with a Moving Surface Boundary

An oscillating rigid body that floats in the water and experiences incident waves both reflects existing waves and generates new waves:

- fluid particles hit the body, causing it to move, and then reflect from it;
- moving body hits fluid particles and makes them move.

Both wave reflection and generation have the same nature—they are caused by the collision of the particles and the body—hence we describe them by the same set of formulae. Hereinafter we borrow the mathematical notation for Lagrangian description of the flow from [4].

In Lagrangian description of flow instantaneous particle coordinates $\boldsymbol{R} = (x, y, z)$ depend on particle positions at rest (independent initial coordinates) $\boldsymbol{\zeta} = (\alpha, \beta, \delta)$ and time t, i.e. $\boldsymbol{R} = \boldsymbol{R}(\alpha, \beta, \delta, t)$. Using this notation equation of motion (conservation of momentum) is written as

$$\boldsymbol{R}_{tt} + g\hat{z} + \frac{1}{\rho}\nabla_{\boldsymbol{R}} p = 0, \tag{1}$$

where $\boldsymbol{R}$—particle coordinates, $\hat{z}$—unit vector in the direction of positive z, p—pressure, ρ—fluid density, g—gravitational acceleration. Continuity equation (conservation of mass) is written as

$$|\mathbb{J}| = 1, \qquad \frac{\partial}{\partial t}|\mathbb{J}| = 0, \qquad \mathbb{J} = \begin{bmatrix} x_\alpha & y_\alpha & z_\alpha \\ x_\beta & y_\beta & z_\beta \\ x_\delta & y_\delta & z_\delta \end{bmatrix}.$$

Multiplying both sides of (1) by $\mathbb{J}$ and noting that $\nabla_{\boldsymbol{\zeta}} p = \mathbb{J}\nabla_{\boldsymbol{R}} p$ gives

$$\mathbb{J}\boldsymbol{R}_{tt} + g\boldsymbol{\nabla}(\boldsymbol{R} \cdot \hat{z}) + \frac{1}{\rho}\boldsymbol{\nabla} p$$

Following [4] we seek solution to this equation in the form of a simultaneous perturbation expansion for position, pressure, and the vorticity function:

$$\boldsymbol{R} = \boldsymbol{R}_0 + \boldsymbol{R}_1 + \boldsymbol{R}_2 + \ldots$$
$$p = p_a - \rho g\delta + p_1 + p_2 + \ldots$$

Zeroth order terms are related to particles positions at rest:

$$\boldsymbol{R}_0 = \boldsymbol{\zeta}$$
$$p_0 = p_a - \rho g \delta$$

First-order terms are solutions to linearised equation of motion and equation of continuity:

$$\boldsymbol{R}_{1tt} + g\boldsymbol{\nabla}\left(\boldsymbol{R}_1 \cdot \hat{z}\right) + \frac{1}{\rho}\boldsymbol{\nabla} p_1 = 0$$
$$\boldsymbol{\nabla} \cdot \boldsymbol{R}_1 = 0$$

We seek solutions of the form $\boldsymbol{R}_1 = \nabla w$ to make the flow irrotational. Plugging this form into the equations gives

$$\begin{aligned} &\boldsymbol{\nabla}\left(w_{tt} + g w_\delta + p_1/\rho\right) = 0 \\ &\Delta w = 0 \end{aligned} \tag{2}$$

The first equation denotes conservation of momentum (Newton's second law, equation of motion) and the second equation denotes conservation of mass (equation of continuity).

When we have no boundary condition we seek solutions of the form

$$w\left(\alpha, \beta, \delta\right) = \mathrm{Re}\, f(u, v) \exp\left(iu\alpha + iv\beta + k\delta - i\omega t\right), \tag{3}$$

where u and v are wave numbers. We plug (3) into continuity equation (2) where p_1 is constant and get $k = \sqrt{u^2 + v^2}$. That means that expression (3) is the solution to this equation when k is wave vector magnitude, i.e. w decays exponentially with increasing water depth multiplied by wave vector magnitude.

Then we plug (3) into equation of motion (2) and get $\omega^2 = gk$, which is dispersion relation from classic linear wave theory. That means that the expression is the solution to this equation when angular frequency depends on the wave number, i.e. waves of different lengths have different phase velocities.

Before solving this system of equations for an arbitrary moving surface boundary, we consider particular cases to substantiate the choice of the form of the solution.

We use $\mathrm{Re}\exp\left(iu\alpha + iv\beta + k\delta - i\omega t\right)$ to describe fluid particle potential. Here u and v are wave numbers, ω is angular frequency, and k is wave vector. This notation makes formulae short and is equivalent to the description that uses traditional harmonic functions. This notation allows for easy transition to irregular waves via Fourier transforms which are essential for fast computations. Such solutions will be studied in future work.

2.2 Stationary Surface Boundary

In this section we explore solutions stationary surface boundary in a form of infinite plane surface. On such a boundary the projection of particle velocity to the surface normal is nought. We write boundary conditions and corresponding solutions for different orientations of this boundary and then generalise these solutions to a parametric surface.

Infinite Wall. On a vertical surface the boundary condition is written as

$$\frac{d}{dt}\boldsymbol{\nabla} w \cdot \boldsymbol{n} = \frac{d}{dt}\frac{\partial}{\partial \alpha} w = 0; \qquad \alpha = \alpha_0; \qquad \boldsymbol{n} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Here we consider only α coordinate, the derivations for β are similar. The potential of incident fluid particle has the form

$$w\left(\alpha, \beta, \delta, t\right) = \exp\left(k\delta - i\omega t\right) \exp\left(iu\alpha + iv\beta\right).$$

Velocity vector of this particle is

$$\frac{d}{dt}\boldsymbol{\nabla} w = i\omega\left(\boldsymbol{d}_k + i\boldsymbol{d}_i\right) \exp\left(k\delta - i\omega t\right) \exp\left(iu\alpha + iv\beta\right);$$
$$\boldsymbol{d}_k = \begin{bmatrix} 0 \\ 0 \\ k \end{bmatrix}; \qquad \boldsymbol{d}_i = \begin{bmatrix} u \\ v \\ 0 \end{bmatrix}.$$

where $\boldsymbol{d}_i$ is horizontal incident wave direction and $\boldsymbol{d}_k$ is a vector that contains amplitude damping coefficient. (We use the vector instead of the scalar to shorten mathematical notation, otherwise we would have write a separate formula for vertical coordinate.) The law of reflection states that the angle of incidence equals the angle of reflection. Then the direction of reflected wave is[1]

$$\boldsymbol{d}_r = \boldsymbol{d}_i - \boldsymbol{d}_s = \boldsymbol{d}_i - 2\boldsymbol{n}\left(\boldsymbol{d}_i \cdot \boldsymbol{n}\right) = \begin{bmatrix} -u \\ v \\ 0 \end{bmatrix}.$$

We seek solutions of the form

$$w\left(\alpha, \beta, \delta, t\right) = \left[C_1 \exp\left(iu\alpha\right) + C_2 \exp\left(-iu\alpha\right)\right] \exp\left(k\delta - i\omega t\right) \exp\left(iv\beta\right). \tag{4}$$

We plug this expression into the boundary condition and get

$$C_1 \exp\left(iu\alpha_0\right) - C_2 \exp\left(-iu\alpha_0\right) = 0,$$

hence $C_1 = C\exp(-iu\alpha_0)$ and $C_2 = -C\exp(iu\alpha_0)$. Constant C may take arbitrary values, here we set it to 1. Plugging C_1 and C_2 into (4) gives the final solution

$$w\left(\alpha, \beta, \delta, t\right) = \cosh\left(iu\left(\alpha_0 - \alpha\right)\right) \exp\left(k\delta - i\omega t\right) \exp\left(iv\beta\right).$$

There are two exponents in this solution with the opposite signs before horizontal coordinate α. These exponents denote incident and reflected wave respectively. The amplitude of the reflected wave does not decay as we go farther from the boundary, but decay only when we go deeper in the ocean. This behaviour corresponds to the real-world ocean waves.

[1] Initially, we included the third component of incident wave direction making the vector complex-valued, however, the solution blew up as a result of mixing real and imaginary parts in dot products involving complex-valued vectors. The problem was solved by reflecting in two dimensions which is intuitive for ocean waves, but not for particles.

Infinite Plate. On a horizontal surface the boundary condition is written as

$$\frac{d}{dt}\boldsymbol{\nabla} w \cdot \boldsymbol{n} = \frac{d}{dt}\frac{\partial}{\partial \delta} w = 0; \qquad \delta = \delta_0; \qquad \boldsymbol{n} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Analogously to wave direction we write vector form of the incident particle trajectory radius damping coefficient as $(0, 0, k)$, hence vector form of the reflected coefficient is $(0, 0, -k)$. We seek solutions of the form

$$w(\alpha, \beta, \delta, t) = [C_1 \exp(k\delta) + C_2 \exp(-k\delta)] \exp(-i\omega t) \exp(iu\alpha + iv\beta). \quad (5)$$

We plug this expression into the boundary condition and get

$$C_1 \exp(k\delta_0) - C_2 \exp(-k\delta_0) = 0.$$

Hence $C_1 = C \exp(-k\delta_0)$ and $C_2 = C \exp(k\delta_0)$. Constant C may take arbitrary values, here we set it to $1/2$. Plugging C_1 and C_2 into (5) gives the final solution

$$w(\alpha, \beta, \delta, t) = \cosh(k(\delta - \delta_0)) \exp(-i\omega t) \exp(iu\alpha + iv\beta).$$

There are two exponents in this solution with opposite signs before vertical coordinate δ. These exponents make the radius of the particle trajectories decay exponentially while approaching the boundary δ_0 (i.e. with increasing water depth). This is known solution from linear wave theory.

Infinite Panel. On an arbitrary aligned infinite surface the boundary condition is written as

$$\frac{d}{dt}\boldsymbol{\nabla} w \cdot \boldsymbol{n} = 0; \qquad \boldsymbol{n} \cdot (\boldsymbol{\zeta} - \boldsymbol{\zeta}_0) = 0,$$

where $\boldsymbol{\zeta_0}$ is the point on the boundary plane and the third component of the normal vector is nought: $\boldsymbol{n} = (n_1, n_2, 0)$, $|\boldsymbol{n}| = 1$. The direction of incident wave is $\boldsymbol{d}_i = (u, v, 0)$ and the direction of reflected wave is $\boldsymbol{d}_r$. We seek solutions of the form

$$\begin{aligned} w(\alpha, \beta, \delta, t) = {} & C_1 \exp((i\boldsymbol{d}_i + \boldsymbol{d}_k) \cdot \boldsymbol{\zeta} - i\omega_1 t) \\ & + C_2 \exp((i\boldsymbol{d}_r + \boldsymbol{d}_k) \cdot \boldsymbol{\zeta} - i\omega_2 t). \end{aligned} \quad (6)$$

We plug this expression into the boundary condition and get

$$(i\boldsymbol{d_i} \cdot \boldsymbol{n}) C_1 + (i\boldsymbol{d_r} \cdot \boldsymbol{n}) C_2 \exp(-i\boldsymbol{d}_s \cdot \boldsymbol{\zeta}_0) = 0.$$

Here we substitute $\boldsymbol{d}_r \cdot \boldsymbol{n}$ with $-\boldsymbol{d}_i \cdot \boldsymbol{n}$ which is derived from the formula for $\boldsymbol{d}_r$. Hence, the boundary condition reduces to

$$C_1 - C_2 \exp(-i\boldsymbol{d}_s \cdot \boldsymbol{\zeta}_0) = 0.$$

Hence $C_1 = \frac{1}{2}\exp\left(-\frac{1}{2}i\boldsymbol{d}_s \cdot \boldsymbol{\zeta}_0\right)$ and $C_2 = \frac{1}{2}\exp\left(\frac{1}{2}i\boldsymbol{d}_s \cdot \boldsymbol{\zeta}_0\right)$. This solution reduces to the solution for the wall when $\boldsymbol{n} = (0, 0, 1)$.

In a computer programme it is more practical to set $C_1 = 1$ and $C_2 = \exp\left(i\boldsymbol{d}_s \cdot \zeta_0\right)$: that way you have to integrate only the second term in the solution over all ship hull panels (see Sect. 3.1).

2.3 OpenCL Implementation

Solution for fluid velocity field was implemented in velocity potential solver in the framework of Virtual testbed. Virtual testbed is a programme for workstations that simulates ship motions in extreme conditions and physical phenomena that causes them: ocean waves, wind, compartment flooding etc. The main feature of this programme is to perform all calculations nearly in real time, paying attention to the high accuracy of calculations, which is partially achieved using graphical accelerators.

Virtual testbed uses several solvers to simulate ship motions. The algorithm for velocity potential solver is the following.

- First of all, we generate wavy surface, according to our solution and using wetted ship panels from the previous time step (if any).
- Second, we compute wetted panels for the current time step, which are located under the surface calculated on the previous step.
- Finally, we calculate Froude—Krylov forces, acting on a ship hull.

These steps are repeated in infinite loop. Consequently, wavy surface is always one time step behind the wetted panels. This inconsistency is a result of the decision not to solve ship motions and fluid motions in one system of equations, which would be too difficult to do.

Let us consider process of computing wavy surface in more detail. Since wavy surface grid is irregular (i.e. we store a matrix of fluid particle positions that describe the surface), we compute the same formula for each point of the surface. It is easy to do with C++ for CPU computation, but it takes some effort to efficiently run this algorithm with GPU acceleration. Our first naive implementation was inefficient, but the second implementation that used local memory to optimise memory loads and stores proved to be much more performant.

First, we optimised storage order of points making it fully sequential. Sequential storage order leads to sequential loads and stores from the global memory and greatly improves performance of the graphical accelerator. Second, we use as many built-in vector functions as we can in our computations, since they are much more efficient than manually written ones and compiler knows how to optimise them. This also decreases code size and prevents possible mistakes in the manual implementation. Finally, we optimised how ship hull panels are read from the global memory. One way to think about panels is that they are coefficients in our model, as array of coefficients is typically read-only and constant.

This type of array is best placed in the constant memory of the graphical accelerator that provides L2 cache for faster loads by parallel threads. However, our panel array is too large to fit in constant memory, so we simulated constant memory using local memory: we copied a small block of the array into local memory of the multiprocessor, computed sum using this block and then proceeded to the next block. This approach allowed to achieve almost 200-fold speedup over CPU version of the solver.

A distinctive feature of the local memory is that it has the smallest latency, at the same time sharing its contents between all computing units of the multiprocessor. Using local memory we reduce the number of load/store operations to global memory, which has larger latency. As far as global memory bandwidth remains a bottleneck, this kind of optimisation would improve performance. To summarise, our approach to write code for graphical accelerators is the following:

- make storage order linear,
- use as many built-in vector operations as is possible,
- use local memory of the multiprocessor to optimise global memory load and stores.

Following these simple rules, we can easily implement efficient algorithms.

3 Results

3.1 Diffraction

In the first experiment we use solution for infinite panel to simulate wave diffraction around Aurora's ship hull (see Fig. 1). In order to apply (6) to this problem we use smoothing kernel that accumulates influence of every panel on a particular point of ocean surface:

$$w = \sum_j K_j w_j; \qquad K_j = \frac{1}{1 + |\zeta - \zeta_0|^2}$$

Here w_j is solution (6) written for panel j, K_j is smoothing kernel, ζ_0 is the centre of the panel. In the centre of the panel $K_j = 0$ and far from the ship hull $K_j \to 0$.

In the experiment waves with amplitude 1 approach the ship from the aft. The results of the experiment are shown in Fig. 2. Near the aft waves change their direction to be tangent to the waterline curve, follow the curve to the bow, and then restore their original direction. The amplitude of waves near the hull is also increased.

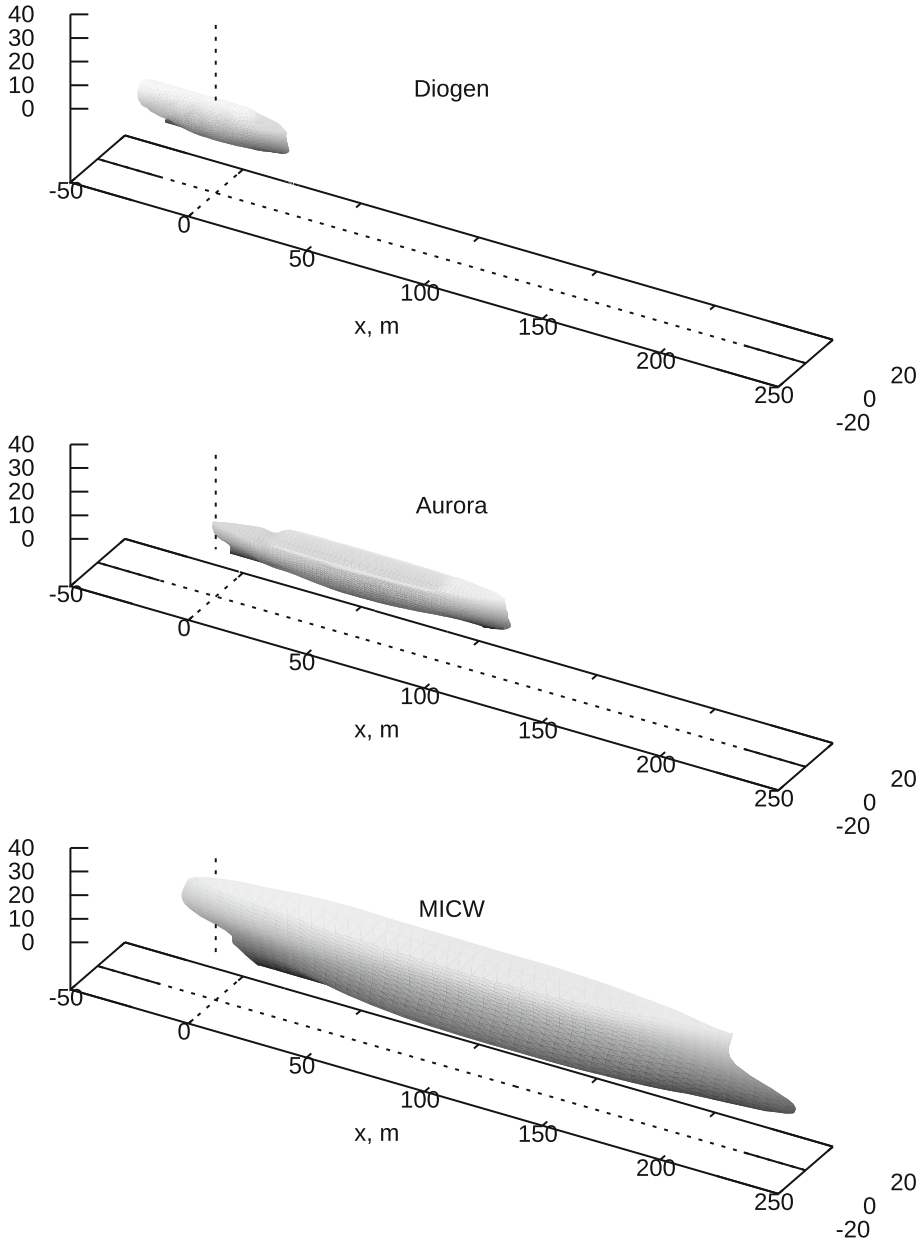

Fig. 1. Diogen, Aurora and MICW three-dimensional ship hull models.

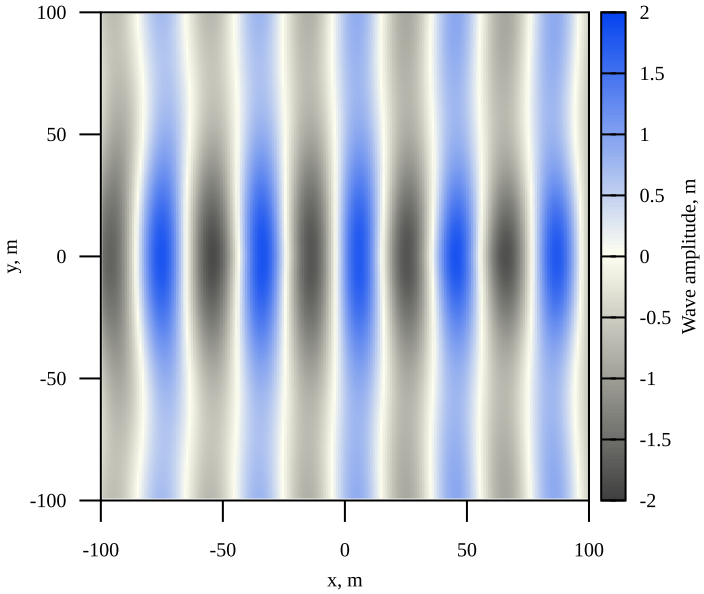

Fig. 2. Wave diffraction around Aurora's hull (the hull is not shown). Top view.

3.2 Performance Benchmarks

We implemented velocity potential solver using OpenMP for parallel computations on a processor and OpenCL for graphical accelerator. The solver uses single precision floating point numbers. Benchmark results are presented in Table 2.

We performed benchmarks for three ships: Diogen, Aurora and MICW. Diogen is a small-size fishing vessel, Aurora is mid-size cruiser and MICW is a large-size ship with small moment of inertia for the current waterline (Fig. 1). The main difference between the ships that affects benchmarks is the number of panels into which the hull is decomposed. These numbers are shown in Table 1.

Table 1. Parameters of ship hulls that were used in the benchmarks.

	Diogen	Aurora	MICW
Length, m	60	126.5	260
Beam, m	15	16.8	32
Depth, m	15	14.5	31
No. of panels	4346	6335	9252

Benchmarks were performed using three workstations: DarkwingDuck, GPUlab, Capybara. DarkwingDuck is a laptop, GPUlab is a desktop workstation, and Capybara is a desktop with professional graphical accelerator server-grade processor (Table 3).

Table 2. Performance benchmarks results. Numbers represent average time in milliseconds that is needed to generate waves with reflection.

Node	Diogen		Aurora		MICW	
	MP	CL	MP	CL	MP	CL
DarkwingDuck	5462	48	7716	41	7725	11
GPUlab	5529	11	8222	10	6481	3
Capybara	2908	16	2091	8	2786	4

Table 3. Hardware configurations for benchmarks. For all benchmarks we used GCC version 9.1.0 compiler and optimisation flags `-O3 -march=native`.

Node	CPU	GPU	GPU GFLOPS	
			Single	Double
DarkwingDuck	Intel i7-3630QM	NVIDIA GT740M	622	
GPUlab	AMD FX-8370	NVIDIA GTX1060	4375	137
Capybara	Intel E5-2630 v4	NVIDIA P5000	8873	277

4 Discussion

All the solutions obtained for various boundaries in this papre satisfy continuity equation and equation of motion, but they are all written for *plane* surface boundaries with different orientations. Typical ship hull three-dimensional model is represented by triangulated surface, and in the centre of each triangular panel fluid particle velocity vector does not depend on the surface normal of the other panels. So, the solution for plane surface boundary is enough to compute fluid velocity field directly *on* the surface boundary.

In order to generalise the solution for fluid velocity field *near* the surface boundary, we need to calculate weighted average of reflection terms of each underwater panel of the surface. Using inverse squared distance as the weight gives acceptable results in our experiments, but may not be appropriate in others.

It is not clear how much we have to increase wave amplitude near the boundary. One way to control the amplitude increase is to introduce the coefficient C into the reflection formula $\boldsymbol{d}_r = \boldsymbol{d}_i - C\boldsymbol{d}_s$ and before reflection term in (6). When $C = 1$ the wave is fully reflected from the boundary and the amplitude is doubled, when $C = 0$ no reflection occurres and the amplitude does not change.

Performance benchmarks showed that graphical accelerator greatly improves performance of velocity potential solver. Linear memory access patterns and large amount of floating point operations make this solver an ideal candidate for running on a graphical accelerator, and these features are the result of deriving explicit solution for fluid motions near the ship hull boundary.

5 Conclusion

This paper proposes a new model for ocean wave diffraction near ship hull. This model uses law of reflection to simulate incident and reflected waves and fluid particle motion. Although, the solutions are written for infinite plane surfaces, they can be used to compute fluid velocity at the centre of each triangle of the triangulated ship hull surface and can be generalised to compute fluid velocity near the ship hull using weighted sum over all panels.

The model was implemented in Virtual testbed velocity potential solver and was found to be highly efficient on a graphical accelerator. Future work is to incorporate radiation into the model and compare the solution to existing empirical approaches.

Acknowledgements. Research work is supported by Saint Petersburg State University (grants no. 51129371 and 51129725) and Council for grants of the President of the Russian Federation (grant no. MK-383.2020.9).

References

1. Fenton, J.D.: Wave forces on vertical bodies of revolution. J. Fluid Mech. **85**(2), 241–255 (1978). https://doi.org/10.1017/S0022112078000622
2. Fenton, J.D.: A spectral method for diffraction problems. In: Proceedings of International Symposium on Waves - Physical and Numerical Modelling, pp. 961–970. The University of British Columbia, Vancouver (1994)
3. Fenton, J.D., et al.: Simulating wave shoaling with boundary integral equations. In: 11th Australasian Conference on Coastal and Ocean Engineering: Coastal Engineering a Partnership with Nature; Preprints of Papers, p. 71. Institution of Engineers, Australia (1993)
4. Nouguier, F., Chapron, B., Guérin, C.A.: Second-order Lagrangian description of tridimensional gravity wave interactions. J. Fluid Mech. **772**, 165–196 (2015). https://doi.org/10.1017/jfm.2015.179
5. Wackers, J., et al.: Free-surface viscous flow solution methods for ship hydrodynamics. Arch. Comput. Meth. Eng. **18**(1), 1–41 (2011). https://doi.org/10.1007/s11831-011-9059-4