The 29th International Workshop on Water Waves and Floating Bodies
IWWWFB 2014
March 30 - April 2, 2014, Osaka, Japan

Towards Real Time Simulation of Ship-Ship Interaction - Part III: Immersed Body Boundary Condition and Double Body Ship-Ship Interaction*

Ole Lindberg [†1], Harry B. Bingham[2], and Allan P. Engsig-Karup[3]

[1] *FORCE Technology, Department of Simulation, Training and Ports, DK-2800 Kgs. Lyngby, Denmark*
[2] *Technical University of Denmark, Department of Mechanical Engineering, DK-2800 Kgs. Lyngby, Denmark*
[3] *Technical University of Denmark, Department of Applied Mathematics and Computer Science, DK-2800 Kgs. Lyngby, Denmark*

This paper considers real time calculation of ship to ship interaction forces in a maritime simulator. Maritime simulators are used for marine engineering and training of naval officers. In the simulator, it is possible to navigate realistic ship models in realistic marine environments. The more accurate the physical modeling of the environment, the higher the quality of the training and engineering. Motion of ships are determined by the forces on the ships from the environment. The forces on ships comes from interaction with other ships, fixed or floating structures, the seabed, wind, waves, viscous friction etc.

The main challenge is that all forces have to be calculated in real time, and that the calculations of the solutions needs to be fast and scalable. This paper considers calculation of ship-ship interaction forces based on an existing potential flow finite difference (FD) model for large-scale ocean wave modeling [2]; extended to include a new weighted least squares (WLS) approximation body boundary condition (BC) on the ship's hull. The model is solved using an efficient multigrid preconditioned defect correction solver (PDC) [1] and implemented in parallel on high-througput many-core graphical processing units (GPU) using the CUDA API [3], [4]. The new WLS approximation of the body BC is similar to immersed boundary methods (IBM) for elliptic equations [9], in particular the inhomogeneous Neumann IBM. This development is a result of an ongoing effort to develop more accurate, efficient and flexible models of wave forces on ships, and earlier developments have been presented at previous workshops and conferences [6], [8], [7]. This ship-ship interaction model is developed in cooperation between the Technical University of Denmark and FORCE Technology and is currently being integrated into the maritime simulator SimFlex4 at FORCE Technology.

The ship to ship interaction time depends on the speed difference between ships and the ship sizes $T = L/\Delta U$. If two ships meet head on, the interaction time is short and if a ship is overtaking another the interaction time is longer. Simple calculations suggest that the force calculation frequency should be in the range 0.1 to 10 Hz. Analysis relevant for determining feasibility of real time calculcations is given in [3], [1].

Double Body Flow Model

The double body flow model is based on potential flow theory, *i.e.* the water is assumed to be incompressible, inviscid and irrotational, and the fluid velocity is calculated from the scalar velocity potential by $\boldsymbol{u} = \nabla\phi$. The velocity potential satisfies mass conservation in the fluid volume Ω through the Laplace

*We wish to thank Maritime Development Center of Europe, FORCE Technology, DTU Mechanical Engineering and DTU Compute for supporting this project.
†Email: old@force.dk

equation and the kinematic BC's on the flat free surface, the sea bed and the on bodies in the fluid

$$\begin{aligned} \nabla^2\phi &= 0, \quad z \in [-h, 0], && (1)\\ \frac{\partial\phi}{\partial z} &= 0, \quad z = 0, && (2)\\ \boldsymbol{n}\cdot\nabla\phi &= 0, \quad z = -h, && (3)\\ \boldsymbol{n}\cdot\nabla\phi &= \boldsymbol{n}\cdot\boldsymbol{u}_B, \quad \boldsymbol{x} \in S_B, && (4) \end{aligned}$$

where h is the sea depth, the still water level is at $z = 0$, $\boldsymbol{n} = (n_x, n_y, n_z)^T$ is the outward normal vector to the fluid boundary, $\boldsymbol{u}_B$ is the body velocity and S_B is the body surface. The forces and moments on the bodies are calculated from integration of the dynamic pressure over the body surfaces. The fluid domain Ω is either unbounded in the horizontal direction and subject to far field BC's or bounded by walls with kinematic BC's.

Numerical Solution

The solution for the velocity potential is found in the finite physical domain $\Omega = [0, L_x] \times [0, L_y] \times [-h, 0]$, and the Laplace equation and the boundary condition are solved using a unit cube computational domain $\Xi = [0, 1]^3$. The transformation between the computational domain and the physical domain is done with algebraic grid stretching functions and transfinite interpolation, and the FD discretization of the Laplace equation and outer boundary conditions are described in [2], [3] and [1].

The following describes the WLS approximation of the body BC (4) and the coupling with the FD approximation of the Laplace equation (1). The Laplace equation is only solved on *fluid points*, *i.e.* the points outside the bodies. To distinguish between points inside and outside the bodies, a sign function is defined on the FD grid which is negative inside bodies and positive outside bodies

$$sgn(\boldsymbol{x}) = \begin{cases} -\texttt{bodyID} & \text{if } \boldsymbol{x} \in \Omega_B,\\ 1 & \text{else,} \end{cases} \qquad (5)$$

where `bodyID` is a unique body index, and Ω_B is the domain inside the bodies. The points inside the bodies which belong to a FD stencil of a point outside a body are marked as *ghost points*. The degrees of freedom associated with ghost points are used to satisfy the body BC. The points on the ship hull where the body BC are satisfied are the *body points*, and they are found from a normal projection of the ghost points onto the ship hull. A stencil $\boldsymbol{x}_{i,j,k} \in \mathcal{S}$ of fluid points is associated with each ghost point. The WLS stencil points are found from a centered stencil around the body point, with stencil half width $\Delta_{\mathcal{S}}$. Only points outside the body tangent plane through the body point are included in the stencil. The body BC is approximated along the normal coordinate $n = \boldsymbol{x}\cdot\boldsymbol{n}$, and the WLS coefficients for the normal derivative $\frac{\partial\phi}{\partial n} = \boldsymbol{n}\cdot\nabla\phi$ are found by Taylor expansions from the body point $\boldsymbol{x}_B$ to the points in the WLS stencil

$$\begin{aligned} \phi_{i,j,k} &\approx \sum_{|\boldsymbol{\alpha}|=0}^{\boldsymbol{\alpha}_{max}} \frac{\Delta\boldsymbol{n}_{i,j,k}^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}(D^{\boldsymbol{\alpha}}\phi)_{\boldsymbol{x}=\boldsymbol{x}_B},\\ \Delta\boldsymbol{n}_{i,j,k} &= \mathbf{N}(\boldsymbol{x}_{i,j,k} - \boldsymbol{x}_B), \quad \forall \boldsymbol{x}_{i,j,k} \in \mathcal{S} \qquad (6) \end{aligned}$$

where $\mathbf{N}$ is the basis change matrix which contains the body surface normal vector and two body surface tangent vectors as basis vectors. The expansions are arranged in a linear system of equations $\mathbf{A}\boldsymbol{d} = \boldsymbol{f}$ where $\boldsymbol{d}$ is the vector containing the derivatives $D^{\boldsymbol{\alpha}}\phi$, and $\boldsymbol{f}$ is the vector containing the function values at the stencil points. A weight is associated with each stencil point, and they are arranged in a diagonal matrix $\mathbf{W}$. The linear system is overdetermined, and the solution is found by weighted quadratic minimization of the residual, resulting in a linear system of equations $\mathbf{A}^T\mathbf{W}\mathbf{A}\boldsymbol{d} = \mathbf{A}^T\mathbf{W}\mathbf{f}$, and it has a unique solution provided that the square matrix $\mathbf{A}^T\mathbf{W}\mathbf{A}$ is non-singular. The coefficients for the normal derivative are in the second row of the matrix $\left[\mathbf{A}^T\mathbf{W}\mathbf{A}\right]^{-1}\mathbf{A}^T\mathbf{W}$.

Parallel GPU Implementation of the WLS Body BC

The GPU implementation of the FD part of the model is described in [2] and [3]. Unlike the FD method, the WLS approximation is based on an unstructured point set. Hence it is not possible to immediately

utilize the data structure of the structured FD grid. Instead the following connectivity information is needed for each ghost point: The index of the body point, the direction vector from the ghost point to the body point, the stencil index list and a list with the WLS coefficients for the body BC. In the current implementation, the WLS coefficients are calculated on the CPU and copied to the GPU where they are stored in the global memory. The Laplace equation is solved on the GPU using a multigrid preconditioned PDC solver in which the residuals are calculated using a matrix-free implementation of the matrix-vector product between the FD Laplace operator and the solution vector. All the boundary conditions are imposed on the solution prior to each multiplication with the FD Laplace operator. The WLS approximation of the body BC is given by the inner product of the solution and the WLS coefficients

$$\frac{\partial \phi}{\partial n} \approx \sum_{m \in \mathcal{S}} a_m \phi_{i(m),j(m),k(m)} = \boldsymbol{n} \cdot \boldsymbol{u}_B. \tag{7}$$

The body BC's are independent of each other, and the solution values at the ghost points can be updated explicitly in independent CUDA kernels. A special Red-Black Z-line smoother is used where the one-dimensional FD Laplace problem in the vertical direction is solved with Gauss-Seidel iteration instead of a direct modified Thomas algorithm [3]. This is done to avoid the difficult implementation of a Thomas algorithm for the WLS IBM.

Two Spheroids in a Rectangular Channel

Interaction between two spheroids in a channel with a rectangular cross section is considered as an example application, similar to the example in [5]. Each spheroid has length $L = 1.0$ m, breadth $B = 0.2$ m and draft $D = 0.1$ m. The channel has length $L_x = 4.0$ m, width $L_y = 1.0$ m and depth $h = 0.25$ m. The first spheroid has a fixed position in the channel at $x_0 = 2.0$ m, $y_0 = 1/3$ m, measured at the center of the spheroid. The second spheroid has an initial position at $x_0 = 1.0$ m, $y_0 = 2/3$ m and a constant forward speed of $U = 0.2$ m/s. This corresponds to a depth-based Froude number of $Fr_h = U/\sqrt{gh} = 0.13$ and length-based Froude number of $Fr_L = U/\sqrt{gL} = 0.06$, hence the speed of the moving spheroid is relatively low in both cases. The forces on the fixed spheroid are examined over the time interval $t = [0\text{ s}, 10\text{ s}]$, that is the time it takes for the moving spheroid to pass the fixed spheroid. The FD and WLS approximations are both of second order. The tolerance on the relative residual in the PDC solver is $\texttt{rtol} = 10^{-4}$. The channel with the spheroids in their initial position and an example of the calculated velocity potential are seen in Figure 1. The computational performance for execution on the GTX 680 GPU with 1536 cores and 2048 MB RAM and the calculated forces are seen in Figure 2. Note that the GPU implementation is not yet optimized, and the setup of the *sgn* function and the WLS BC are still done sequentially on the CPU.

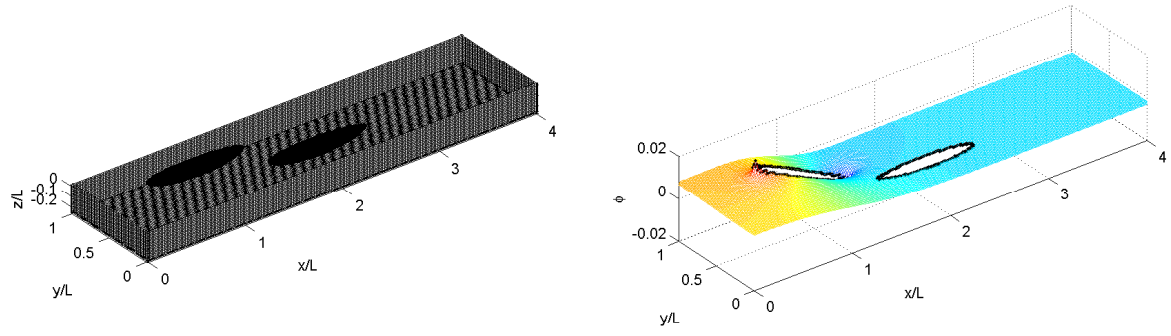

Figure 1: Left: Initial configuration of the two spheroids in a rectangular channel. The size of the FD grid is $257 \times 65 \times 17$. Right: The corresponding velocity potential at the surface at $t = 0$. The black dots mark the ghost point associated with the body BCs.

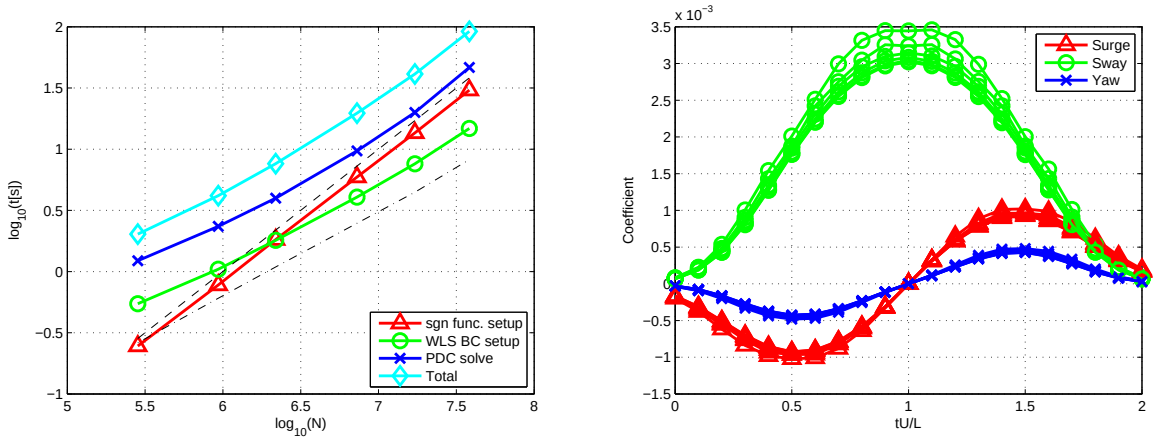

Figure 2: Left the absolute computation time per time step for grid sizes of $257 \times 65 \times 17$, $385 \times 97 \times 25$, $513 \times 129 \times 33$, $769 \times 193 \times 49$, $1025 \times 257 \times 65$ and $1537 \times 257 \times 97$. The $--$ line shows $\mathcal{O}(N)$ scaling, and the $- \cdot -$ line shows linear scaling with the number of ghost points. Right: Surge, sway and yaw on the fixed spheroid for the same problem sizes as on the left figure.

Conclusion

A new WLS IBM for the inhomogeneous Neumann body BC has been presented. The WLS IBM is combined with the FD approximation of Laplace equation for the velocity potential. This combination enables calculation of flow around complex geometries like a ship hull without the trouble of complicated grid generation. An example application with interaction between two spheroids in a channel demonstrates the potential of the new WLS IBM. The efficient multigrid preconditioned PDC solver and the parallel GPU implementation makes it possible to solve the double body flow problem within the range of real-time application. Significant performance gains are still to be expected from algorithmic tuning, implementation optimization and better hardware.

[1] A. P. Engsig-Karup. Analysis of efficient preconditioned defect correction methods for nonlinear water waves. *Int. J. Num. Methods in Fluids*, 2014. to appear.

[2] A. P. Engsig-Karup, H. B. Bingham, and O. Lindberg. An efficient flexible-order model for 3d nonlinear water waves. *Journal of Computational Physics*, 228:2100–2118, 2009.

[3] A. P. Engsig-Karup, M. G. Madsen, and S. L. Glimberg. A massively parallel gpu-accelerated model for analysis of fully nonlinear free surface waves. *Int. J. Num. Methods in Fluids*, 2011.

[4] S. L. Glimberg. *Designing Scientific Software for Heterogeneous Computing*. PhD thesis, Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2013.

[5] F. T. Korsmeyer and C. H. N. J. N. Lee. Computation of ship interaction forces in restricted waters. *Journal of Ship Research*, 4:298–306, 1993.

[6] O. Lindberg, H. B. Bingham, A. P. Engsig-Karup, and P. A. Madsen. Towards Realtime Simulation of Ship-Ship Interaction. In *The 27th Int. Workshop on Water Waves and Floating Bodies*, 2012.

[7] O. Lindberg, S. L. Glimberg, H. B. Bingham, A. P. Engsig-Karup, and P. J. Schjeldahl. Real-Time Simulation of Ship-Structure and Ship-Ship Interaction. In *Third International Conference on Ship Manoeuvring in Shallow and Confined Water*, 2013.

[8] O. Lindberg, S. L. Glimberg, H. B. Bingham, A. P. Engsig-Karup, and P. J. Schjeldahl. Towards Real Time Simulation of Ship-Ship Interaction- Part II: Double Body Flow Linearization and GPU Implementation. In *The 28th Int. Workshop on Water Waves and Floating Bodies*, 2013.

[9] X.-D. Liu, R. P. Fedkiw, and M. Kang. A boundary condition capturing method for poissons equation on irregular domains. *Journal of Computational Physics*, 160:151–178, 2000.