Real-time Dynamics Simulation of Unmanned Sea Surface Vehicle for Virtual Environments

Atul Thakur
Simulation-Based System Design Laboratory
Department of Mechanical Engineering
University of Maryland
College Park, Maryland 20742
Email: athakur@umd.edu

Satyandra K. Gupta*
Simulation-Based System Design Laboratory
Department of Mechanical Engineering and the Institute for Systems Research
University of Maryland
College Park, Maryland 20742
Email: skgupta@umd.edu

ABSTRACT

The role of virtual environments (VE) is crucial in efficient design and operation of unmanned vehicles. VEs are extensively used in operator training for tele-operation, planning using programming by demonstration, and hardware and software design. VE for unmanned sea surface vehicles (USSV) requires a six degree of freedom dynamics simulation in the time domain. In order to be interactive, the VE requires real-time performance of the underlying dynamics simulator. In general, the dynamics simulation of USSVs involves the following four main operations: (1) computation of dynamic pressure head due to fluid flow around the hull under the ocean wave, (2) computation of wet surface, (3) computing the surface integral of the dynamic pressure head over the wet surface, and (4) solving the rigid body dynamics equation. The first three operations depend upon the boat geometry complexity and need to be performed at each time step, making the simulation run very slow. In this paper, we address the problem of physics preserving model simplification for real-time potential flow based simulator for a USSV in the time domain, with an arbitrary hull geometry. The paper reports model simplification algorithms based on clustering, temporal coherence and hardware acceleration using parallel computing on multiple cores to obtain real time simulation performance for the developed VE.

1 Introduction

Unmanned vehicles have emerged as an important tool in search, rescue, recovery, and surveillance operations. VE play a significant role in the effective design and operation of unmanned vehicles. First, VEs are useful for training operators that tele-operate unmanned vehicles. Virtual environment-based training significantly reduces the training cost and enables training of a large number of operators without requiring significant time on the physical platforms. Second, programming by demonstration has emerged as an important paradigm for acquiring autonomous behavior [1]. Virtual environments are expected to play a very important role in the deployment of programming by demonstration paradigm. Third, VEs can be a very valuable tool for hardware and software design of unmanned vehicles. For example, new designs can be first prototyped in the virtual environment.

In the paper we are focusing on the development of a VE for simulating USSVs. This environment will be used for simulating the motion of USSVs under different sea states to facilitate operator training, design, planning, and control of the USSVs. Figure 1 shows a snap-shot of the VE developed by [13]. The VE simulates a typical maritime mission where the USSV's goal is to protect the oil tanker from the approaching intruder with malicious intent. For example, in the Figure 1,

*Address all correspondence to this author.

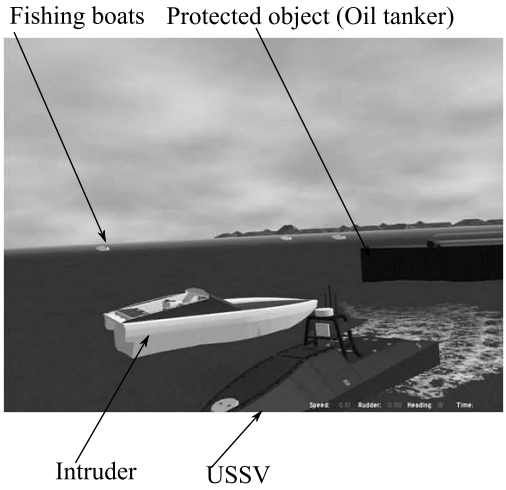


Fig. 1: *USSV Virtual Environment developed by [13].*

the operator can drive the USSV and demonstrate the strategies to block the intruder from reaching the protected object. The efficacy of the planning and control strategy obtained from any of the algorithms significantly depends upon the accuracy and computational speed with which the motion of the boats moving in the ocean can be predicted by the simulation. The problem of motion simulation of boats under the influence of the ocean waves is known as sea-keeping. Sea-keeping is inherently a computationally intensive operation, as it is an instance of the rigid-fluid two way coupling problem.

One way to solve the sea-keeping problem is to solve it as a two-way coupling problem. The two way coupling can be understood as the simultaneous effect of fluid flow on the motion of a floating USSV and the effect of the moving USSV on the surrounding fluid flow. Computational fluid dynamics (CFD) based approaches are very accurate in solving such problems. Unfortunately, the speed of computation using CFD approaches is extremely slow and there is hardly any hope to perform CFD simulation in real time using the current computing technology.

Another approach used for sea-keeping simulation is the strip-theory based computations, wherein the boat geometry is idealized as the nearest regular geometry such as ellipsoid, cylinder, etc. The approximated formulas for computing the hydrostatic and hydrodynamic forces using the idealized geometries are then used. The strip theory based approaches are very fast but introduce significant errors. Moreover the technique is insensitive to variations in hull geometries. In other words, various hull geometries can be idealized into the same regular geometry leading to same force acting on them.

Potential flow theory based approaches imposes assumptions of irrotational, inviscid, and incompressible flow. Due to the assumptions, the Laplacian of the velocity potential becomes zero. The kinematic boundary condition involving the movement of fluid on the free surface and the dynamic boundary condition involving the pressure acting on the free surface is used to solve the partial differential equation. The solution yields velocity potential, which in conjunction with the Bernoullis law is used to compute the pressure due to the fluid flow. The pressure is then used to compute the force acting on the floating boat. The effects of boat on fluid flow are modeled as the added mass. The turbulence and viscosity related effects are modeled by the damping matrix. This leads to a much faster although a less accurate approach compared to CFD based approaches. The potential flow based approach is much more accurate compared to the classical strip theory based approaches. In most of the applications where a moderate accuracy is desired at a higher computational speed, the potential flow theory based approaches prove to be the acceptable choice for the sea-keeping simulation.

Although, potential flow based approach is faster than the CFD based approach, it is far from real time. The reason for slow computation of USSV motion simulation can be explained as follows. The motion simulation using potential flow theory involves the following four main operations: (1) computation of the dynamic pressure head due to the fluid flow around the hull under the influence of the ocean wave, (2) computation of the instantaneous wet surface, (3) computation of the surface integral of the dynamic pressure head over the wet surface, and (4) solving the rigid body dynamics equation. First three operations depend upon the boat geometry complexity and need to be performed at each time step, making the simulation run very slow. To obtain the real time performance, in this paper we describe simplification algorithms based on clustering, parallelization, and temporal coherence. These algorithms have been incorporated into the USSV virtual simulation system as shown in Figure 1.

The paper is organized as follows. In the section 2 we discuss the related work. In section 3 we present the governing equations of the fluid flow and rigid body dynamics model. Section 4 presents the problem of model simplification in the context of the seakeeping simulation. Sections 5 to 8 present our algorithms to reduce the computation time for the seakeeping simulation.

2 Literature Review

A survey on sea keeping computations reported by Beck and Reed in the year 2001 [3] classifies the methods of computation into two broad categories, namely, frequency based and time based. Earlier approaches using linearized computation in the frequency domain was found successful for zero speed simulation, however for forward speed problems, the frequency domain techniques did not work well [12, 14]. Time based techniques can be classified into strip theory based, potential flow theory based and Reynold Averaged Navier Stokes (RANS) code based techniques.

The Strip theory is mainly used for slowly moving slender geometries [10]. Strip theory based techniques are not suitable for taking non-linear effects due to the hull geometry and wave interactions. This is because, in strip theory, the hull geometry is approximated to the nearest ideal shape (such as ellipsoids, spheres, etc.). This idealization might yield significant errors in hydrodynamic and hydrostatic force estimations. In potential flow theory, fluid flow is assumed to be irrotational and inviscid [22].

The applicability of the potential flow theory is wider compared to the strip theory based method in terms of hull geometry and operating speed. However, phenomenon like boundary layer effects and turbulence are ignored in potential flow methods. The implementations of the potential flow based technique to compute the force acting on the boat geometry due to the ocean waves and the motion of the boat under the action of such forces can be found in [18, 23].

In recent years, RANS based techniques for fluid flow around boats for simulating boat motion are becoming popular because of their accuracy in the problems involving boundary layer effects, turbulence, wake etc. [11] The limitation of RANS code based techniques is the slow computation. In one of the implementations of RANS code by Kim, the reported computation time for 360 time steps is about 24 hours using 84 processors on Mauis IBM-SP3 computer [16]. Some of the other implementations of RANS code can be found in [6, 17, 24].

Some of the related techniques, wherein the two way coupling problem between a fluid and a floating rigid body simulation (not necessarily USSVs or boats) is solved using the following approaches: Euler's momentum equation, Navier-Stokes law, Smoothed Particle Hydrodynamics (SPH) technique and Lattice Boltzmann Method (LBM). In Euler's momentum equation based techniques, the momentum equation in continuum mechanics is solved for fluids numerically. Batty et al. reported a computation time of 25 seconds per frame using a grid size of 60×90 [2]. In Navier-Stokes law based techniques, interaction between a viscous fluid and rigid bodies are simulated by numerically solving the Navier-Stokes equations. Carlson et al. reported a computation time of 27.5 seconds per frame for a domain of size 64×64 [5]. The problem with the Euler's equation and the Navier-Stokes equation based techniques is the dependence of computation time on the domain size and the slow speed of computation despite the impressive accuracy. In SPH technique, the fluid is assumed as a collection of particles and the motion of fluid particles and their effect on a floating rigid body is modeled based on a kernel function weighted by the distance of the particle from the floating rigid body. Becker et al. reported a computation time of 3.47 seconds per simulation step for simulating fluid with 850000 particles [4]. The problem with the SPH is requirement of large number of particles to simulate the fluid.

There are several papers reported in the area of underactuated controller design for USSVs that utilize 3 degrees of freedom simplified kinematic models which neglects the rolling, pitching and heaving motions [8, 15, 19–21]. 3DOF simulations are computationally fast, however, results of such simulations may not be accurate because they neglect dynamics effects.

In this paper, we utilize a potential flow theory based fluid flow model and assume the USSV as a rigid body, incorporating the environmental effects such as the wave-boat interaction forces, damping, restoring forces, and the added mass effect to simulate the USSV. We report new physics preserving model simplification techniques based on (1) clustering of boat model facets, (2) parallel computing, and (3) temporal coherence to achieve significant speed up in the simulation without introducing large errors.

3 USSV Dynamics Simulation

In this section we present the governing equations of the model we implemented followed by a brief description of the developed simulator and time profile for computationally intensive steps in the simulation. The governing equations of fluid flow and boat motion discussed in this section are adapted from Newman [22], Fossen [9], and Krishnamurthy et al. [18].

3.1 Governing Equation

For determining the dynamic pressure due to wave boat interaction we used the well known potential flow theory as given in Newman's textbook [22]. We assume the flow to be irrotational ($\nabla \times \vec{V} = \vec{0}$), where, $\vec{V}$ is the fluid velocity. This means that there exists a velocity potential ϕ, such that,

$$\nabla \phi = \vec{V} \tag{1}$$

Now, let us assume that the fluid is incompressible ($\nabla.\vec{V}$), which gives us Laplace's equation.

$$\nabla^2\phi = 0 \tag{2}$$

The boundary conditions are:

i Kinematic boundary condition: A fluid particle on the free surface of fluid will remain on the free surface.
ii Dynamic boundary condition: Pressure experienced by a particle on the free surface is the same as the atmospheric pressure.

The above boundary conditions are reduced to Equation 3 as given by Newman [22].

$$g\frac{\partial\phi}{\partial z} + \frac{\partial^2\phi}{\partial t^2} + 2\nabla\phi.\nabla(\frac{\partial\phi}{\partial t}) - \frac{1}{g}\frac{\partial\phi}{\partial t}\frac{\partial}{\partial z}\left(\frac{\partial^2\phi}{\partial t^2} + g\frac{\partial\phi}{\partial z}\right) = 0 \tag{3}$$

The solution of Equation 2 satisfying the boundary condition (Equation 3) yields the expression for the velocity field ϕ and an ocean wave free surface ζ for the ocean bed with infinite depth as shown below [22].

$$\phi = \frac{gA}{\omega}\exp(kz)\sin(kx\cos\theta_w + ky\sin\theta_w - \omega t) \tag{4}$$

where,
g: acceleration due to gravity in m/s^2,
θ_w: wave direction in *radian*,
ω: wave frequency in $radian/s$, and
A: wave amplitude in m.

$$\begin{aligned}\zeta(x,y,t) = A\cos(kx\cos\theta_w + ky\sin\theta_w - \omega t) + \\ 0.5A^2k\cos(2kx\cos\theta_w + 2ky\sin\theta_w - 2\omega t)\end{aligned} \tag{5}$$

The wave number k is related to the wave frequency ω for the infinite depth by the dispersion relationship given in Equation 6.

$$k = \frac{\omega^2}{g} \tag{6}$$

The velocity potential ϕ and the wave elevation ζ thus obtained is the starting point for determining the dynamic pressure head due to the wave boat interaction. Using Bernoulli's equation, the dynamic pressure Φ can be expressed in terms of the velocity potential ϕ as follows:

$$\Phi(x,y,z,t) = -\rho\left[\frac{\partial\phi}{\partial t} + \frac{1}{2}\nabla\phi.\nabla\phi\right] \tag{7}$$

The dynamic pressure head Φ can be integrated over the instantaneous wet surface of the boat S_B to obtain the force due to the wave boat interaction.

$$F_W = \begin{bmatrix} \rho\oint_{S_B}\left[\frac{\partial\phi}{\partial t} + 0.5\nabla\phi.\nabla\phi\right]d\vec{S} \\ \rho\oint_{S_B}\left[\frac{\partial\phi}{\partial t} + 0.5\nabla\phi.\nabla\phi\right]\left(\vec{r}\times d\vec{S}\right) \end{bmatrix} \tag{8}$$

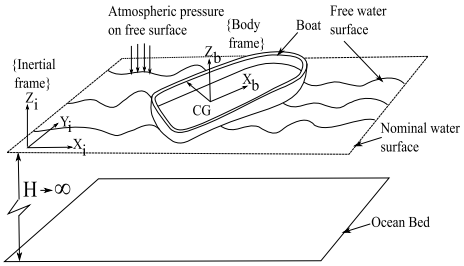


Fig. 2: *Inertial and body coordinate system.*

where,

S_B: instantaneous wet surface of the USSV.

We implemented the 6-DOF dynamics model for USSVs given by Fossen [9]. In this model, the USSV is assumed to be a rigid body. The origin of the inertial frame of reference is set at the nominal water level with Z axis vertical and pointing upwards. The body coordinate system is attached to the boat's center of gravity. The coordinate systems used are shown in Figure 2.

The governing dynamics equation is given in Equation 9.

$$\begin{aligned} &M_H\dot{v} + C_H(v)v + D_H(v)v + g(p) = F_E + F_P \\ &\dot{p} = J_p(v) \end{aligned} \tag{9}$$

where,

$p = [x, y, z, \theta_x, \theta_y, \theta_z]^T$: pose vector expressed in the inertial frame, (x, y, z) is the Cartesian coordinate in m and θ's are Euler angles about subscript axes in *radians*,

$v = [v_x, v_y, v_z, \alpha_x, \alpha_y, \alpha_z]^T$: velocity vector expressed in the body frame relative to the inertial frame, ($v_t = [v_x, v_y, v_z]^T$) is linear velocity in m/s and $v_r = [\alpha_x, \alpha_y, \alpha_z]^T$'s is angular velocity in $radian/s$,

$R = \begin{bmatrix} c_y c_z & s_x s_y c_z - c_x s_z & c_x s_y c_z + s_x s_z \\ c_y s_z & s_x s_y s_z + c_x c_z & c_x s_y s_z - s_x c_z \\ -s_y & s_x c_y & c_x c_y \end{bmatrix}$: rotation matrix rotating a vector expressed in the body frame to the inertial frame, c_x means $\cos\theta_x$,

$J = \begin{bmatrix} R & 0_{3\times3} \\ 0_{3\times3} & J_r \end{bmatrix}$: Jacobian Matrix,

$J_r = \begin{bmatrix} 1 & s_x t_y & c_x t_y \\ 0 & c_x & -s_x \\ 0 & \frac{s_x}{c_y} & \frac{c_x}{c_y} \end{bmatrix}$,

$\vec{x}\times \equiv S(\vec{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$: matrix dual (for cross product) of vector $\vec{x} = [x_1, x_2, x_3]^T$,

$p_{G,B}$: vector representing the position of center of gravity in the body frame of reference,

m: mass of the USSV in kg,

I_b: 3×3 matrix representing the inertia tensor of the USSV in kgm^2,

$M_{RB} = \begin{bmatrix} mI_{3\times3} & -mS(p_{G,B}) \\ mS(p_{G,B}) & I_b \end{bmatrix}$: matrix representing inertia tensor of the USSV,

M_A: (6×6) diagonal matrix representing the added mass of the USSV,

$$
\begin{aligned}
M_{A,11} &= 0.1m \\
M_{A,22} &= 4.75\rho a^2 \\
M_{A,33} &= 4.75\rho a^2 \\
M_{A,44} &= 4.75\rho a^2 \\
M_{A,55} &= 0.396\rho a^2 L_x^2 + 0.0833\frac{0.1m}{L_X} L_z^3 \\
M_{A,66} &= 0.0833\frac{0.1m}{L_X} L_y^3 + 0.396\rho a^2 L_x^3
\end{aligned}
$$

,

where L_x, L_y, and L_z are length, width and height of the bounding box of the hull respectively,

$M_H = M_{RB} + M_S$: $(6\ times 6)$ matrix representing the the total inertia,

$C_{RB} = \begin{bmatrix} mS(v_r)_{3\times 3} & -mS(v_r)S(p_{G,B}) \\ mS(v_r)S(p_{G,B}) & -S(I_b v_r) \end{bmatrix}$: Coriolis and centripetal matrix,

$C_A = \begin{bmatrix} 0_{3\times 3} & -S(M_{A,11}v_t + M_{A,12}v_r) \\ -S(M_{A,11}v_t + M_{A,12}v_r) & -S(M_{A,21}v_t + M_{A,22}v_r) \end{bmatrix}$: matrix representing the effect of added mass,

$M_{A,ij}$ represents (i, j) sub-matrix of M_A of size 3×3,

$C_H = C_{RB} + C_A$: 6×6 matrix representing the total effect of Coriolis and added mass term,

D_H: 6×6 damping matrix,

$g(p)$: 6×1 vector representing the restoring force expressed in the body frame in N,

F_E: 6×1 vector representing the environment force vector expressed in the body frame in N,

F_W: 6×1 vector representing the ocean wave and the boat interaction force expressed in the body frame in N, and

F_P: 6×1 actuation force vector expressed in the body frame in N.

In the right hand side of Equation 9, term F_E means environmental force. The environmental force consists of effects due to ocean current, wind, and ocean wave. We ignore the effects of ocean current and wind and only consider the force due to ocean wave F_W in the implementation. However, given a suitable model of the force due to ocean current and wind, it can be easily incorporated into the simulation framework. We thus replace F_E by F_W in Equation 9. The force due to ocean wave is computed by using Equation 8, which is obtained by the potential flow theory in the implementation.

3.2 Implementation of Simulator

We implemented the model given in the previous section to simulate the motion of a USSV under given initial state and wave conditions. We assume that the USSV geometry is provided in polygonal form. We chose .STL as the input file format for the USSV model. To compute the wave force we discretized Equation 8 as follows:

$$
F_W = \begin{bmatrix} \rho \sum_{j=1}^{n} \Phi_j d\vec{S}_j \\ \rho \sum_{j=1}^{n} \Phi_j \left(\vec{r}_j \times d\vec{S}_j \right) \end{bmatrix} \tag{10}
$$

where,

$$
\Phi_j = \left[\frac{\partial \phi}{\partial t} + 0.5\nabla\phi.\nabla\phi \right]_{p_j} \tag{11}
$$

and n is the number of instantaneous wet facets. So, the steps of simulation are enumerated below.

i Determine the instantaneous wet surface (S_B) of the USSV by finding out the facets lying below and on the wave surface given by Equation 5.
ii Use Equation 4, 7, 10, and S_B determined in the previous step to compute the wave force F_W.
iii Determine the displaced volume of water by the surface S_B and compute the buoyancy force $g(p)$.
iv Determine the inertia matrix (M_H), Coriolis matrix ($C(v)$) and damping matrix ($D(v)$). We estimated the added mass matrix M_A using strip theory as explained by Fossen [9]. The added mass can, in practice, be computed by performing system identification for a given USSV model under various sea-states. For the simulation purpose, we utilized strip theory to estimate the added mass matrix M_A. It should be noted that strip theory is only used here to estimate the added mass, and not the forces due to ocean wave.
v Solve Equation 9 numerically using Runge-Kutta fourth order (RK4) integration technique by substituting $F_E = F_W$.

3.3 Time Profile of Simulator

The simulator explained in the previous section has three main operations being repeated in each time step as listed below:

i Computation of the list of wet facets.
ii Computation of the surface integral for determining the wave force and the buoyancy force.
iii Solving the differential equation given in Equation 9.

We ran the simulation for five different wave directions namely $\theta_w = 0, 0.63, 1.25, 1.88, 2.51$ radian. The boat model for the simulation had 960 triangular facets. We computed the average time taken for the boat model for 1500 simulation time steps (with each time step of length 0.07 sec) for each operation over all five wave directions and the result is shown in Table 1. We performed the simulation on a computer with Intel(R) Core(TM)2 Quad 2.83GHz CPU and 8GB RAM. All the computations reported in this paper are performed on the same computer and for the same boat model, so that the comparisons between computation times are fair.

Table 1: *Typical time profile for the simulation process for* 1500 *time steps of size* 0.07 *s.*

Computation Item	Computation Time (seconds)	Percentage
Ocean wave force	740.70	62.45
Wet facet list	444.65	37.49
ODE solution	0.67	0.06
Total	1186.02	100.00

It is evident from the Table 1 that most of the time (about 99.94%) is spent in computing the force due to ocean waves and wet surface determination. Also, Table 1 suggests that the time taken for computing one time step of length 0.07 s is about $\frac{1186.02}{1500} \approx 0.8$ s, which is far from the real time performance.

4 Problem Statement and Solution Approach

As discussed in the previous section, the computation time for the USSV simulation is too high for attaining real time refresh rate, which motivates us to find ways to simplify the computations to reduce the computation time. In this section we shall present the problem statement formally and give an overview of approach to solve the problem.

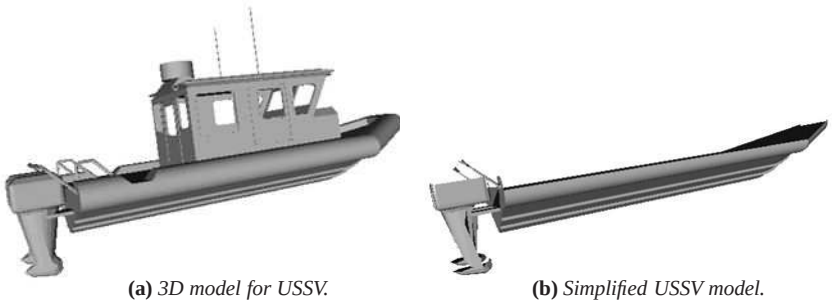
4.1 Problem Statement

Given a polygonal representation of the USSV hull geometry and its dynamics properties like mass, moment of inertia tensor, damping matrix, etc. and fluid and ocean wave characteristics such as density, wave direction, amplitude, frequency, etc., perform model simplification to do dynamics simulation of the USSV in real time.

4.2 Approach

Based on the observations in the previous section, the computations can be made faster by:

i Determining the list of facets in the USSV model, which will never come in contact with the water under normal operating conditions (without USSV sinking completely or getting turned over) and removing them to get the wettable region of the boat model.



(a) *3D model for USSV.* **(b)** *Simplified USSV model.*

Fig. 3: *USSV model simplification by removing facets that never come in contact with water.*

ii Performing clustering of the USSV model facets to club all the facets with similar dynamic pressure value, in order to reduce the time taken to compute the wet facets and force due to ocean wave and hull interaction.
iii Exploiting the parallel nature of the computation and reducing the computation time by utilizing multi-core computers.
iv Estimating the force due to ocean wave acting on the USSV in a time step using force in the previous time step instead of explicitly computing forces in each time step. We call this approach as temporal coherence, as the forces do not change significantly in short period of time and depend on the force in the previous time step.

In the following sections we discuss the above simplification strategies in detail.

5 Preprocessing of Boat Model to Determine Wettable Region

A USSV model can be very complicated and can have many unnecessary details from the point of view of computing hydrostatic and hydrodynamic forces. If the features that are covered by the hull are retained during the force computations then the following problems might arise:

i The facets of the features covered by the hull would be determined as the wet facets and would introduce error into the force computation. This is because, those facets never come in contact with the water as they are occluded by the facets of the boat hull.
ii Testing of such facets during the simulation for whether they touch water, would be an additional computational overhead during the simulation runtime leading to performance deterioration.

For computation of hydrodynamics and hydrostatic forces, we are only interested in facets that might come in contact with the water (without the USSV sinking completely or getting turned over). This is a simple problem to solve but a crucial step for preprocessing the USSV model. We assume that the USSV model is oriented in such a way that the axis of yaw is aligned to inertial Z axis. Also, we assume that the undercut features on the boat hull are negligible. The steps we follow to automate this process are enumerated below:

i For all facets, if the normal makes an angle larger than 120^o, with the negative Z direction, remove them.
ii For all the remaining facets, shoot ray from the center of gravity of the facets in the negative Z direction. If the rays intersect at least one facet other than itself, remove the facet.

By following the above two steps the wettable region of the boat model as shown in Figure 3 is obtained.

6 Simplification Using Clustering

The preprocessed *wettable* model obtained by the technique described in the previous section is used for determining the hydrodynamic and hydrostatic forces by computing the surface integral of the dynamic pressure head acting over the USSV's instantaneous wet surface. The computational effort spent on the surface integral computation using Equation 10 can be enumerated as follows:

i Computing the value of Φ at each facet center in the list of wet facets S_B in each time step using Equation 12.

$$\Phi(x,y,z) = \frac{gAk}{\omega}\exp(kz)(\cos\theta_w\cos f\dot{x}+ \sin\theta_w\cos f\dot{y}+\sin f\dot{z}-\frac{\omega}{k}\cos f)+0.5(\frac{g^2k^2A^2}{\omega^2}\exp(2kz)) \tag{12}$$

where,

$$f = kx\cos\theta_w + ky\sin\theta_w - \omega t \tag{13}$$

Equation 12 is obtained by using Equation 4 and 11, in terms of wave amplitude, wave number, wave frequency and wave direction.

ii Evaluating the sum in Equation 10.

A look at Equation 12 shows that we need to perform at least four trigonometric evaluations and one exponential function evaluation. Now, even if there are on an average 500 wet facets in each time step of the differential equation solver, we need to perform 2000 evaluations of trigonometric and exponential functions. Trigonometric and exponential function calls are computationally expensive and must be minimized. In order to minimize such function calls we can make use of the fact that the dynamic pressure values at the facets close to each other are similar. Such contiguous facets can be clustered to reduce the number of trigonometric and exponential function calls by introducing some small error. It should be noted that the clustering is performed before the simulation (i.e. off-line) and thus the computational complexity of clustering does not affect the simulation speed.

In this section, we introduce the concept of clustering the hull facets to reduce the time taken to compute the force due to the ocean waves and to compute the wet facets. We also theoretically analyze the effect of clustering the facets on the computation time and error introduced and provide numerical validation of the theoretical results.

6.1 Clustering Algorithm

Let us represent the USSV's center of gravity (or the origin of the body frame of reference attached to the USSV) by $\vec{x_c}$, and position of j^{th} wet triangular facet by $\vec{x_j}$ in the inertial frame of reference. Also, let us represent, the position of the j^{th} wet triangular facet by $\vec{r_j}$ with respect to $\vec{x_c}$. Now, from Equation 11 and using Taylor series expansion, Φ at a given wet triangular facet can be expressed as follows:

$$\Phi(\vec{x_j}) = \Phi(\vec{x_c} + \vec{r_j}) \approx \Phi(\vec{x_c}) + \vec{r_j}^T \nabla\Phi(\vec{x_c}) + 0.5\vec{r_j}^T \nabla^2\Phi(\vec{x_c})\vec{r_j} \tag{14}$$

We can ignore higher order terms because the r^{th} term contains $\nabla^{r-1}\Phi$, which is proportional to k^r, and for ocean waves usually $0.0 < k < 1.0$. Thus, the terms $r > 3$ are negligible. Equation 14 suggests that the value of Φ depends only on $\vec{r_j}$ for a given position $\vec{x_c}$ of the USSV. This implies that the value of Φ for facets which are sufficiently close to each other geometrically can be assumed to be the same. This observation establishes the basis of the decomposition of the facet set to form clusters. We define what we mean by *cluster* formally in Definition 6.1.

Definiton 1 A cluster κ of size $\gamma \in \mathbb{R}$ is defined as the set of triangular facets t_j such that the diagonal of the bounding box of κ is smaller than or equal to γ.

Definiton 2 Cluster center of a cluster κ is defined as a point $P_c \in \mathbb{R}$, such that,

$$P_c = \frac{\sum_{i=1}^{M} a_i P_i}{\sum_{i=1}^{M} a_i} \tag{15}$$

where, M is the cardinality of κ and P_i is the centroid of the i^{th} facet belonging to κ.

The problem of clustering can be viewed as a set partitioning problem. The set of facets representing the hull of the USSV is partitioned into subsets (clusters). The partitioning is done in such a way that the average distance of all the facets belonging to a cluster from the cluster center is minimized. We utilized K-means algorithm to accomplish the clustering. The steps of clustering algorithm we used, are listed as follows:

Algorithm - K-means Clustering

Input: List of n triangular facets Θ, cluster count C, and subdivision threshold $\varepsilon \in \mathbb{R}$

Output: List of C clusters Υ such that $C < n$

Steps:

i Determine the bounding box B of Θ.

ii Recursively subdivide each triangle inside Θ using Loop subdivision until triangle area is less than $\varepsilon Diagonal(B)$.

iii Determine $q = \lceil \sqrt{\frac{width(B)}{length(B)} C} \rceil$ and $p = \lceil \frac{C}{q} \rceil$. It can be seen that $pq \geq C$. Divide the XY plane of B into $p \times q$ rectangular grids. If $pq > C$, merge $pq - C$ rectangles. Use the exactly C regions on the XY plane, thus obtained to partition the

set of facets obtained in step (ii) into C subsets. For each region, project vertices of each facet on the XY plane. If at least two of the projected vertices lie inside the region add facet into the subset and mark the facet as visited. Repeat the process for each region and unvisited facets. This step returns C clusters of facets. This heuristic step is performed to generate an initial clustering.

iv Determine the cluster center of each of the C clusters obtained in step (iii), using Equation 15.

v For each cluster center, determine the facets nearest to the cluster center, than any other center. This will yield a new partition and hence new set of C clusters.

vi Determine cluster centers for each of the C new clusters and determine the mean squared distance from the respective old centers.

vii If the mean squared distance is less than a small value (we chose 10^{-5}), return the C clusters, else go to step (v).

We used the clustering approach (with $C = 45$, $\varepsilon = 10^{-3}$) for simulation of the USSV model shown in Figure 3b with 960 facets, for 1500 time steps (representing time step of size 0.07 s) and for five different wave directions. We used Equation 16 to compute the force error introduced due to the clustering simplification.

$$e_F = \frac{\|\langle F_W - g(p)\rangle_{\text{baseline}}\| - \|\langle F_W - g(p)\rangle_{\text{simplified}}\|}{\|\langle F_W - g(p)\rangle_{\text{baseline}}\|} \tag{16}$$

where, subscript baseline signifies the computation performed without using the clustering approximation approach, whereas subscript simplified signifies the computation performed using clustering approximation approach. The average computation time without using clustering simplification was found to be 1191.4s whereas the average computation time with clustering simplification was found to be 216.8s in our tests. Thus, the average computation time reduced by a factor of $\frac{1191.40}{216.80} = 5.50$ with an average force error of 2.88% using the described clustering based simplification approach.

6.2 Effect of Cluster Count on Error and Computation Time

Intuitively speaking, increasing the number of clusters should increase the time taken to compute and should reduce the error. In this section, we derive the dependence of computation time and error introduced on the chosen cluster count and present numerical validation of the derived relationship. Let us assume that the boat hull has n facets and we partition the facets into C clusters using the K-means algorithm discussed before. Let the j^{th} cluster contain N_j facets. Let us denote the total area vector of the facets contained by the j^{th} cluster as $\vec{s_j}$. We can express $\vec{s_j}$ as follows:

$$\vec{s_j} = \sum_{p=1}^{N_j} d\vec{S}_{p,j} \tag{17}$$

where, $d\vec{S}_{p,j}$ is the area of the p^{th} facet of the j^{th} cluster. Based on Table 1 and the discussion in the previous sections, we know that the major amount of time taken in computations can be divided into three main operations:

i Computation of the dynamic pressure

ii Computation of the list of wet facets

iii Computation of the instantaneous volume of the wet region

Let, the time taken for computing dynamic pressure at a given point be T_p, the time taken for doing a test on one cluster for wet facets be T_w, and the average time taken for the computation of total volume of the wet region be T_v. The time taken for computing the volume of the wet region (T_v) depends only on the number of wet facets which for a boat hull does not vary much and thus, we can assume it to be independent of the cluster count.

Firstly, let us estimate the variation of the average time for computation with the cluster count (C). In the case of C clusters, we compute the dynamic pressure C number of times in one time step, and thus, the total time taken for computing the dynamic pressure is CT_p. Similarly, for determining wet facets we perform C tests which takes CT_w time. Thus, the total time (T) taken for computing the force for one simulation time step is given in Equation 18.

$$T = T_v + C(T_w + T_p) \tag{18}$$

The simulation time, thus, varies as $O(C)$ based on Equation 18.

In order to compute the force using clustering, we only compute the dynamic pressures at the cluster centers and use them to compute the total force as follows:

$$\vec{F}_{cluster} = \sum_{j=1}^{C} \Phi(\vec{x}_j)\vec{s}_j \tag{19}$$

where, $\vec{x}_j$ is the cluster center of j^{th} cluster.

The exact total force (computed by considering all facets instead of just the clusters) can be given as follows:

$$\begin{aligned} \vec{F}_{exact} = \sum_{j=1}^{C}\sum_{k=1}^{N_j} \Phi(\vec{x}_j + \vec{r}_{k,j})\vec{dS}_{k,j} = \\ \sum_{j=1}^{C}\left[\Phi(\vec{x}_j)(\sum_{k=1}^{N_j}\vec{dS}_{k,j}) + \sum_{k=1}^{N_j}\left(\vec{r}_{k,j}.\nabla\Phi(\vec{x}_j)\right)\vec{dS}_{k,j})\right] + H\vec{O}T = \\ \sum_{j=1}^{C}\Phi(\vec{x}_j)\vec{s}_j + \sum_{j=1}^{C}\sum_{k=1}^{N_j}(\vec{r}_{k,j}.\nabla\Phi(\vec{x}_j))\vec{dS}_{k,j} + H\vec{O}T = \\ \vec{F}_{cluster} + \sum_{j=1}^{C}\left[\sum_{k=1}^{N_j}(\vec{r}_{k,j}.\nabla\Phi(\vec{x}_j))\vec{dS}_{k,j}\right] + H\vec{O}T \end{aligned} \tag{20}$$

where $\vec{r}_{k,j}$ is the position vector of centroid of the k^{th} facet of the j^{th} cluster, N_j is the number of facets in the j^{th} cluster, and HOT represent the higher order terms. Now, we can compute error in the force estimation $\vec{e}_{force}$ due to the use of the clustering approach as follows:

$$\begin{aligned} \vec{e}_{force} = \vec{F}_{exact} - \vec{F}_{cluster} = \sum_{j=1}^{C}\left[\sum_{k=1}^{N_j}(\vec{r}_{k,j}.\nabla\Phi(\vec{x}_j))\vec{dS}_{k,j}\right] + H\vec{O}T \approx \\ \sum_{j=1}^{C}\left[\sum_{k=1}^{N_j}(\vec{r}_{k,j}.\nabla\Phi(\vec{x}_j))\vec{dS}_{k,j}\right] \text{ ignoring higher order term } H\vec{O}T \end{aligned} \tag{21}$$

The magnitude of error in the force estimation can be expressed and approximated by the following equation.

$$\|\vec{e}_{force}\| = \|\sum_{j=1}^{C}\left[\sum_{k=1}^{N_j}(\vec{r}_{k,j}.\nabla\Phi(\vec{x}_j))\vec{dS}_{k,j}\right]\| \leq R\alpha S \tag{22}$$

where, α is the maximum value attained by the function $\|\nabla\Phi\|$ defined in the instantaneous region bounded by the hull geometry. We know that α must be a scalar and finite value as the hull region is a finite region. S is the absolute value of the surface area of the wet surface, given as follows:

$$S = \|\sum_{j=1}^{C}\sum_{k=1}^{N_j}\vec{dS}_{k,j}\| \tag{23}$$

R is an upper bound on the bounding radius of cluster. Since, the total surface area of the hull is S and the hull is divided into C almost equal area regions, the total surface area $A_{cluster}$ of facets in a cluster can be approximated as follows:

$$A_{cluster} \approx \frac{S}{C} \tag{24}$$

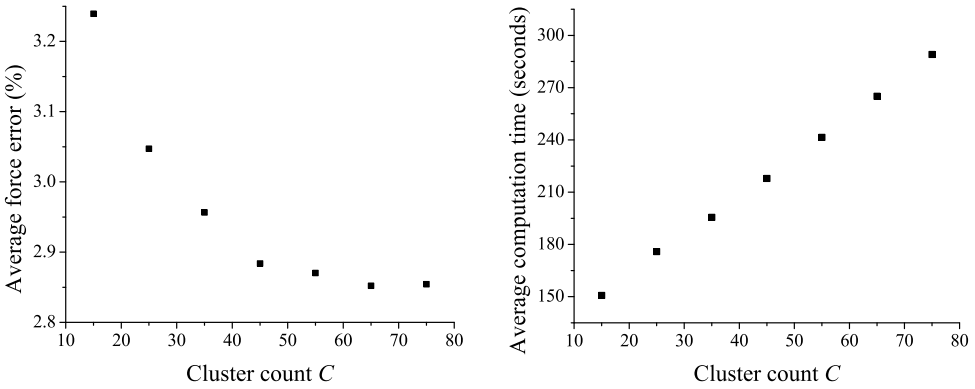
Since the boat geometry is smooth (without many undercuts), the surface area of the bounding sphere of a cluster is always more than the surface area of the facets contained in the cluster We can thus, estimate R as follows:

$$R < \sqrt{\frac{A_{cluster}}{4\pi}} < \sqrt{\frac{S}{4\pi C}} \tag{25}$$

Using, Inequality 25 and Equality 22, we get:

$$\|\vec{e}_{force}\| < \frac{\alpha S^{1.5}}{2\sqrt{\pi C}} \tag{26}$$

Thus the error in the force computation introduced due to the cluster approximation varies as $O(C^{-0.5})$.



(a) *Variation of average force error with cluster count C.*

(b) *Variation of average computation time with cluster count C.*

Fig. 4: *Variation of average force error and computation time with cluster count C.*

We conducted numerical tests to validate the derived results. In these tests, the USSV is hit by waves from five different directions for 1500 time steps (with each time step of length 0.07s) and the computations are performed using cluster sizes ranging from 15 to 75 in the increments of 10. The variation of the average error and average time taken for 1500 time steps (averaged over 5 different wave directions) is plotted against the cluster sizes and presented in Figure 4. It can be observed in Figure 4 that the average computation time increases almost linearly while the error reduces non-linearly with the increasing cluster count agreeing well with Equations 18 and 26. It can be noticed in Figure 4 that the error percentage gets stalled at the value of about 2.85% and does not reduce significantly after $C = 65$. This is because, the triangles defining the boat model are subdivided into smaller triangles before clustering. The clusters are then formed on the basis of geometric nearness of constituent triangles and not the similarity from the original face set. This causes a small error from the baseline computation and causes the stalling phenomenon in Figure 4a. To summarize, cluster count C is a parameter that can be used as an approximation handle to control the time taken for computation and the error introduced due to the clustering approximation.

7 Parallelization

The problem of computing the wet facets and performing vector operations is amenable to parallelization as results of computation taking place at each cluster are independent of each other. After performing the clustering, computation time can further be reduced by using parallelization by changing the sequential code to parallel, without changing the error introduced.

Computations performed on each facet are not dependent on the results from other facets. Thus, we can spawn a multiple number of threads based on available computing resources to balance the computation over each processor.

We performed the tests on Intel's Quad processors and employed four threads with dynamic scheduling. It should be noted that the computer we used for testing the parallel version of the code is the same as explained in the earlier section. We chose four threads with dynamic scheduling because the performance was best with four threads working in a numerical

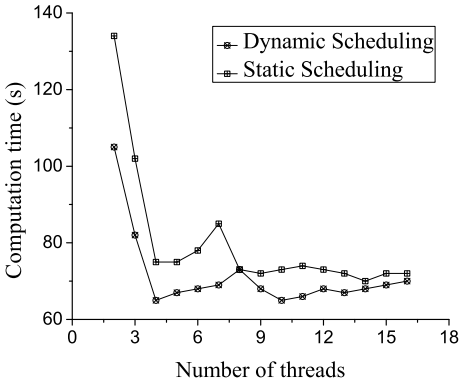


Fig. 5: *Variation of computation time with number of threads.*

experiment we performed. We ran a simulation of 1500 time steps with consecutively 2 to 16 threads and the result is shown in Figure 5. Figure 5 shows that the dynamic scheduling, in which jobs are dynamically allocated to free threads performs better than static scheduling in which the jobs are preallocated to the threads and even if a thread finishes first, it waits for others to finish. The performance was best when four threads were chosen. We used OpenMP to perform parallelization [7]. In the OpenMP based approach, preprocessor directives are placed before each looping statement and the OpenMP optimizes the assignment and scheduling of the tasks to the threads internally.

For the same example from the previous section we ran the parallel version of the code on top of clustering based simplification. With the parallel version of the code the average computation time was found to be 67.0*s*. Thus, the average computation time was reduced by a factor of $\frac{1191.40}{67.00} = 17.78$ with an average force error of 2.88% using the parallelization and clustering based approach. It can be noted that the computation time reduced by a factor of $\frac{216.80}{67.00} = 3.24$ with respect to that of clustering and there is no change in the average force error after parallelization. The average time taken for one time step, by using parallelization along with clustering approach is about $\frac{67.00}{1500} = 0.045$ seconds, which can be used to simulate the USSV motion in real time for time step of length 0.07 second.

8 Temporal Coherence

In the clustering based approach, we utilized the fact that the dynamic pressure values do not change significantly in the close spatial vicinity to simplify the computations. Another fact that can be used to further simplify the computations is that the force due to the ocean waves on the USSV does not change significantly in a short period of time. This simple but useful fact enables us to skip the computation of the list of the wet facets and ocean wave force in some time steps and reuse the list of the wet facets and force due to the ocean wave obtained in the previous time step. This leads to a significant reduction of the computation time with the introduction of some error. The basis of skipping the computation of the list of the wet facets and ocean wave force is the similarity of the instantaneous ocean wave height-field acting on the USSV in the consecutive time step. In other words, if the ocean wave in the proximity of the USSV does not change significantly in a given period of time, we can safely assume that the force acting on the USSV due to the ocean wave also does not change significantly and the force computed in the previous time step can be reused.

8.1 Temporal Coherence Algorithm

In this section, we present the algorithm implementing the idea of temporal coherence explained before. In order to explain the algorithm, we present the following two definitions first.

Definiton 3 Let, B be the bounding box of the USSV and a rectangle R_B be the projection of B on the XY plane. Let Λ denote uniform grid of size $m \times n$ on R_B.

We define the instantaneous ocean wave height-field for a moving USSV as a vector $\vec{G} \in \mathbb{R}^{mn}$, such that the elements of $\vec{G}$ are the ordered elevations of the ocean wave at the mn grid points on Λ.

Definiton 4 For a pair of ocean wave height-fields $\vec{G_1}$ and $\vec{G_2}$, we define the distance d_{hf} between $\vec{G_1}$ and $\vec{G_2}$ as the following second order norm.

$$d_{hf} = \|\vec{G_1} - \vec{G_2}\| \tag{27}$$

Algorithm - Temporal Coherence

Input: USSV model, initial conditions and tolerance τ
Output: Velocity and pose of the USSV at all time steps
Steps:

i In the first time step, i.e. when the time is zero, determine the list of wet facets $S_{B,P}$ and the force due to the ocean wave $F_{W,P}$ on the USSV as described in the previous sections. Also determine the instantaneous ocean wave height-field $\vec{G}_i$. Solve Equation 9, by setting $F_W = F_{W,P}$ and $S_B = S_{B,P}$ and advance the time step.
ii Determine the instantaneous height-field $\vec{G}$. Determine the distance d_{hf} between $\vec{G}_i$ and $\vec{G}$ using Equation 27. If $d_{hf} \geq \tau$ then compute the instantaneous list of wet facets and ocean wave force and store (by overwriting) in $S_{B,P}$ and $F_{W,P}$ respectively. Set $F_W = F_{W,P}$ and $S_B = S_{B,P}$ and solve Equation 9 and advance the time step.
iii Repeat step (ii) until the simulation is terminated.

We applied the above algorithm on top of the clustering and parallel version of the code as described in the previous sections. We chose $m = 3, n = 5$ and $\tau = 0.10$ and performed a simulation of the USSV model shown in Figure 3b under the action of 5 different wave directions as described in the previous sections. The average computation time for 1500 time steps (of length 0.07 s) for the test case is about 41.80 seconds.

The average computation time reduced by $\frac{1191.40}{41.80} = 28.50$ with an introduction of an average force error of 5.8% with respect to the baseline computation using the approaches based on clustering, parallelization, and temporal coherence together. The average time for computing one time step (of length 0.07 s) is about 0.028 s after performing the simplification based on the temporal coherence on top of the clustering and parallelization based simplification.

8.2 Effect of Wave Tolerance on Error and Computation Time

In this section, we derive the dependence of the computation time and error introduced on the chosen wave tolerance and present numerical validation of the derived relationship. Let us suppose that the component of ocean wave with highest amplitude (say A_m has the frequency of ω_m. The component of the ocean wave with the highest amplitude has the largest affect on the variation of the force on the USSV. A wave component with amplitude A_m and the frequency ω_m takes $\frac{2\tau\pi}{A_m\omega_m}$ time for a variation of τ in water level at a given point. Also the wave has a time period given by $\frac{2\pi}{\omega_m}$. This implies that if we ignore the computation of the list of the wet facets (T_w) and ocean wave force (T_p) for the time when the variation of the wave is less than τ, the amount of computing time is given by Equation 28.

$$T = T_v + C(1 - \frac{\tau}{A_m})(T_w + T_p) \tag{28}$$

The terms C and T_v are explained in Equation 18.

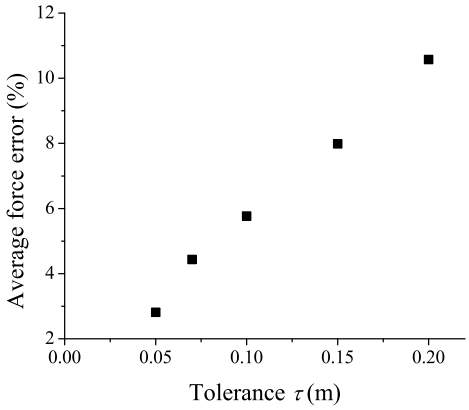
In order to estimate the error in the force calculation due to the approximation introduced in this section, let us consider the variation of τ in the wave height at each point on the boundary of the USSV wet surface. If the perimeter of the curve obtained by the intersection of wave surface and the USSV surface is denoted by P, the maximum variation in force is given as $\Phi_m P\tau$, where Φ_m is the maximum possible dynamic pressure head. Thus the error in the force computation can be bounded by using Equation 29.

$$\|\vec{e}_{force}\| < \Phi_m P\tau \tag{29}$$

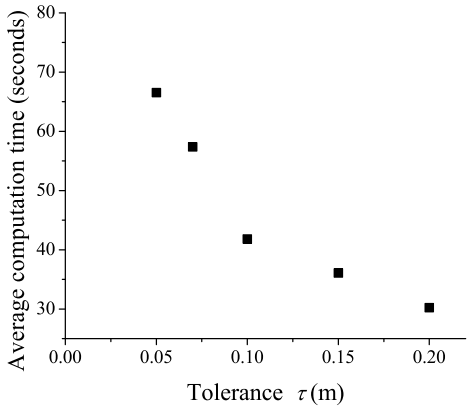
Equation 29 gives us the variation of the error in the force estimation due to the approximation introduced in this section as linear in terms of the tolerance.

In order to verify the above derived relationships, we performed numerical simulations for five different values of τ for the cluster size $C = 45$ and $\varepsilon = 10^{-3}$ for five different wave directions. The variation of the average value of the force magnitude and computation time over five different wave directions with the tolerance τ is shown in Figure 6. The choice of tolerance dictates the reduction in the computation time and an increase in the average error in the force computation. This is because the tolerance τ is the magnitude of the difference between the wave surface experienced by the USSV at a time step and the previous time step which can be ignored to avoid the computation of the ocean wave force without introducing significant error.

In summary, the tolerance can be used as an approximation parameter similarly as the cluster count explained in the previous section to control the desired gain in the computation time and the error introduced due to the temporal coherence based simplification.



(a) *Variation of average force error with tolerance.*



(b) *Variation of average computation time with tolerance.*

Fig. 6: *Variation of average error and average computation time with tolerance. (*1500 *time steps,* $C = 45$*, and* $\varepsilon = 10^{-3}$*).*

9 Conclusions

We developed a virtual environment for a six degrees of freedom simulation of an unmanned sea surface vehicles. We obtained real time performance of the simulator using a simplification approach based on the clustering, temporal coherence, and parallelization approach to perform a physics preserving model simplification for the potential flow based six degrees of freedom, time domain simulation of the USSV. The average computation time was reduced by a factor of about 28.50 introducing an average error of about 5.8% with respect to the baseline computations, using the simplification techniques described in this paper. The approach can take care of any arbitrary hull geometry as long as it can be expressed as a polygonal model. We established theoretically that the time taken for computing the forces increases linearly with the increasing cluster count and reduces linearly with the increasing tolerance, whereas the error introduced reduces with the cluster count in inverse square root fashion and increases with the tolerance in a linear fashion. The numerical simulation results holds well with these theoretical results. Cluster count and tolerance can thus be used as a a high level approximation parameter in the simplification scheme to control the computation time to achieve a desired level of accuracy.

The potential flow assumption about inviscid fluid flow holds well as the boundary layer phenomenon is only observed in the close vicinity of the hull surface. Nevertheless, the phenomenon related to wave breaking, wake and turbulence cannot be taken care of using the potential flow theory alone. We accounted for these terms by introducing the damping and the added mass matrix in the dynamics model. The damping and added mass matrices can be determined by performing parameter identification for a given USSV model under given sea states by experiments and trials. A more general modeling however is required to cover wide range of sea states. Also, potential flow theory is based on displacement physics. The hydroplaning phenomenon observed in the boats moving at very high speeds can not be modeled using the potential flow theory and requires use of a fluid flow model based on the planing physics. The simplification approach based on clustering and temporal coherence is however, independent of any fluid flow model used and can be employed in any kind of fluid flow to improve the speed of computation.

Achnowledgements

This research has been supported by Office of Naval Research and NSF Grant CMMI-0727380. Opinions expressed in this paper are those of the authors and do not necessarily reflect opinions of the sponsors.

References

[1] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469 – 483, 2009.

[2] C. Batty, F. Bertails, and R. Bridson. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph.*, 26(3):100, 2007.

[3] R. Beck and A. Reed. Modern seakeeping computations for ships. In *Twenty-Third Symposium on Naval Hydrodynamics*. Naval Studies Board (NSB), 2001.

[4] M. Becker, H. Tessendorf, and M. Teschner. Direct forcing for lagrangian rigid-fluid coupling. *Visualization and Computer Graphics, IEEE Transactions on*, 15(3):493 –503, may-june 2009.

[5] M. Carlson, P. J. Mucha, and G. Turk. Rigid fluid: animating the interplay between rigid bodies and fluid. *ACM Trans. Graph.*, 23(3):377–384, 2004.

[6] P. M. Carrica, R. V. Wilson, R. W. Noack, and F. Stern. Ship motions using single-phase level set with dynamic overset grids. *Computers & Fluids*, 36(9):1415 – 1433, 2007.

[7] B. Chapman, G. Jost, and A. R. V. D. Pas. *Using OpenMP Portable Shared Memory Parallel Programming*. The MIT Press, 2008.

[8] K.D. Do, Z.P. Jiang, and J. Pan. Underactuated ship global tracking under relaxed conditions. *IEEE Transactions on Automatic Control*, 47(9):1529–1536, Sep 2002.

[9] T. I. Fossen. *Guidance and control of ocean vehicles,*. Wiley, Chicester, England, 1994.

[10] T. I. Fossen and Ø. N. Smogeli. Nonlinear time-domain strip theory formulation for low-speed manoeuvering and station-keeping. *Modeling, Identification and Control*, 25(4):201–221, 2004.

[11] J. J. Gorski. Present state of numerical ship hydrodynamics and validation experiments. *Journal of Offshore Mechanics and Arctic Engineering*, 124(2):74–80, 2002.

[12] P. Guevel and J. Bougis. Ship motions with forward speed in infinite depth. In *Int. Shipbuilding Progress*, 1982.

[13] S. K. Gupta, D. K. Anand, A. D. Thakur, P. Svec, and M. Schwartz. A simulation based framework for discovering planning logic for unmanned surface vehicles. In *ASME Engineering Systems Design and Analysis Conference, July 12-14 2010, Istanbul, Turkey.*, 2010. Accepted for publication.

[14] R. B. Inglis and W. G. Price. A three-dimensional ship motion theorycomparison between theoretical prediction and experimental data of the hydrodynamic coefficients with forward speed. *Transactions Royal Inst. Naval Architecture*, 124:141–157, 1981.

[15] M.R. Katebi, M.J. Grimble, and Y. Zhang. H ∞ robust control design for dynamic ship positioning. *Control Theory and Applications, IEEE Proceedings*, 144(2):110–120, Mar 1997.

[16] K. H. Kim. Simulation of surface ship dynamics using unsteady RANS codes. In *Reduction of Military Vehicle Acquisition Time and Cost through Advanced Modelling and Virtual Simulation, Paris, France*, 2002.

[17] K. H. Kim, J. Gorski, R. Miller, R. Wilson, F. Stern, M. Hyman, and C. Burg. Simulation of surface ship dynamics. In *User Group Conference, 2003. Proceedings*, pages 188–199, June 2003.

[18] P. Krishnamurthy, F. Khorrami, and S. Fujikawa. A modeling framework for six degree-of-freedom control of unmanned sea surface vehicles. In *Proc. and 2005 European Control Conference Decision and Control CDC-ECC '05. 44th IEEE Conference on*, pages 2676–2681, December 12–15, 2005.

[19] E. Lefeber, K.Y. Pettersen, and H. Nijmeijer. Tracking control of an underactuated ship. *Control Systems Technology, IEEE Transactions on*, 11(1):52–61, Jan 2003.

[20] A. Loria, T. I. Fossen, and E. Panteley. A separation principle for dynamic positioning of ships: Theoretical and experimental results. *IEEE Transactions on Control Systems Technology*, 8:332–343, 2000.

[21] F. Mazenc, K. Pettersen, and H. Nijmeijer. Global uniform asymptotic stabilization of an underactuated surface vessel. *Automatic Control, IEEE Transactions on*, 47(10):1759–1762, Oct 2002.

[22] J. N. Newman. *Marine Hydrodynamics*. MIT Press, Cambridge, MA, 1977.

[23] S.P. Singh and D. Sen. A comparative linear and nonlinear ship motion study using 3-D time domain methods. *Ocean Engineering*, 34(13):1863 – 1881, 2007.

[24] G. D. Weymouth, R. V. Wilson, and F. Stern. RANS computational fluid dynamics predictions of pitch and heave ship motions in head seas. *Journal of Ship Research*, 49(18):80–97, June 2005.