Development of naoeFOAM-os-SJTU Solver Based on Overset Grid Techniques for Self-Propulsion of Ship

Zhirong Shen and Decheng Wan

State Key Laboratory of Ocean Engineering,
School of Naval Architecture, Ocean and Civil Engineering,

Shanghai Jiao Tong University

**International Research Exchange Meeting on Ship and Ocean Engineering,
December 20-21, 2013, Osaka University, Japan**

Contents

- Introduction
- Development of Solver Package：

  naoeFOAM-os-SJTU
- Numerical Examples
- Closing Remarks

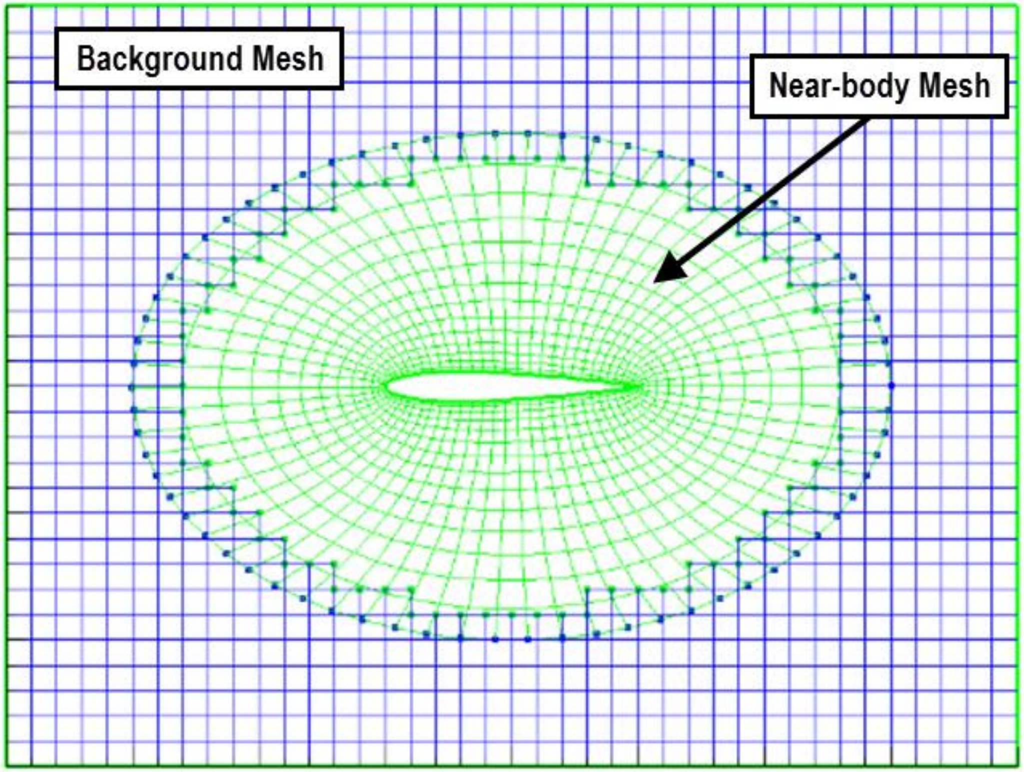
Introduction

- **Seakeeping, Self-propulsion and Maneuvering are still great challenges for computational ship hydrodynamics**
- **Limitations of traditional mesh methodologies**
  - Deforming and sliding grids
- **Advantages of overset grids**
  - Move grids without restriction
  - Include hierarchy of objects , which allows appendages (moving rudder, rotating propeller) move independently with respect to the moving ship

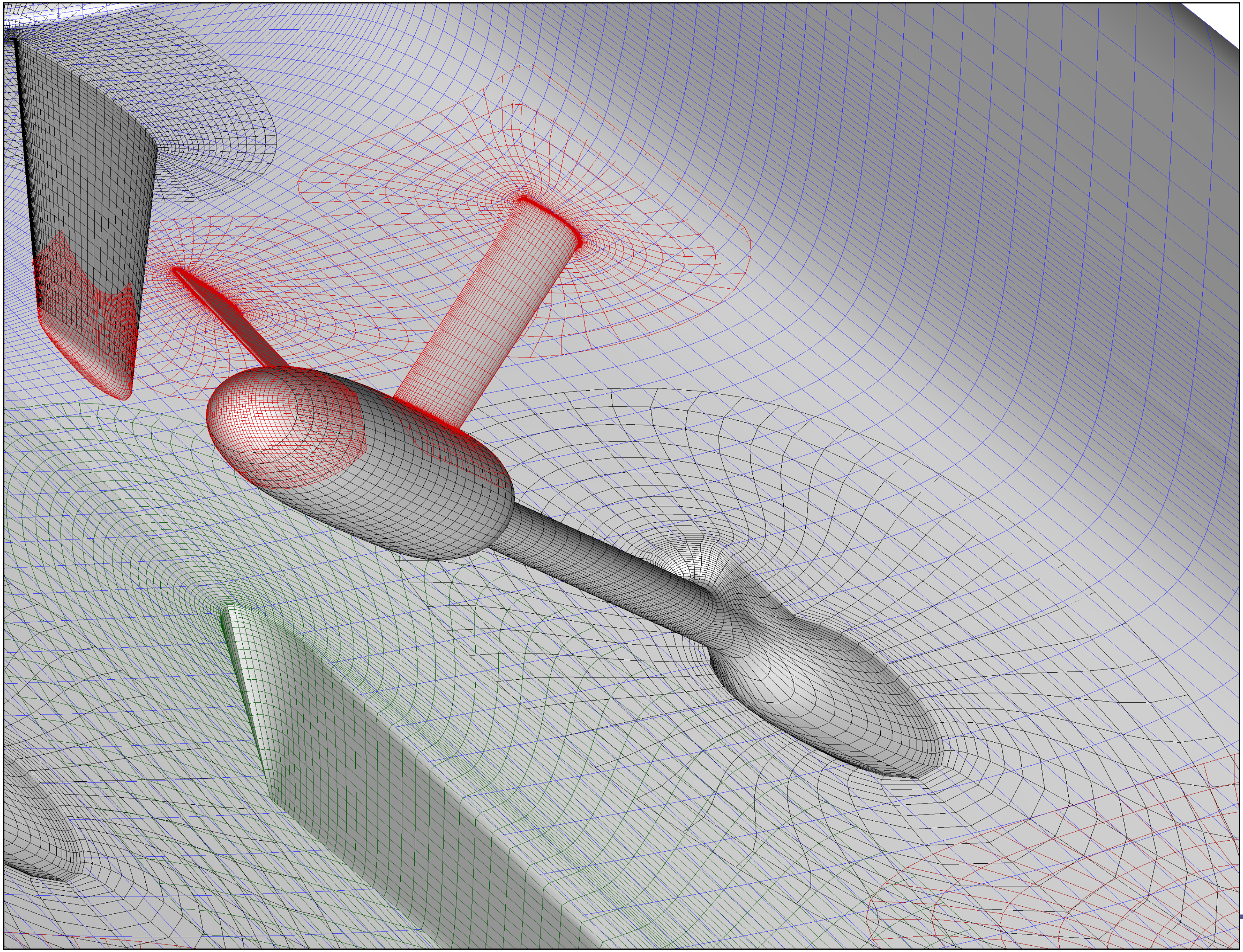
Dynamic overset grids

Overset Grid

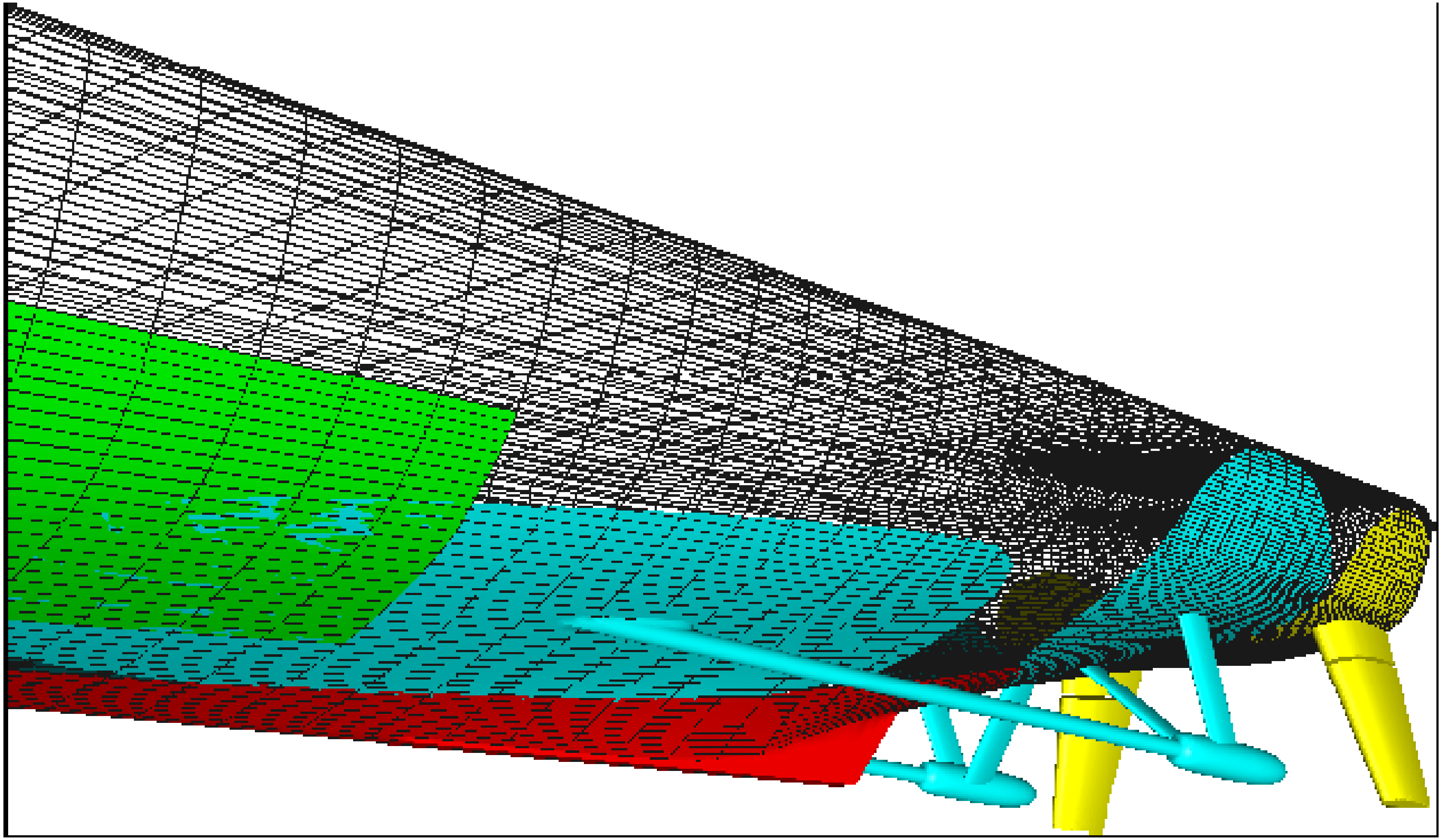
- A body fitted grid can be embedded into a Cartesian background mesh.
- Two grids are mutual independence.
- Body fitted grid can be moved without restriction.
- Two grids build the connectivity through the interpolation coefficients.



Dynamic overset grids



Dynamic overset grids



Development of Solver Package：

naoeFOAM-os-SJTU

Object:

naoe-FOAM-SJTU + Overset

↓

naoeFOAM-os-SJTU

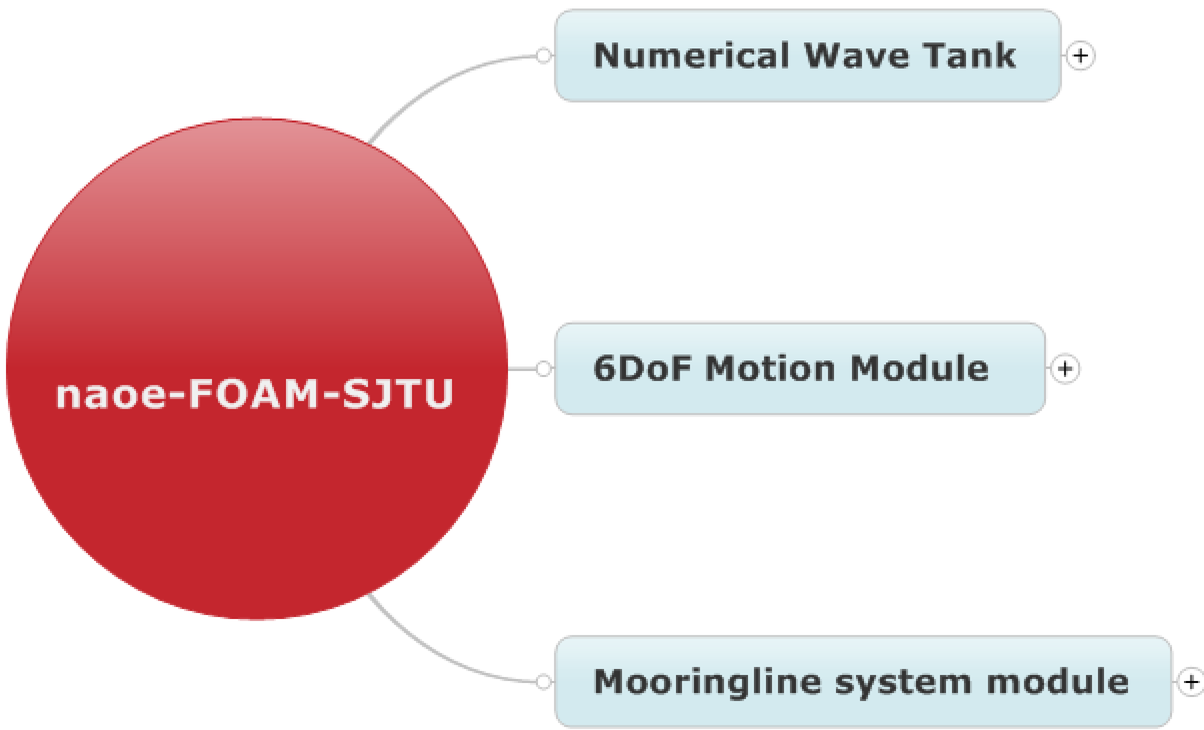
To solve the problem of Self-Propulsion of Ship

Solver Package (naoe-FOAM-SJTU 1.0)

naoe-FOAM-SJTU is a 3D Numerical Marine Basin based on OpenFOAM platform:

- take viscous effect into consideration, including violent flow (high Re flows, breaking waves)
- provide different types of waves (numerical wave generation and absorption)
- study wave(current, wind)-floating structures interaction easily (nonlinear, 6DOF, mooring)

Introduction to naeo-FOAM-SJTU



Functions

Provide wave conditions

Model structures motion

provide mooring force,

keep body steady,

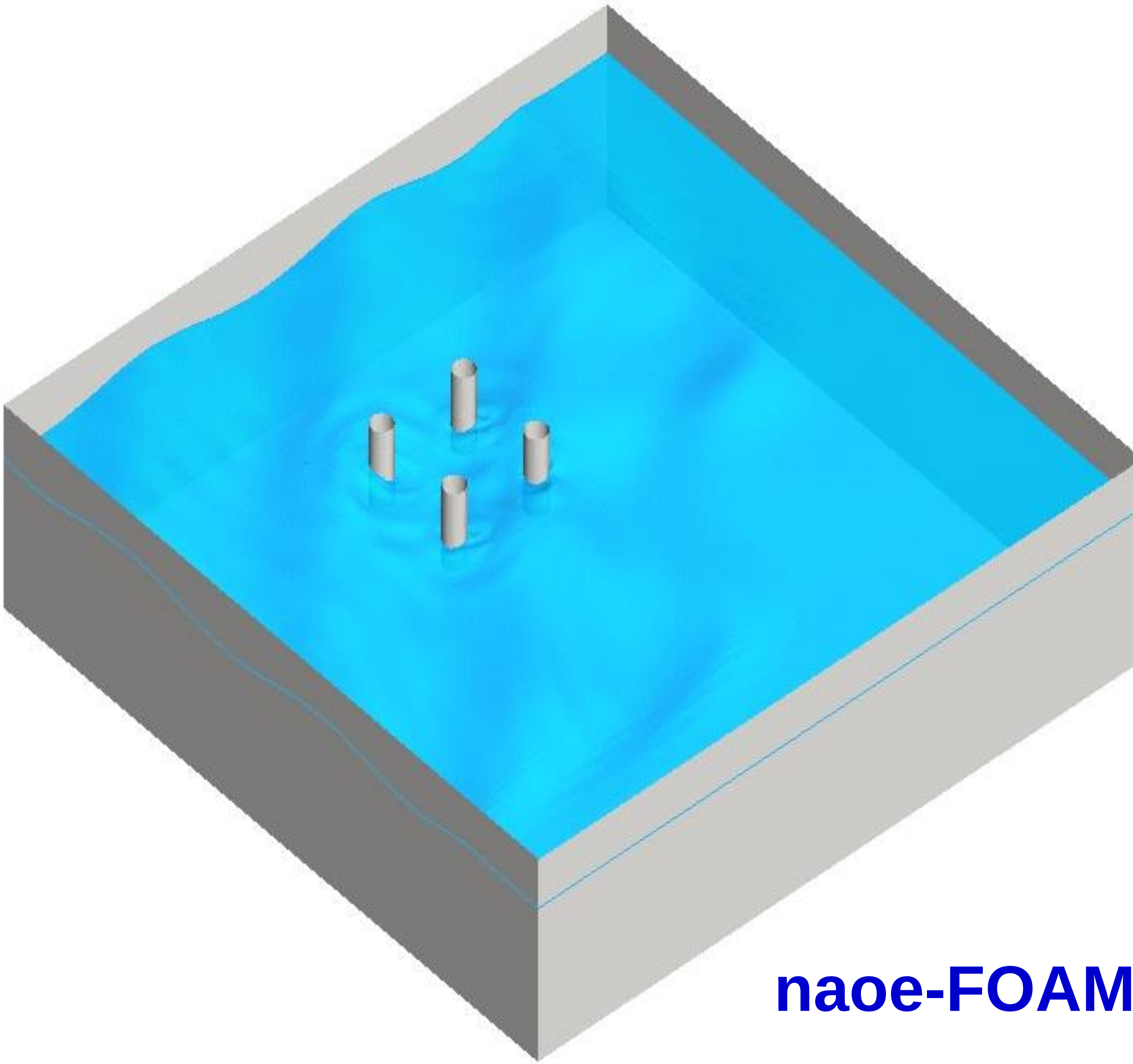
restrain structures motion

3D Focused Waves

focusing wave

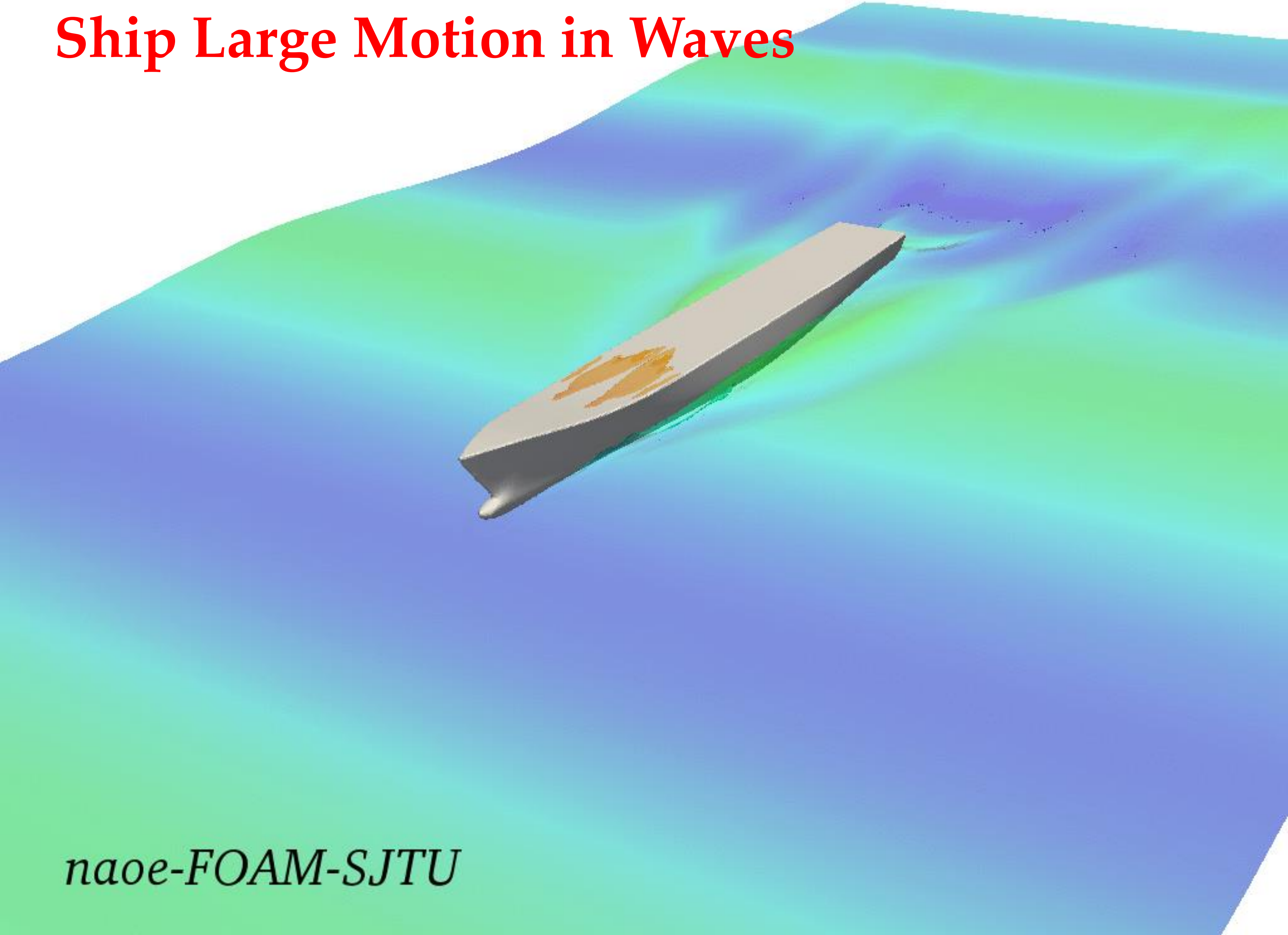
naoe-FOAM-SJTU©

Waves around Multi-Cylinders

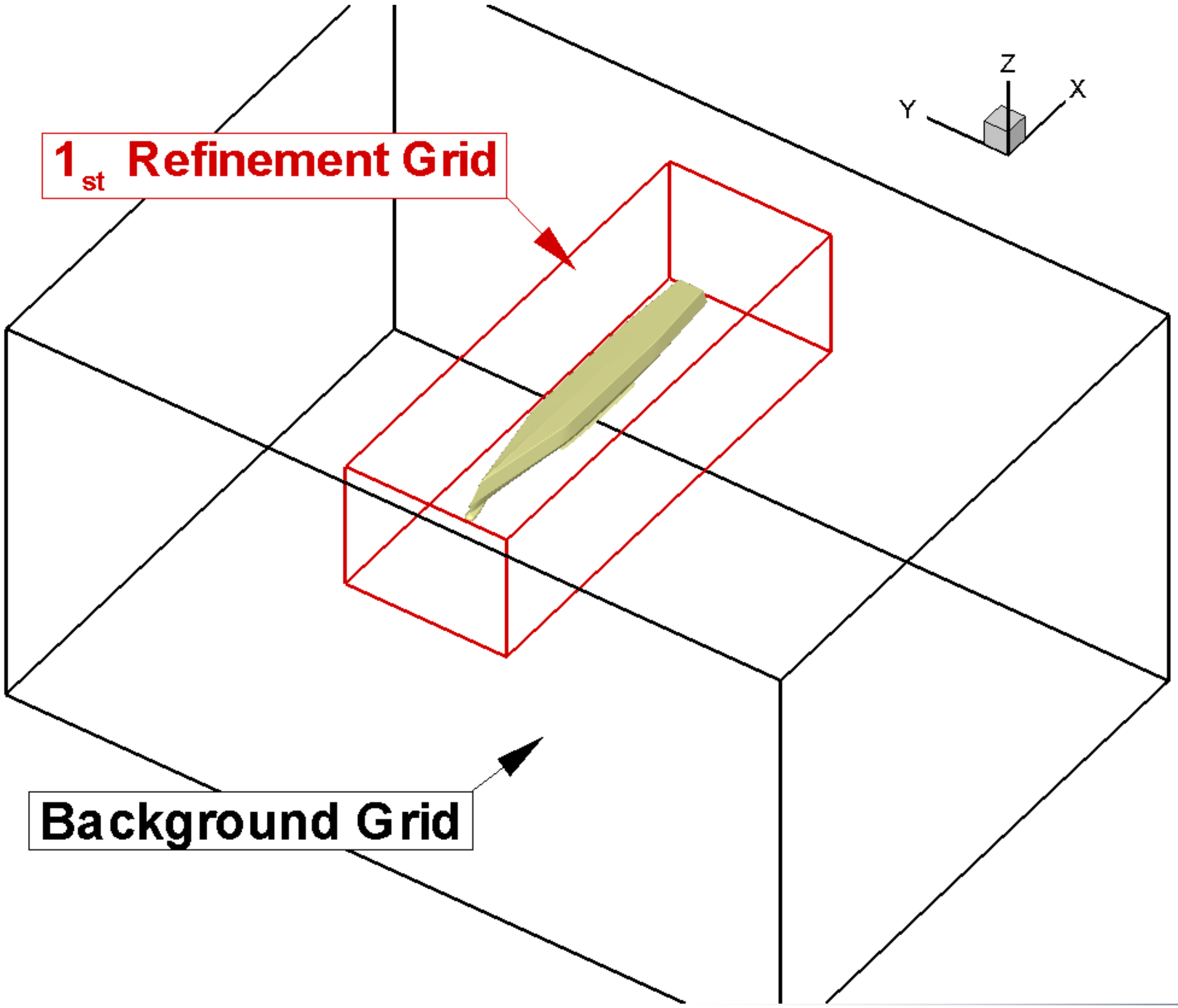


naoe-FOAM-SJTU©

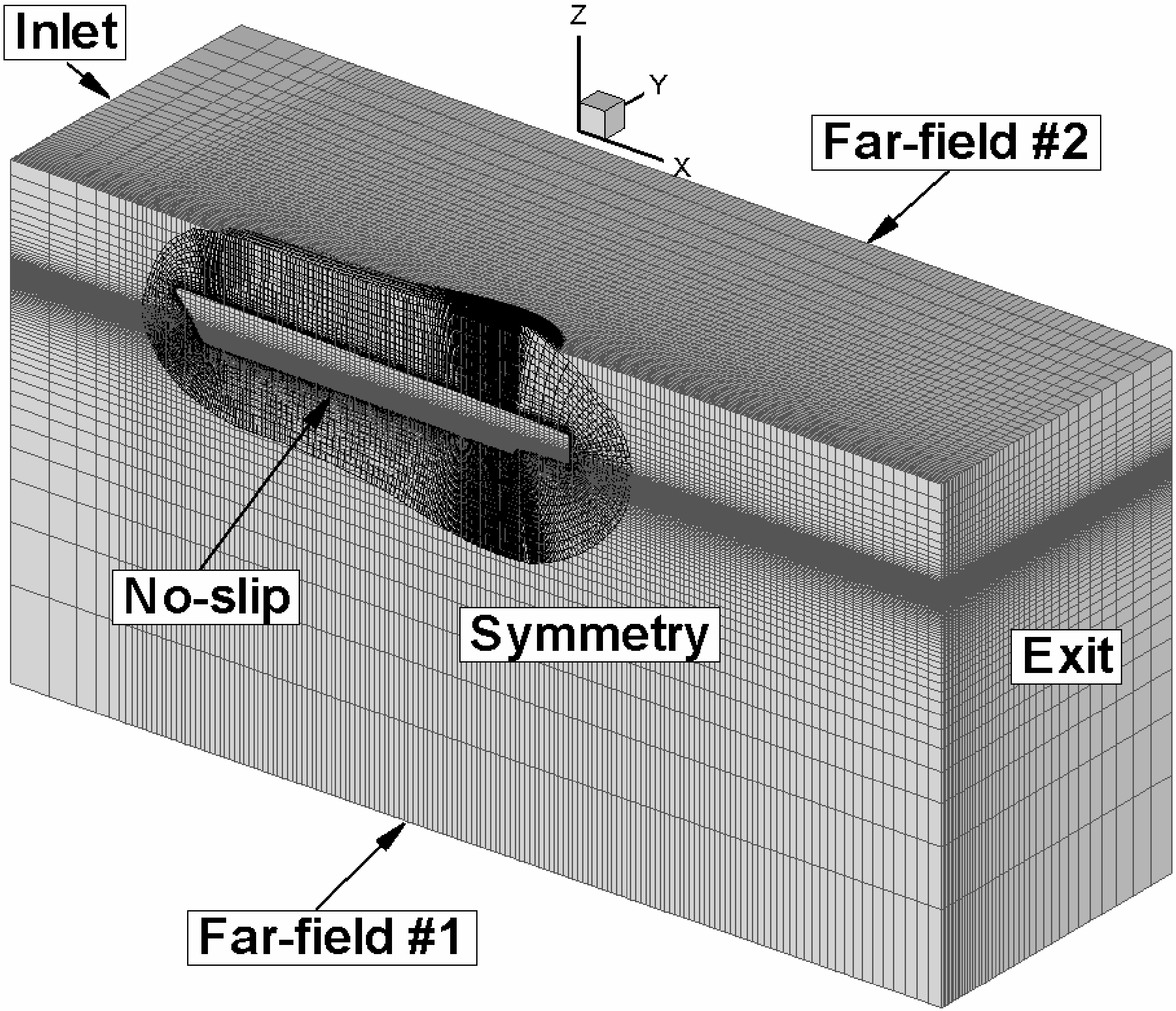
Ship Large Motion in Waves



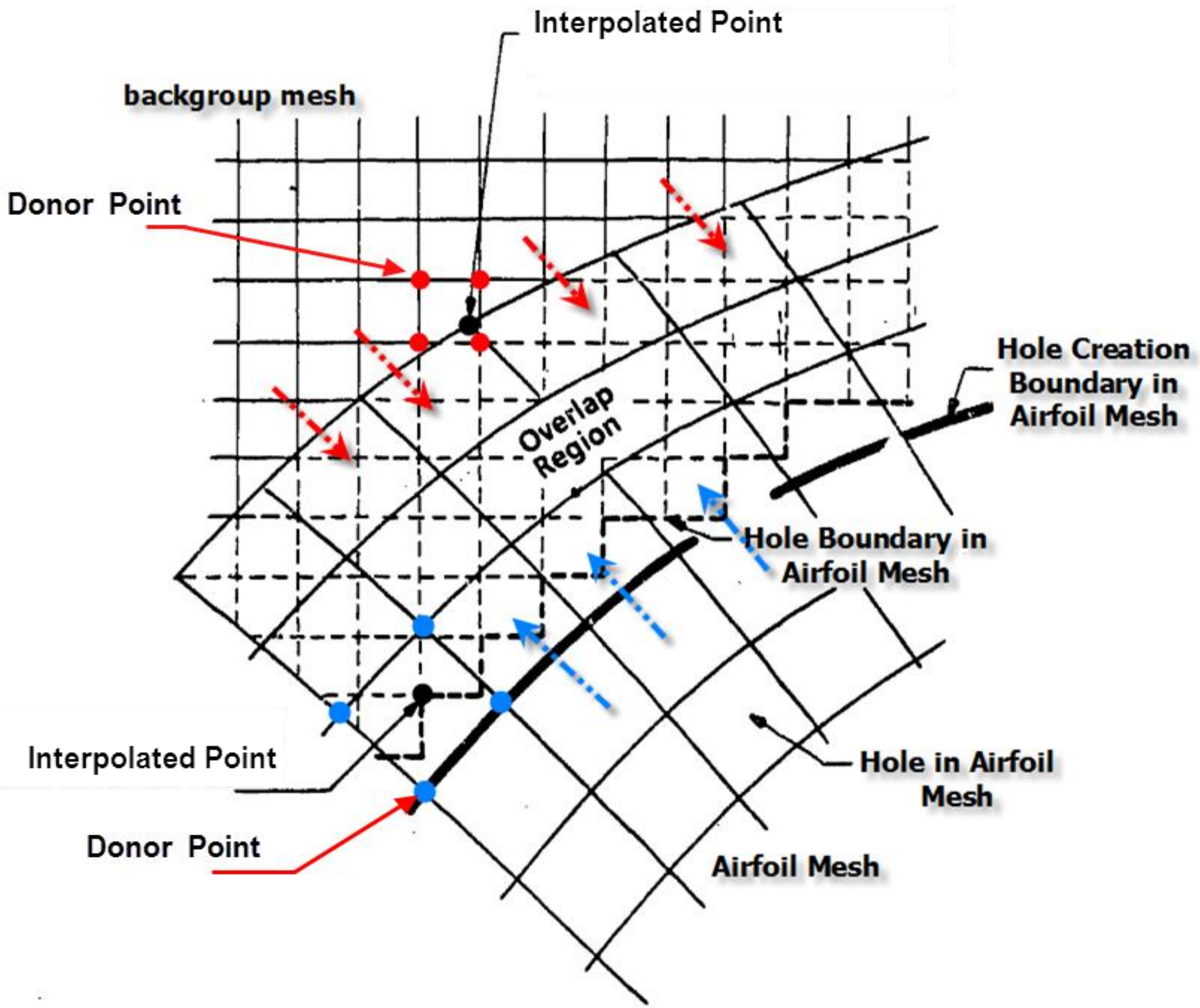
Dynamic overset grids



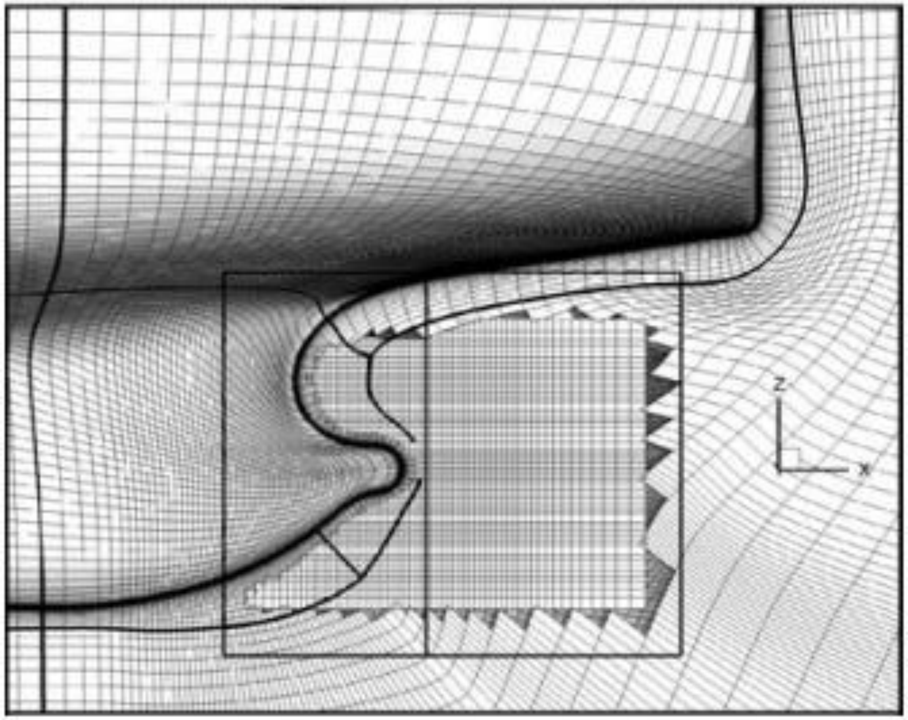
Dynamic overset grids

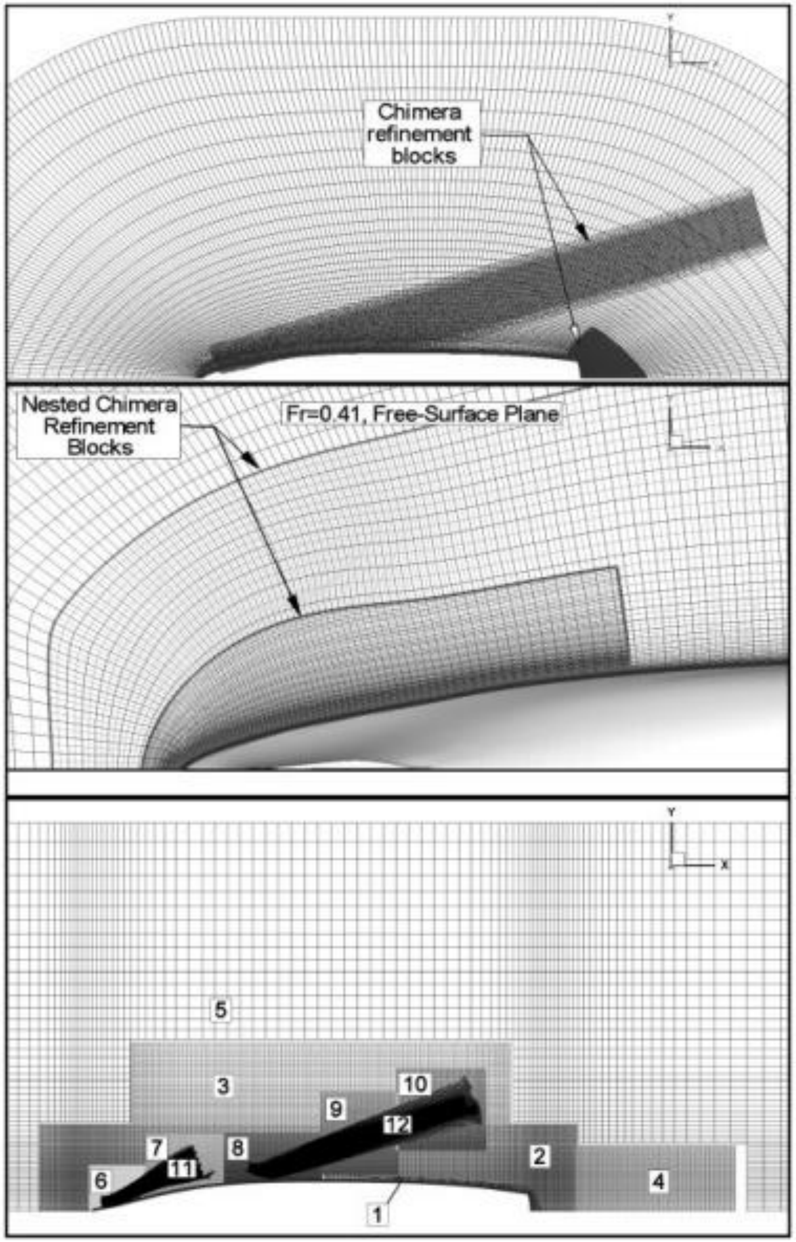


Donor & Interpolated points



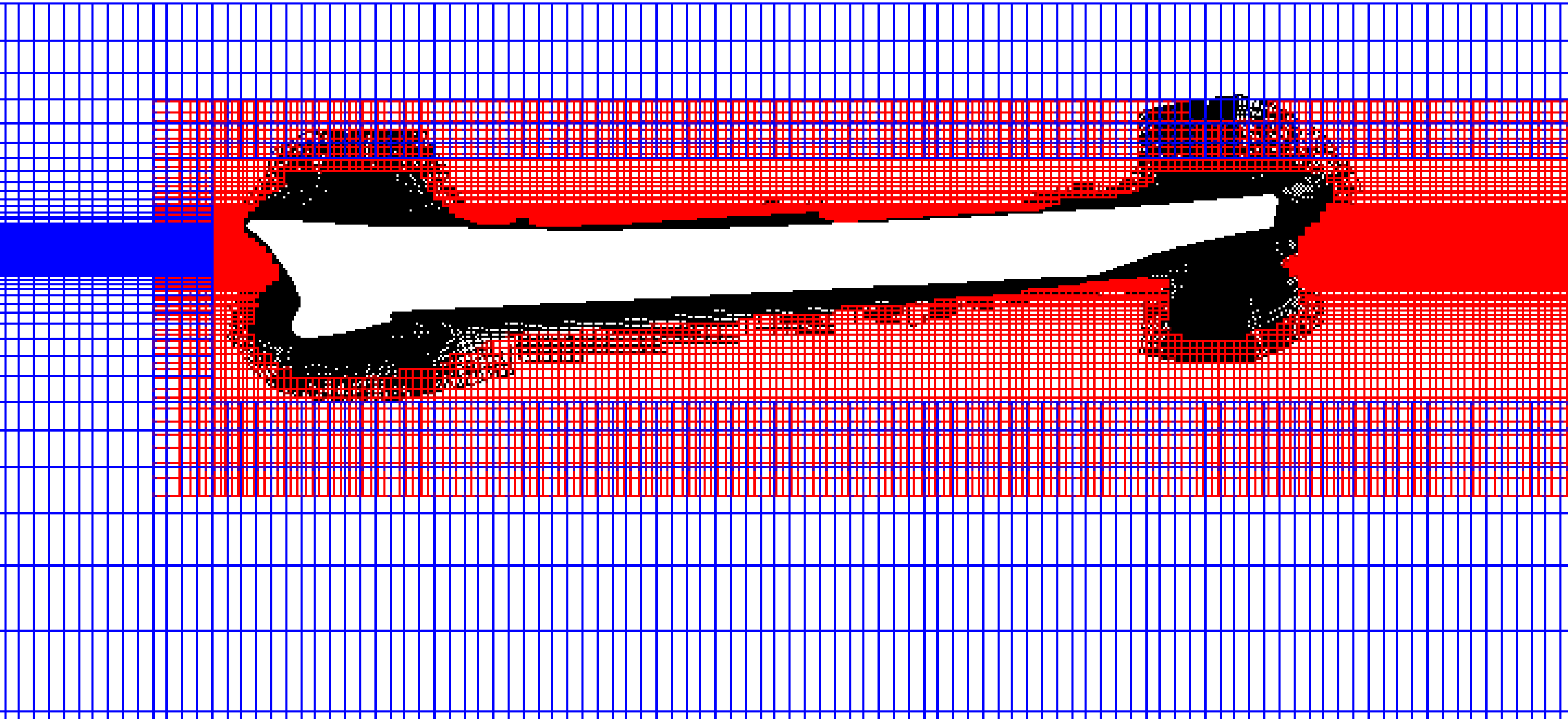
Dynamic overset grids





Dynamic overset grids

Advantages of Overset Method

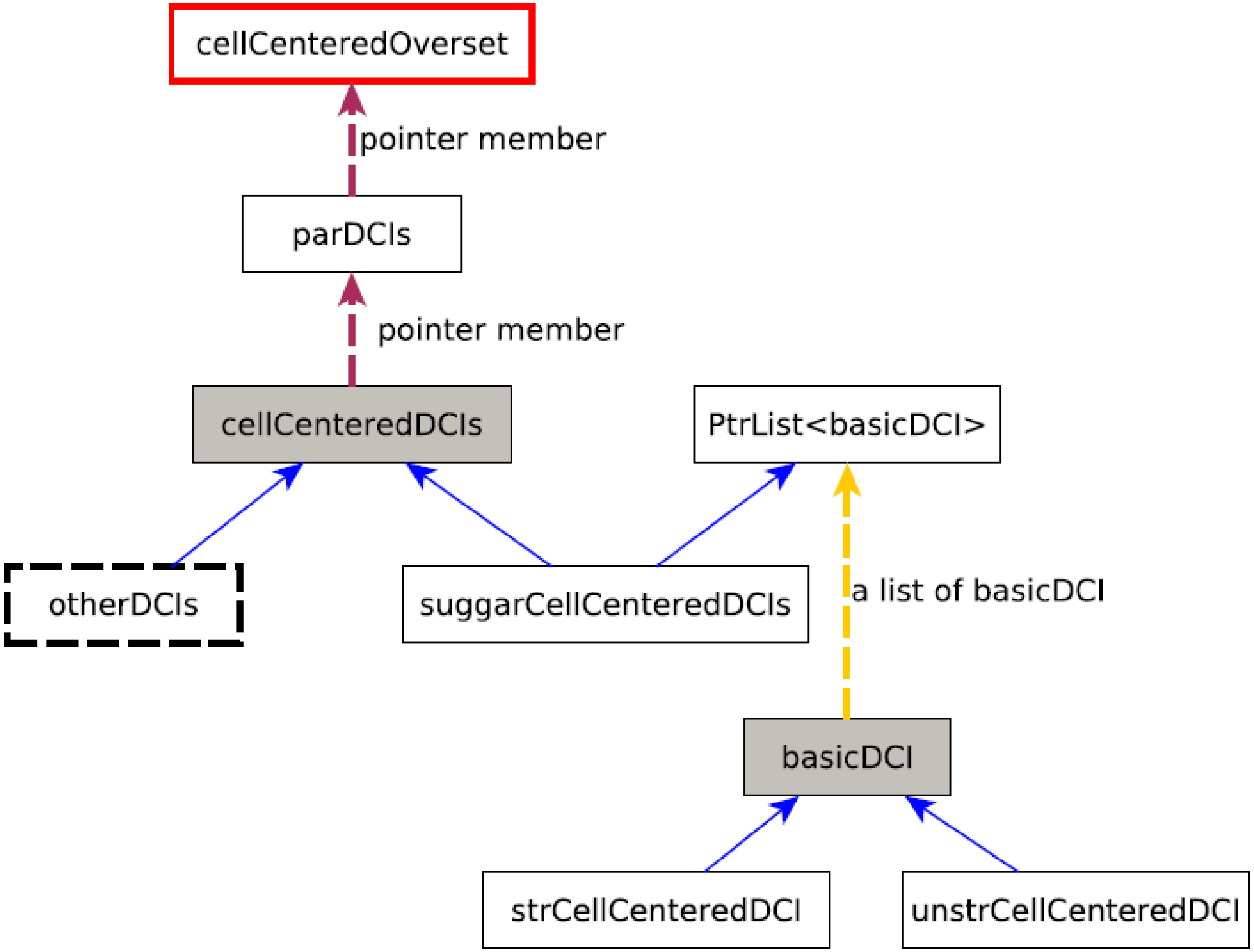


Implement Overset in naoe-FOAM-SJTU

- **Read DCI from overset grid data.**
- **Computed interpolated values from donors.**
- **Solve N-S Equations.**
- **Solve VOF Equation.**
- **Solve Turbulence Equation.**
- **Parallelization.**
- **Validation.**

Code Structure

- `liboverset:` a library makes naoe-FOAM-SJTU capable of overset.



Implementation

- How to implement overset capability in naoe-FOAM-SJTU solver by using `liboverset`?

- Example:
  - An incompressible laminar flow solver: icoFoam
    - Step I: Include two header files:

```
#include "cellCenteredOverset.H“
#include "createOverset.H"
```

Implementation

- Build Matrix:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) - \nabla \cdot (\nu \nabla \mathbf{U}) = -\nabla p$$

```
fvVectorMatrix UEqn
(
    fvm::ddt(U)
    + fvm::div(phi, U)
    - fvm::laplacian(nu, U)
    ==
    - fvc::grad(p)
);
```

Implementation

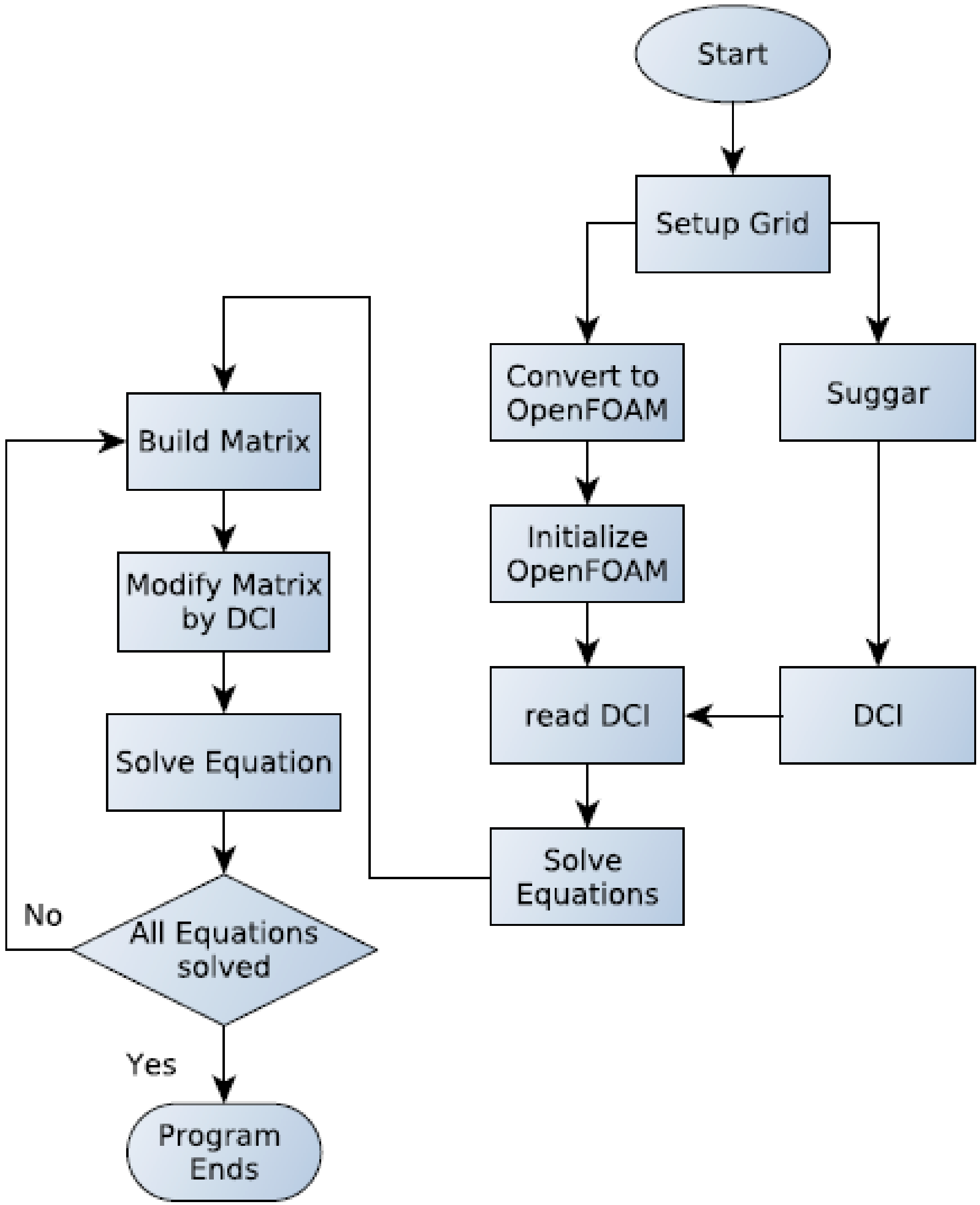
- Step II: Modify Matrix and solve:

```
overset.updateFvMatrix<vector>(UEqn);
UEqn.solve();
```

- Step III: Solve other equation (e.g. pressure)

```
overset.updateFvMatrix<scalar>(pEqn);
pEqn.solve();
```

Flow-Chart

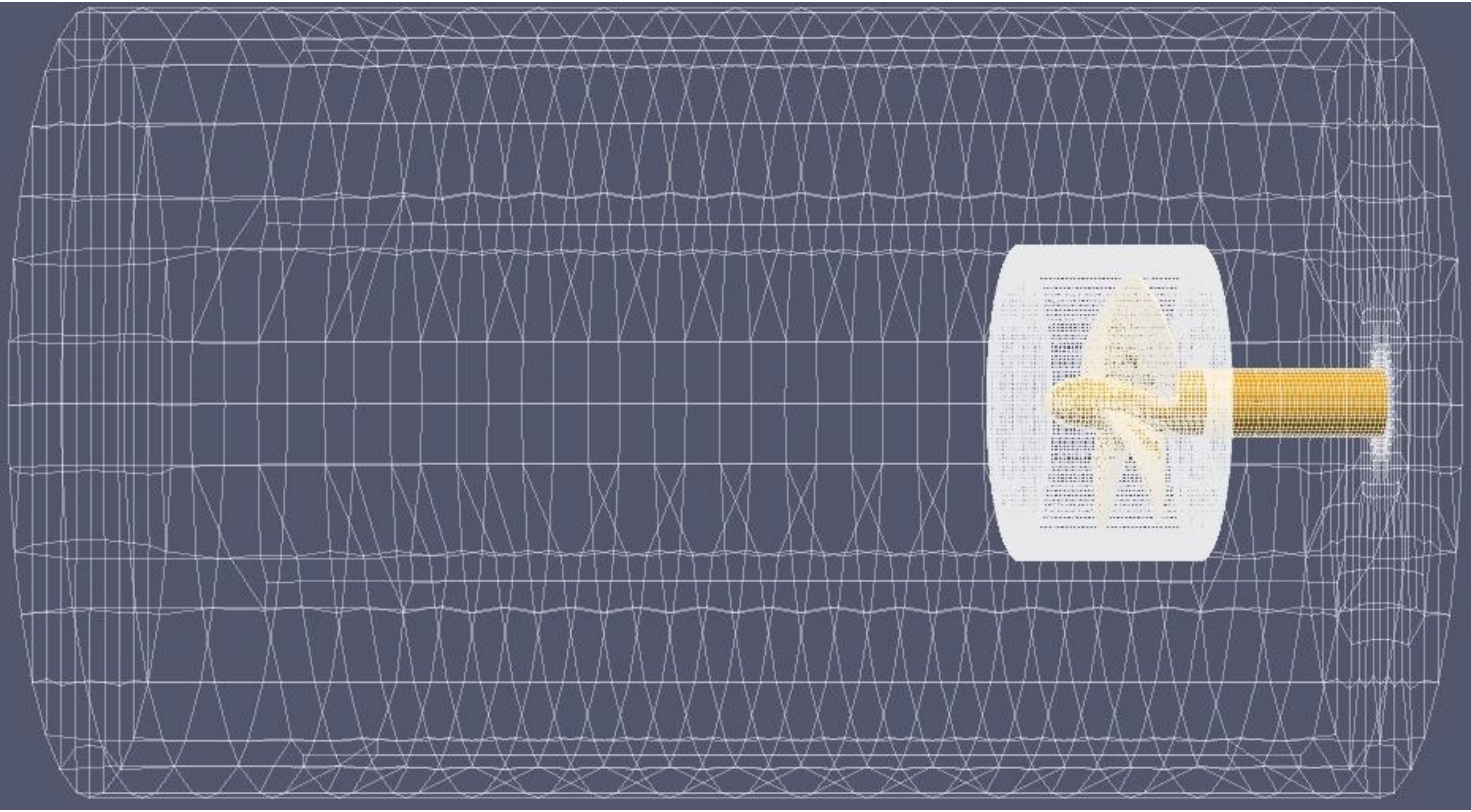

Start
Setup Grid
Convert to OpenFOAM
Suggar
Initialize OpenFOAM
read DCI
DCI
Solve Equations
Build Matrix
Modify Matrix by DCI
Solve Equation
All Equations solved
No
Yes
Program Ends

Numerical Experiments

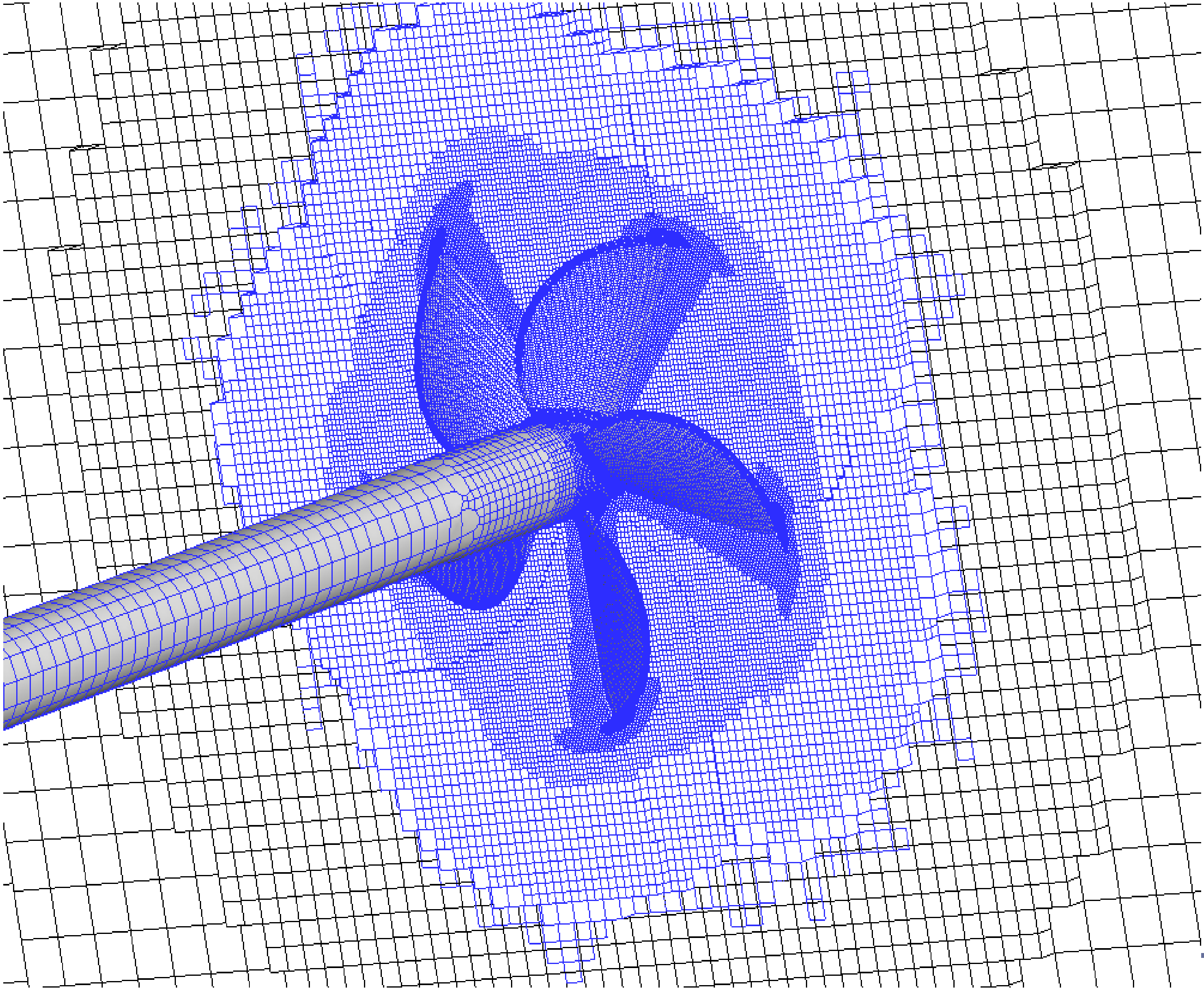
- A full CFD prediction of Self-propulsion characteristics
  - Open water curves
  - Towed condition
  - Self-propulsion condition

Propeller Flows and Self-Propulsion of Ship Motion

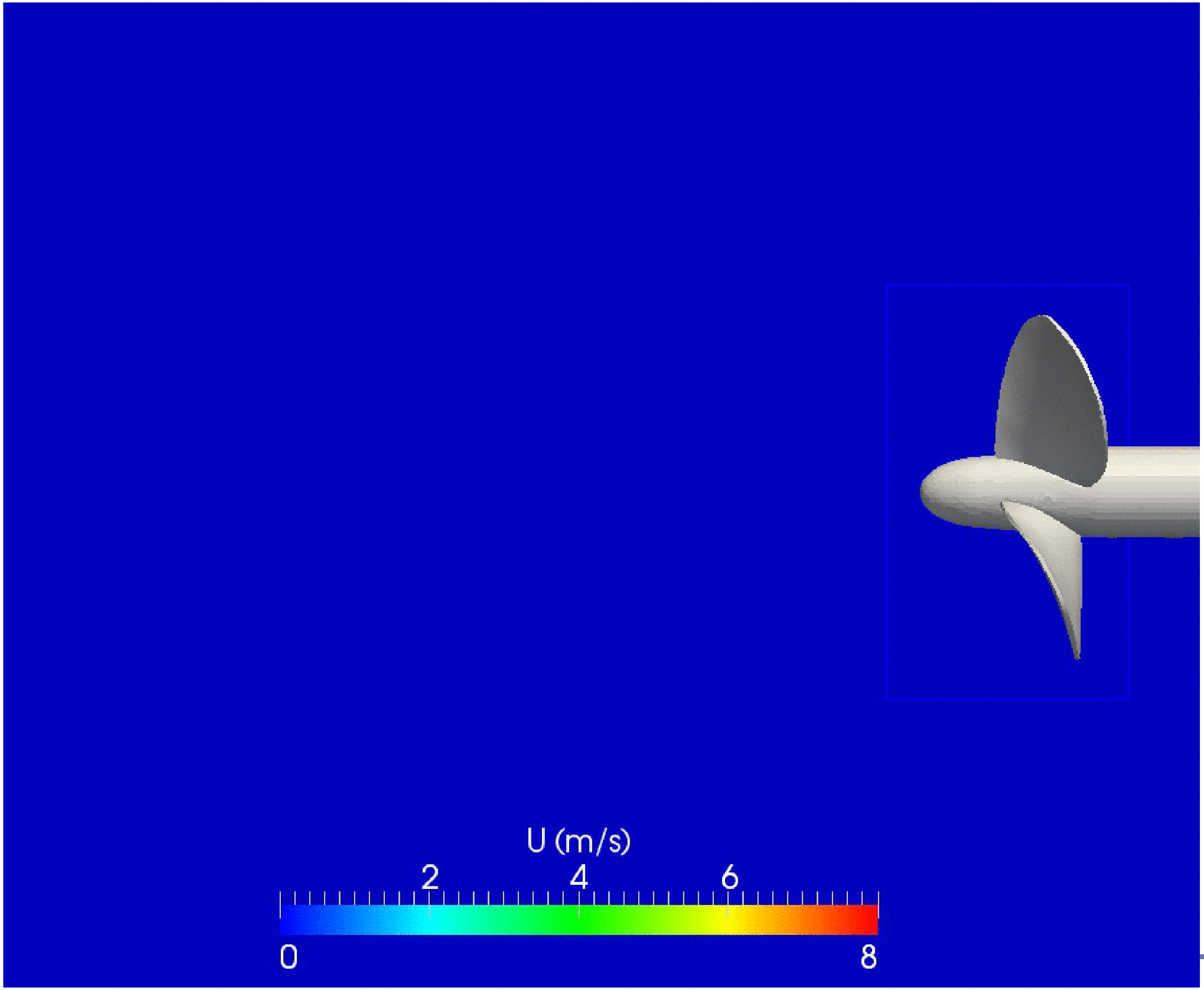
Rotating Propeller in Open Water



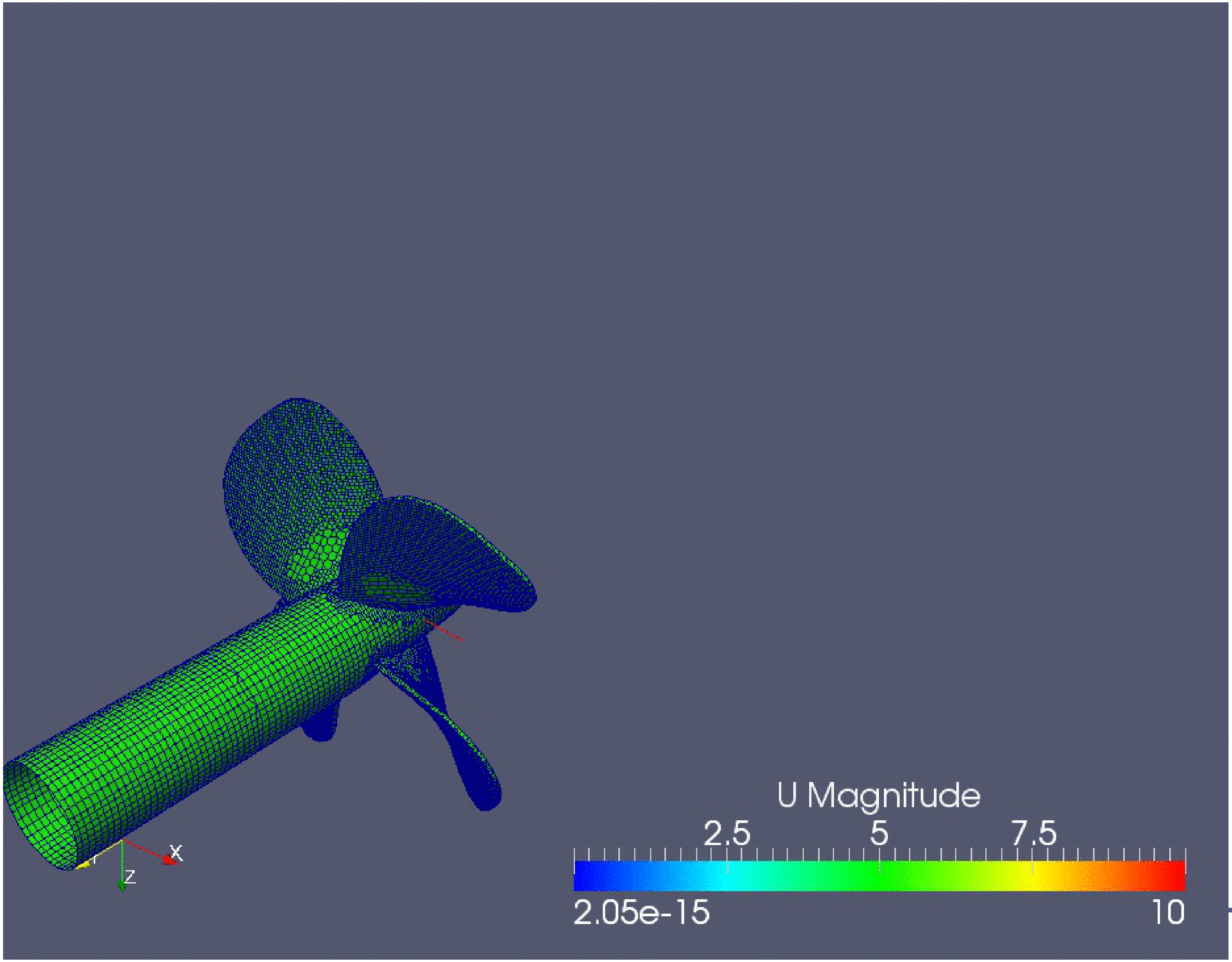
Rotating Propeller in Open Water



Rotating Propeller in Open Water

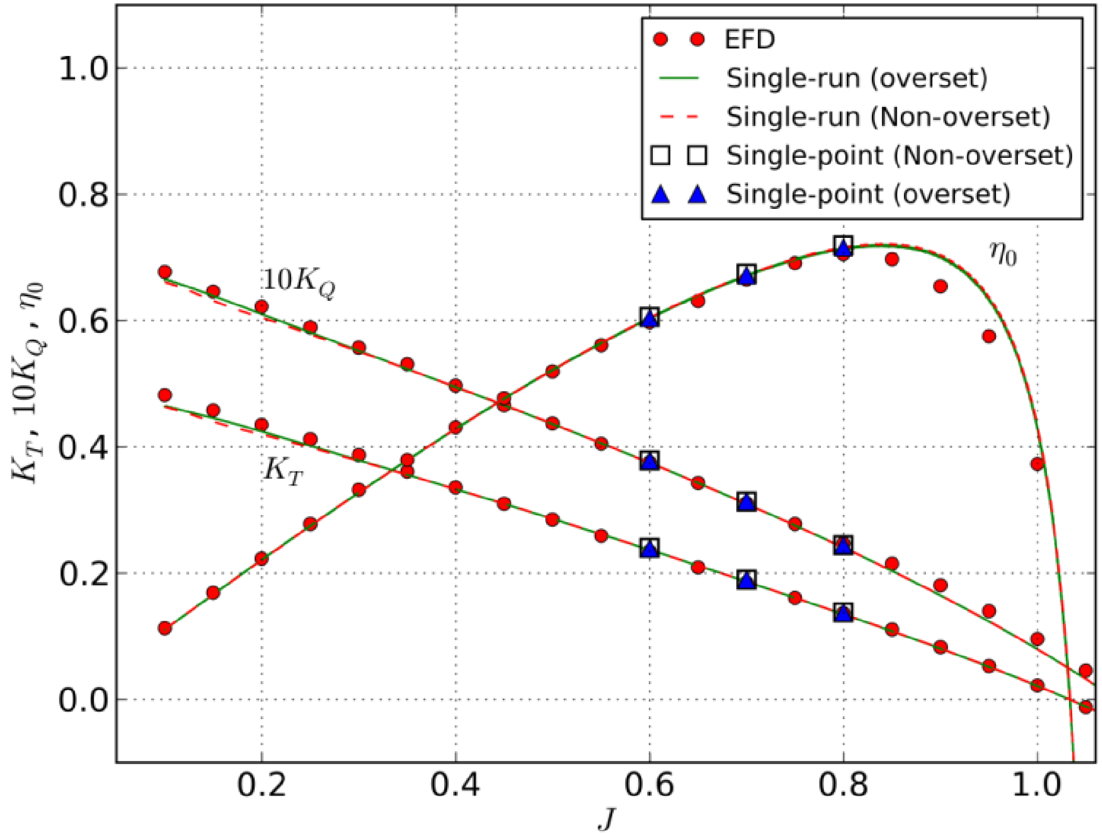


Rotating Propeller in Open Water



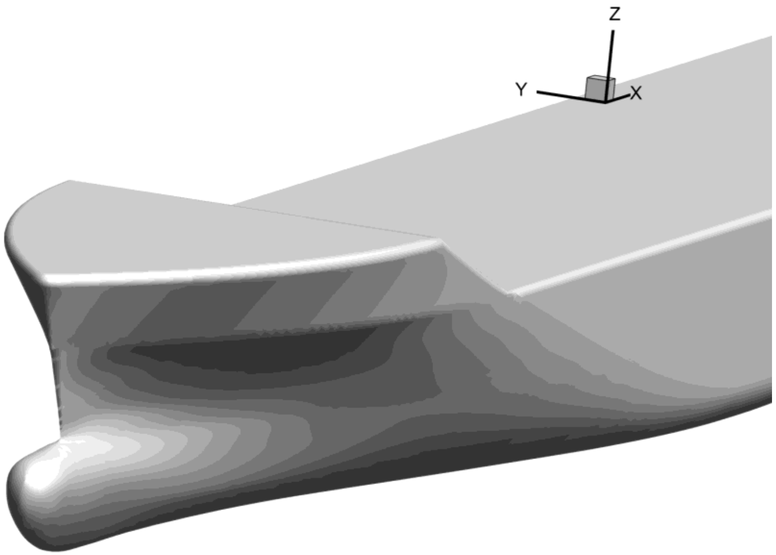
Rotating Propeller in Open Water

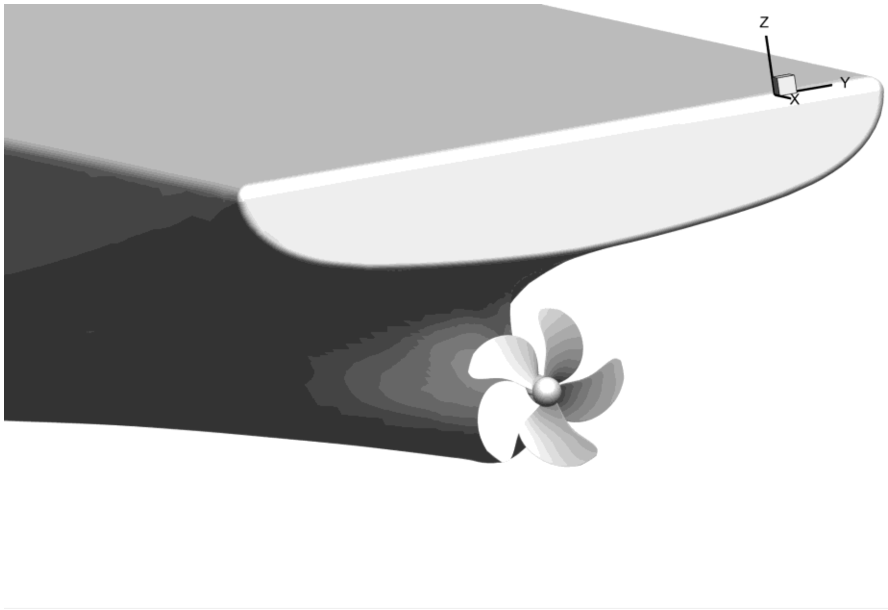
- Single-point:
  - Fixed V_a and RPS.
  - J=0.6, 0.7, 0.8
- Single-run:
  - J = 0.05 ~1.05,
  - Ramp time = 5 s



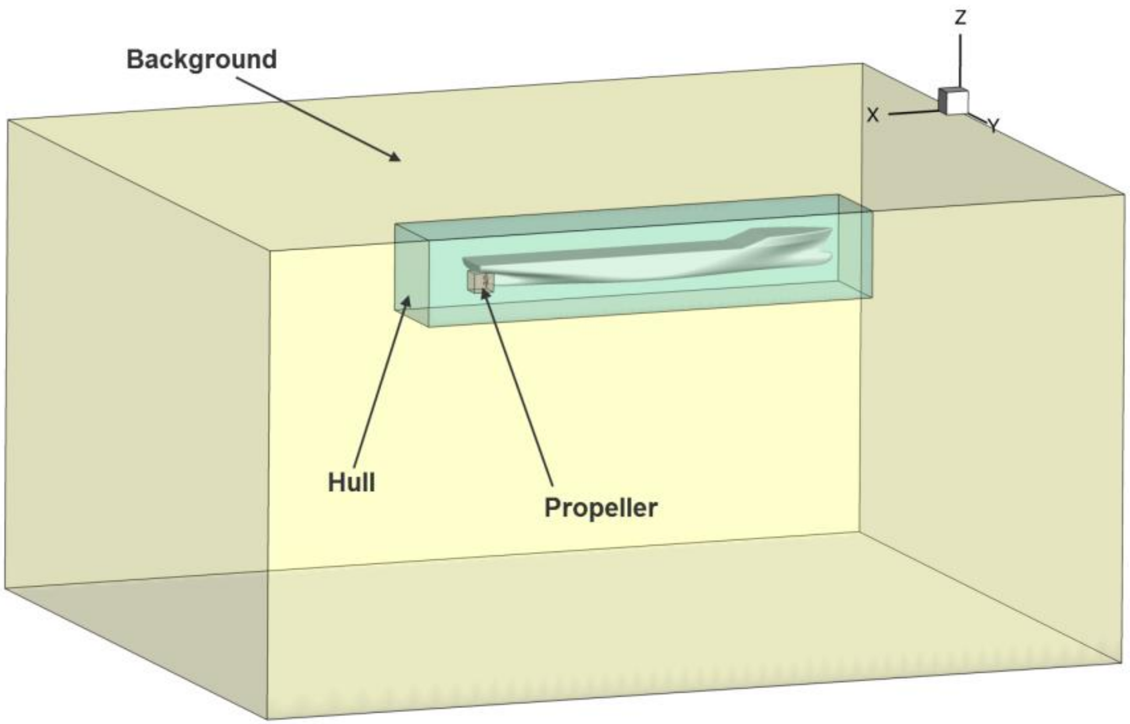
SELF-PROPULSION OF KCS

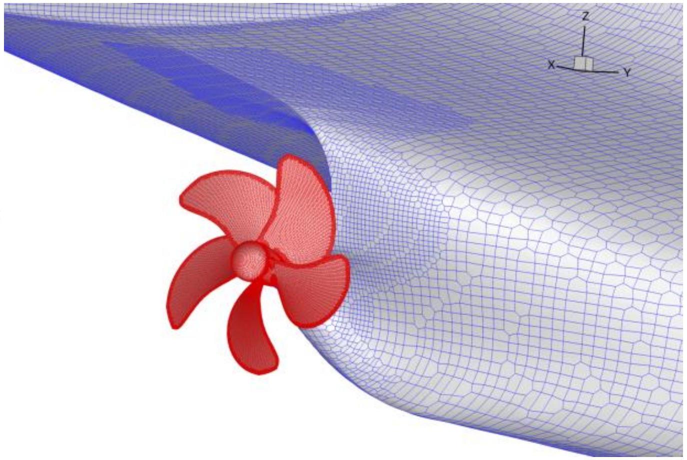
HSVA KCS Model

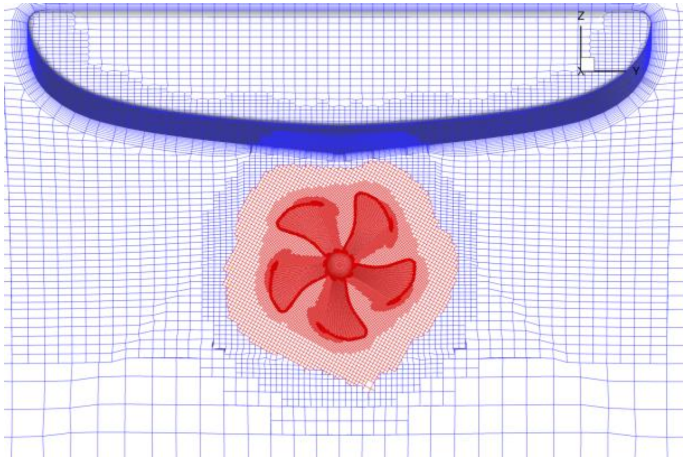




Towed and self-propulsion







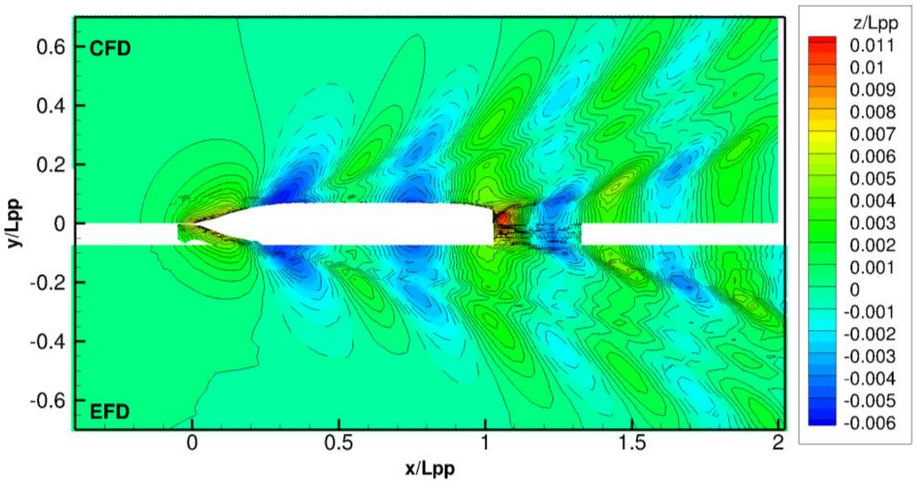
Grids of self-propulsion

Grid sizes

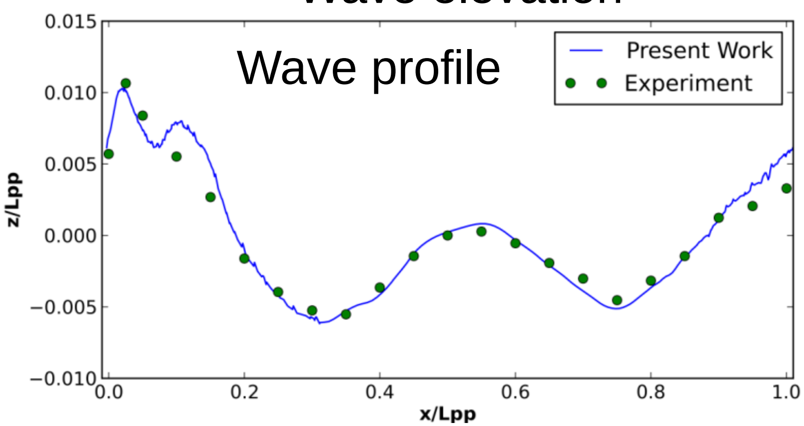
Mesh size	Hull	Background	Propeller	Total
Towed	0.959 M	0.716 M	-	1.675 M
Self-propelled	1.129 M	0.716 M	1.368 M	3.213 M

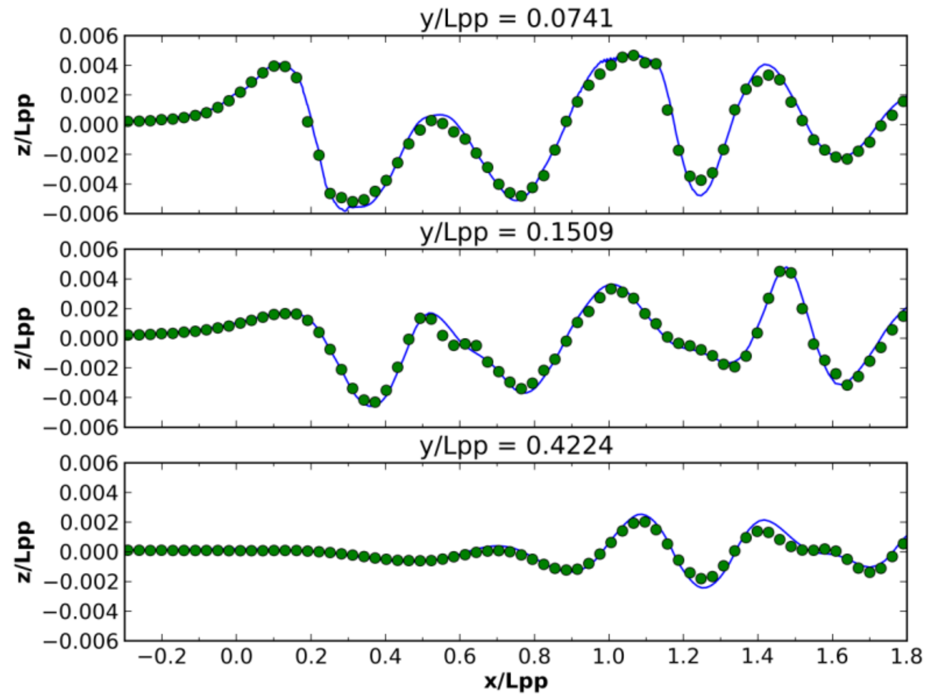
- The grid used for the towed computations is the same grid but without the propeller and related refinement.

Towed condition (bare hull)



Wave elevation





Free-surface cut

Self-propulsion condition

- Fixed at even-keel condition.
- Performed at ship point.
- Different viscous force in model and ship scales.
- Skin friction correction:

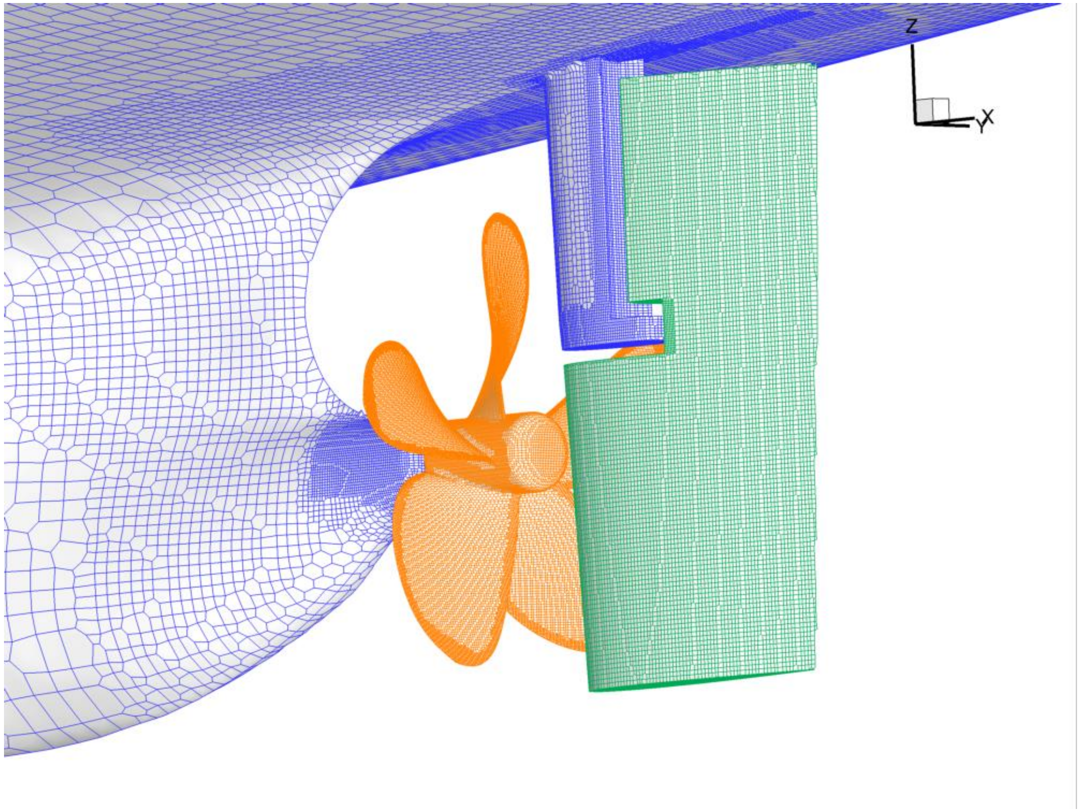
$$\mathrm{SFC} = \{(1+\mathrm{k})(C_{F0M} - C_{F0S}) - \Delta C_F\} \times \frac{1}{2}\rho U_0^2 A_W$$

- PI Controller to adjust RPS of propeller until final balance is reached:

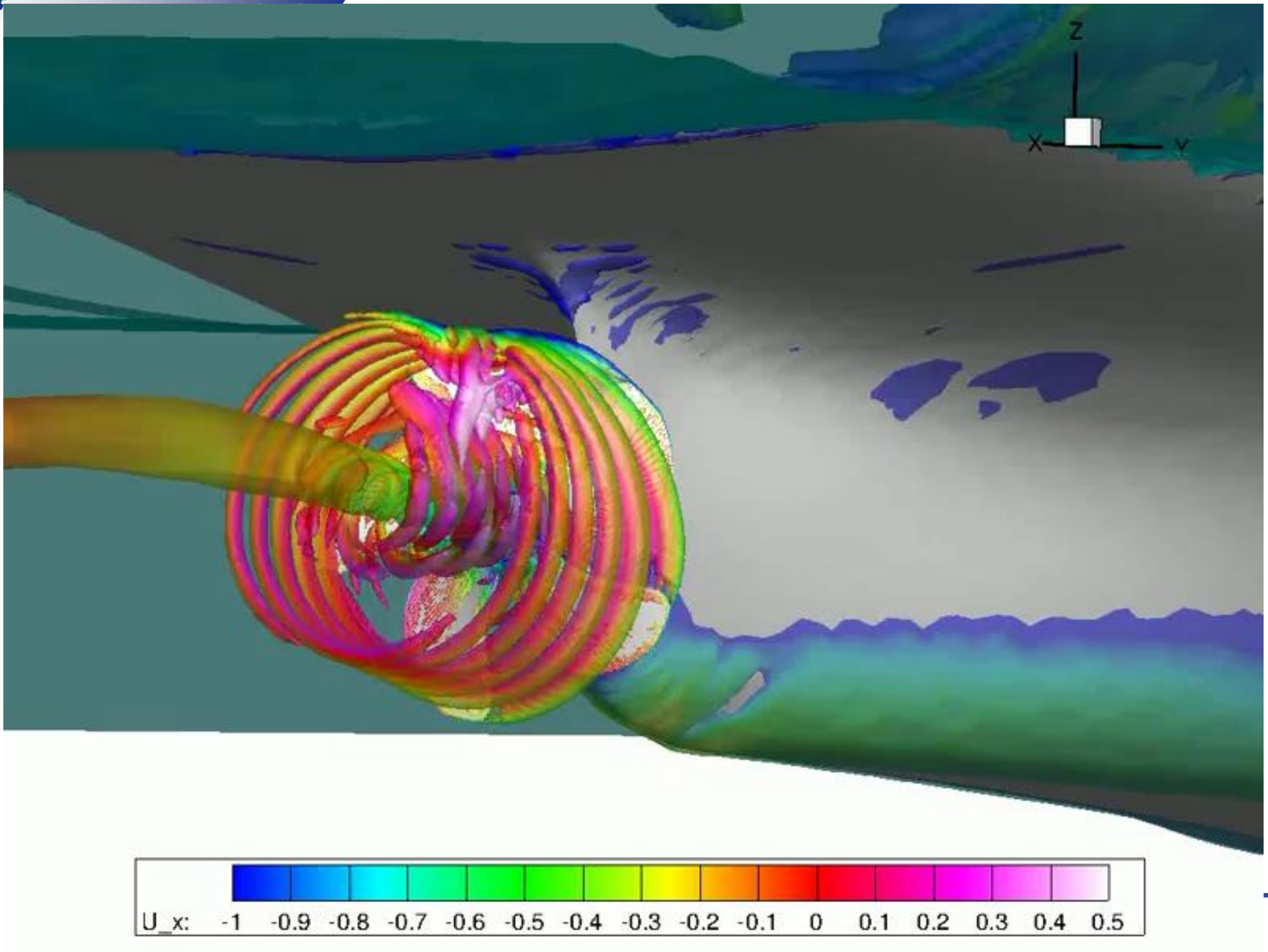
$$\mathrm{T} = \mathrm{R}_{\mathrm{T(SP)}} - \mathrm{SFC}$$

HSVA KCS Model

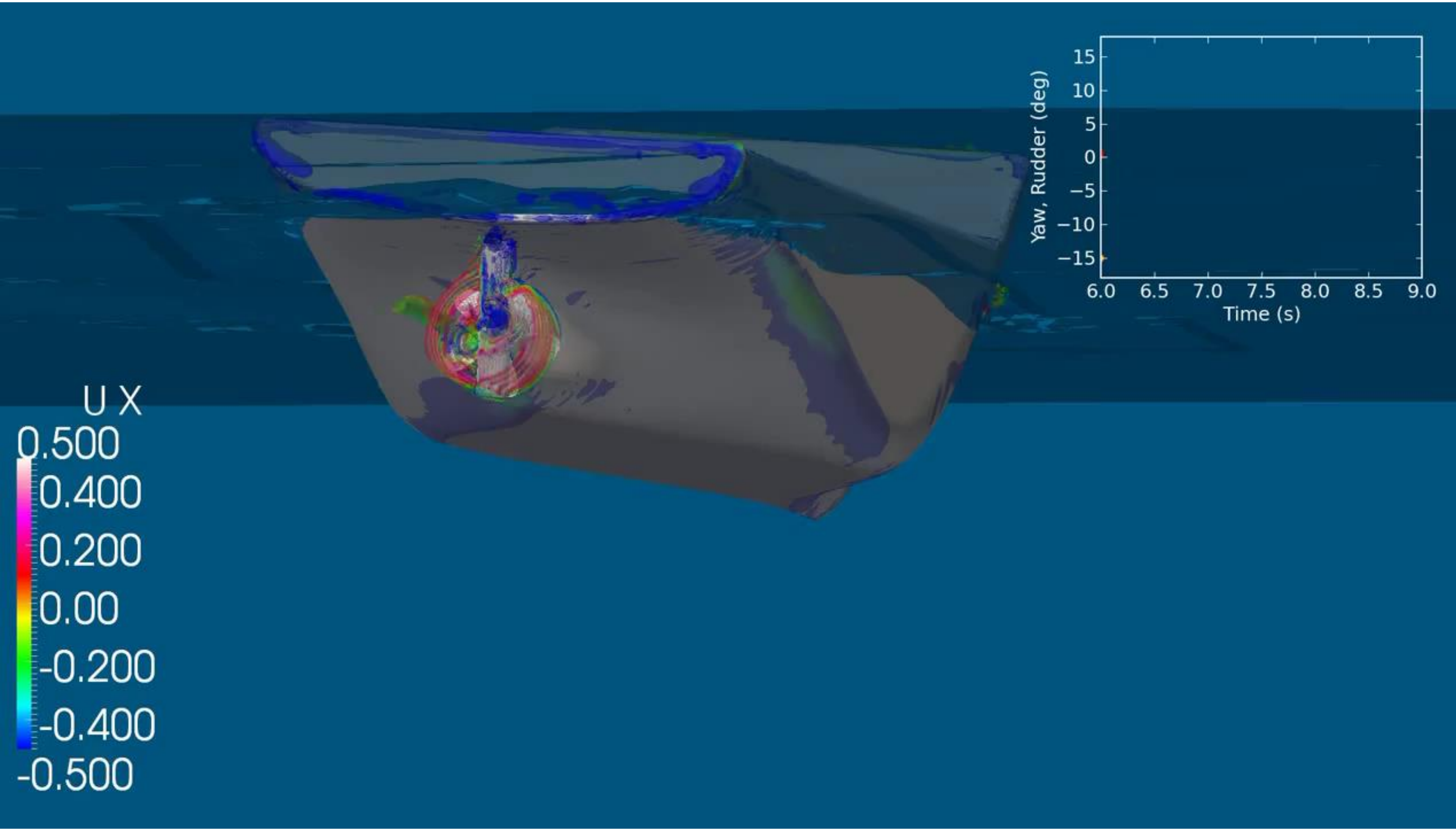
- Semi-balanced horn rudder
- Propeller: SVP1193 (5-blade)
- Full 6DOF
- Rotating propeller
- Moving rudder.



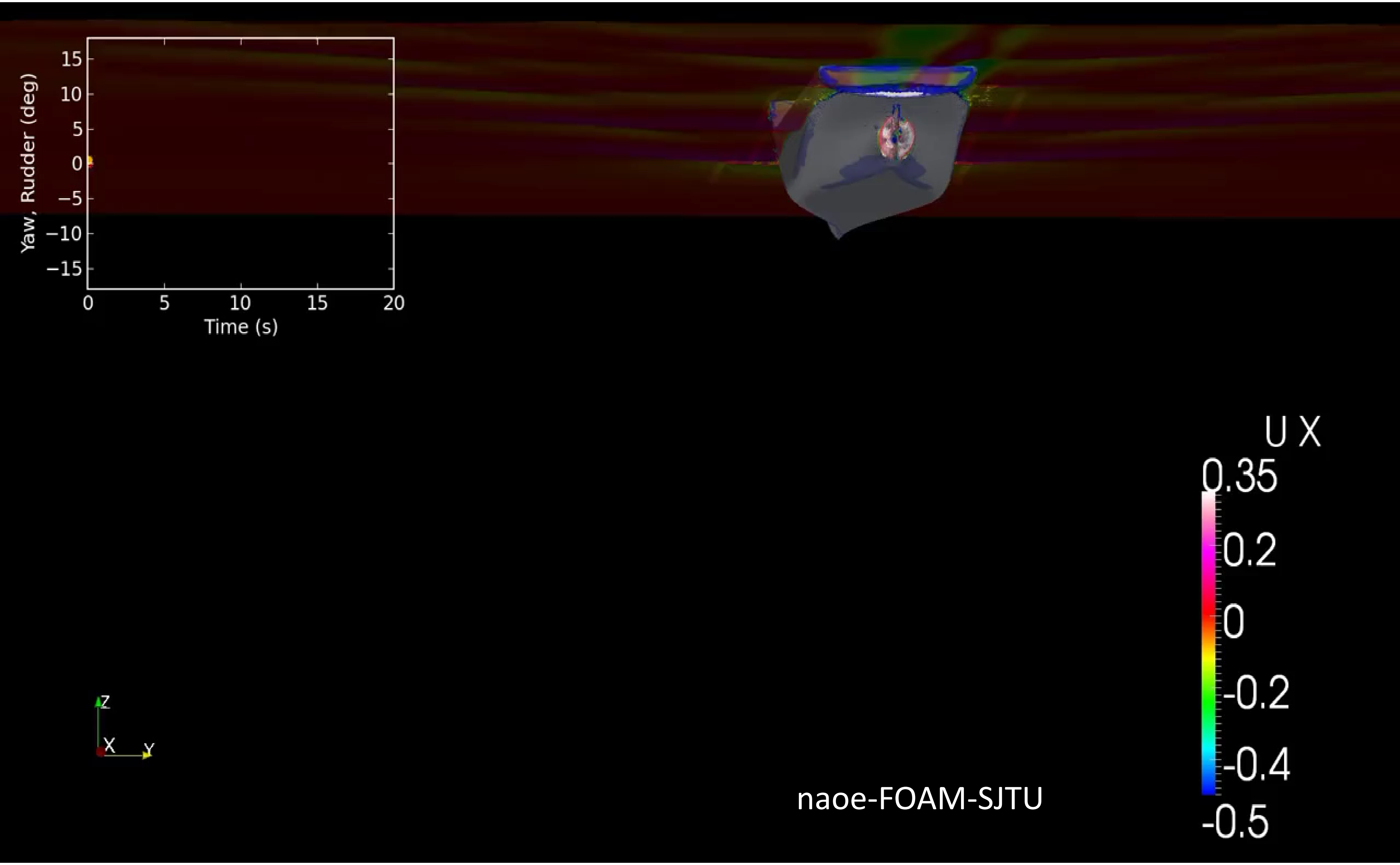
Self-Propulsion of Ship Motion



Self-Propulsion of Ship Motion

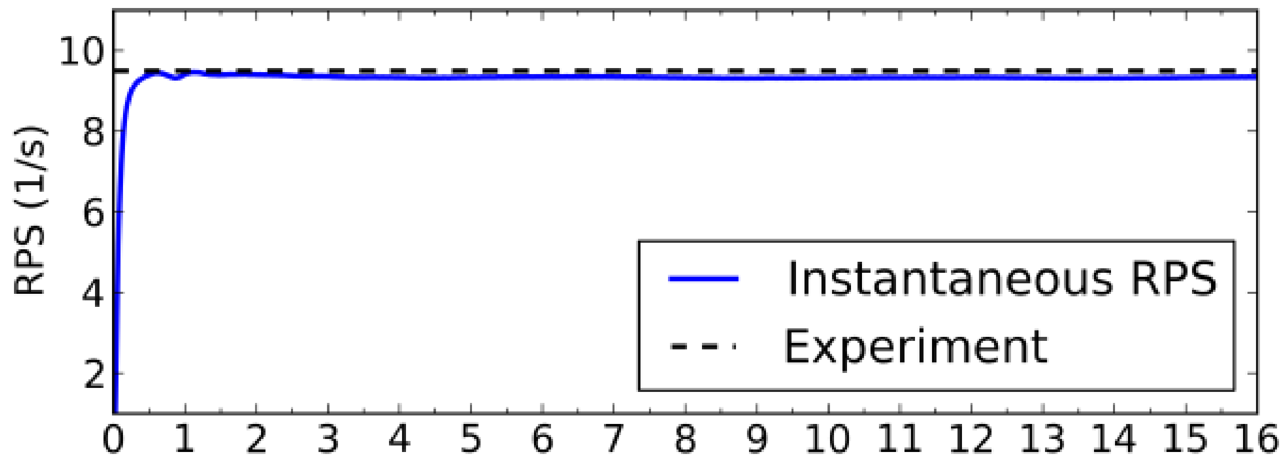


Self-Propulsion of Ship Motion

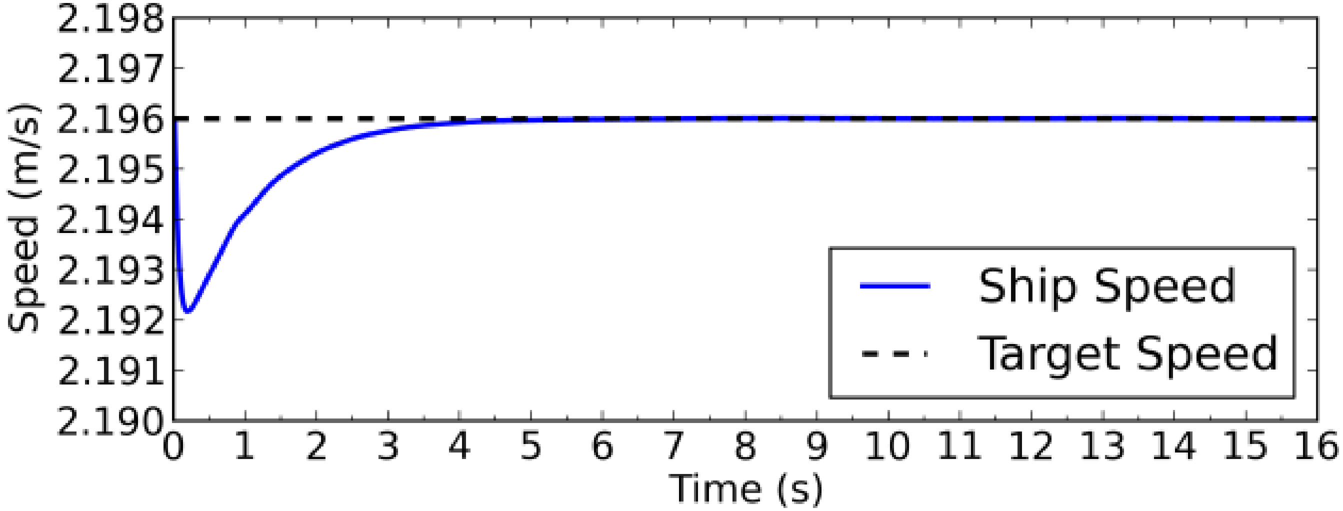


Self-Propulsion of Ship Motion

Propeller speed



Ship speed

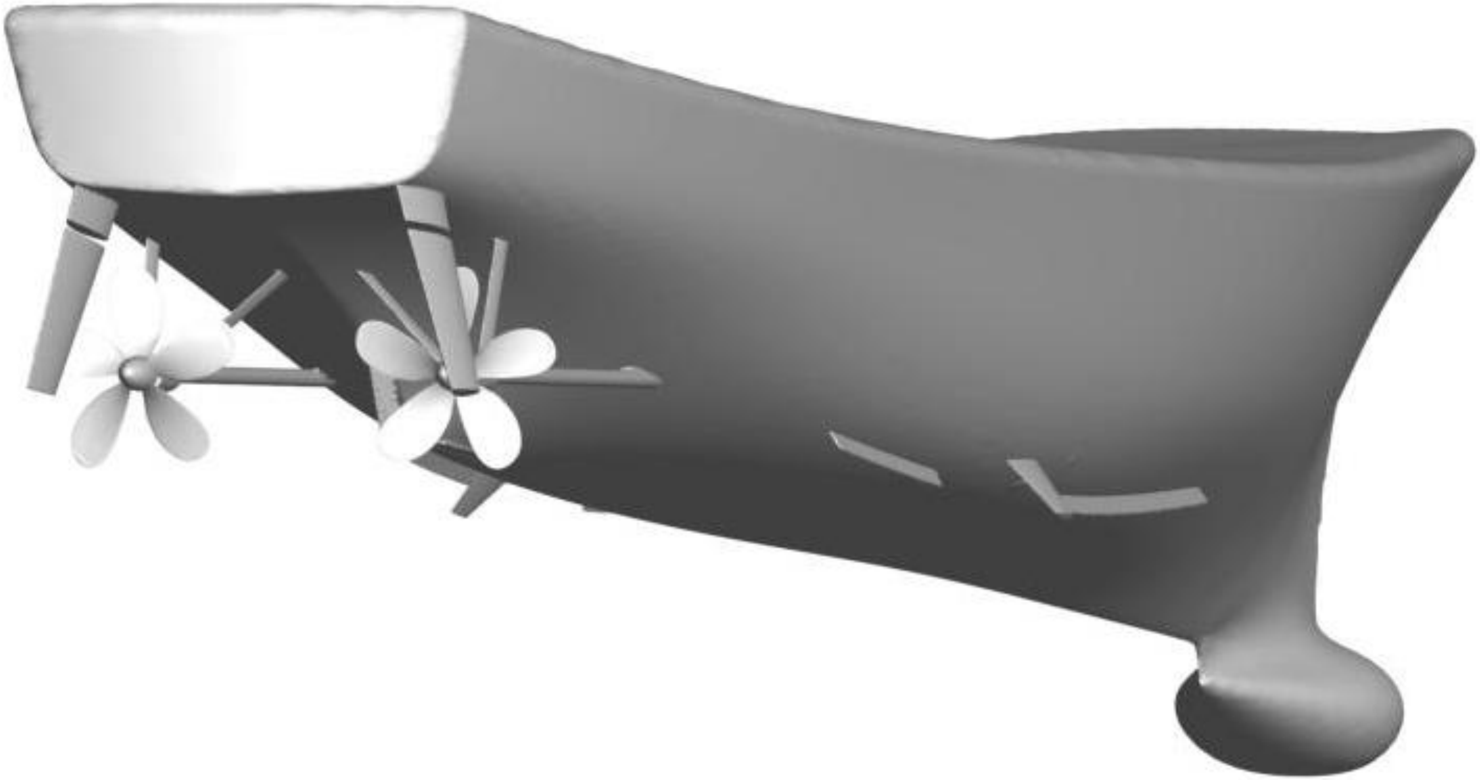


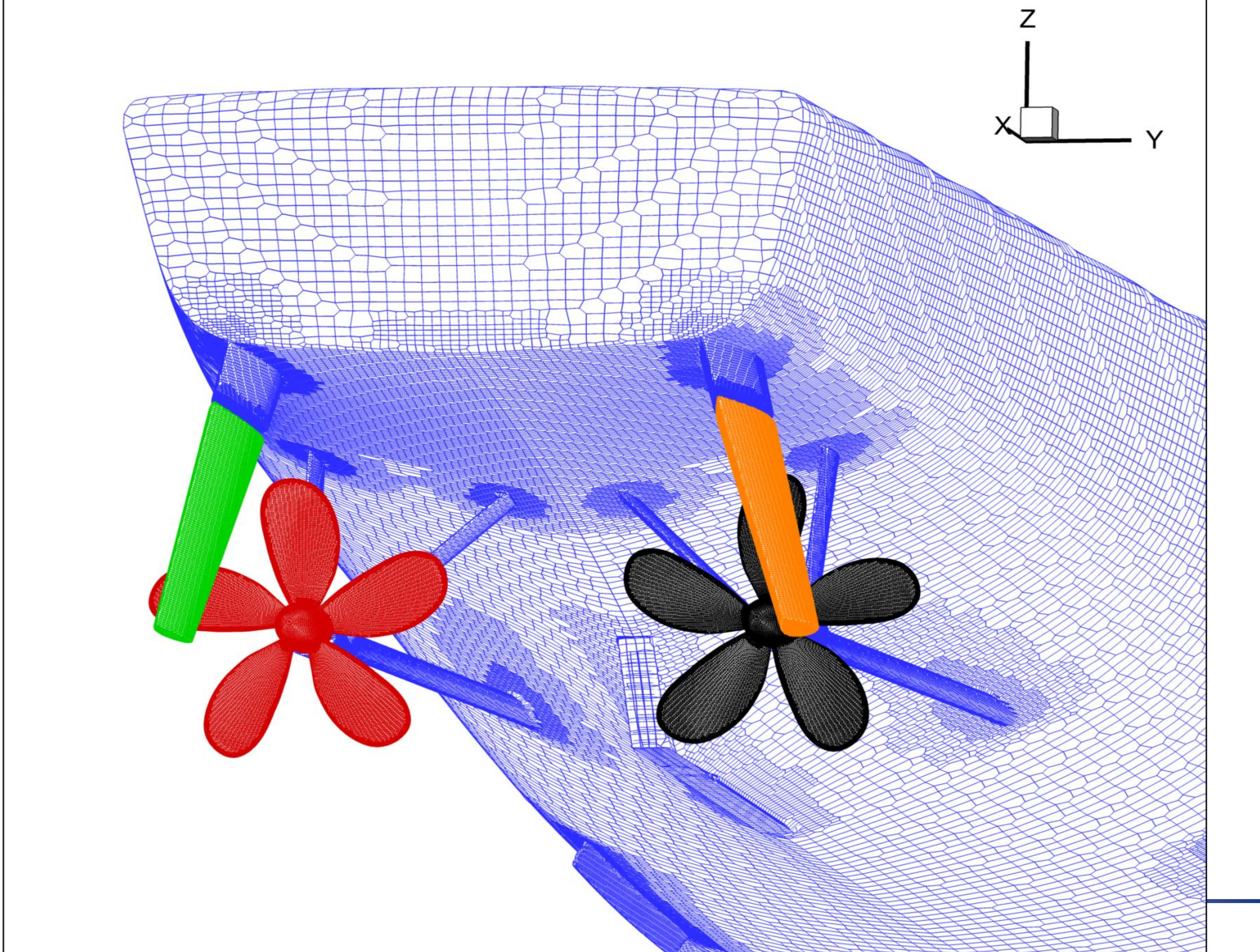
Self-Propulsion of Ship Motion

	Experiment	Present Work	% error	CFDShip-Iowa (DES)
C_T	3.942×10^{-3}	3.840×10^{-3}	-2.586%	4.011×10^{-3}
K_T	0.17	0.1682	-1.061%	0.1689
K_Q	0.0288	0.0290	0.863%	0.02961
$1-t$	0.853	0.8857	3.838%	0.8725
$1-W_t$	0.792	0.7815	-1.326%	0.803
η_o	0.682	0.6785	-0.507%	0.683
η_R	1.011	0.9811	-2.955%	0.976
J	0.728	0.7363	1.142%	0.733
n	9.5	9.3231	-1.862%	9.62
η	0.74	0.7545	1.963%	0.724

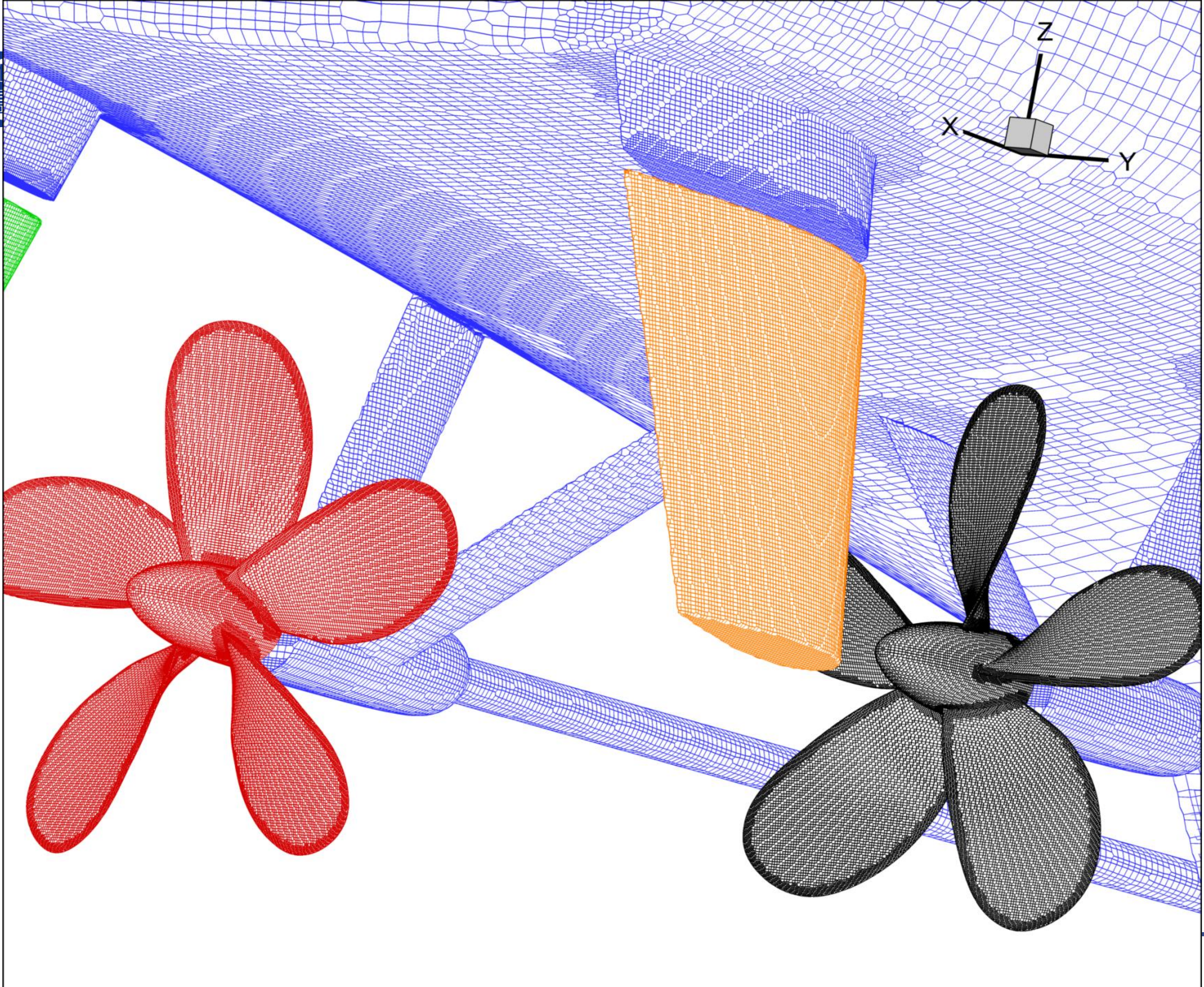
SELF-PROPULSION FOR TWO PROPELLERS SHIP

Self-Propulsion of Ship Motion

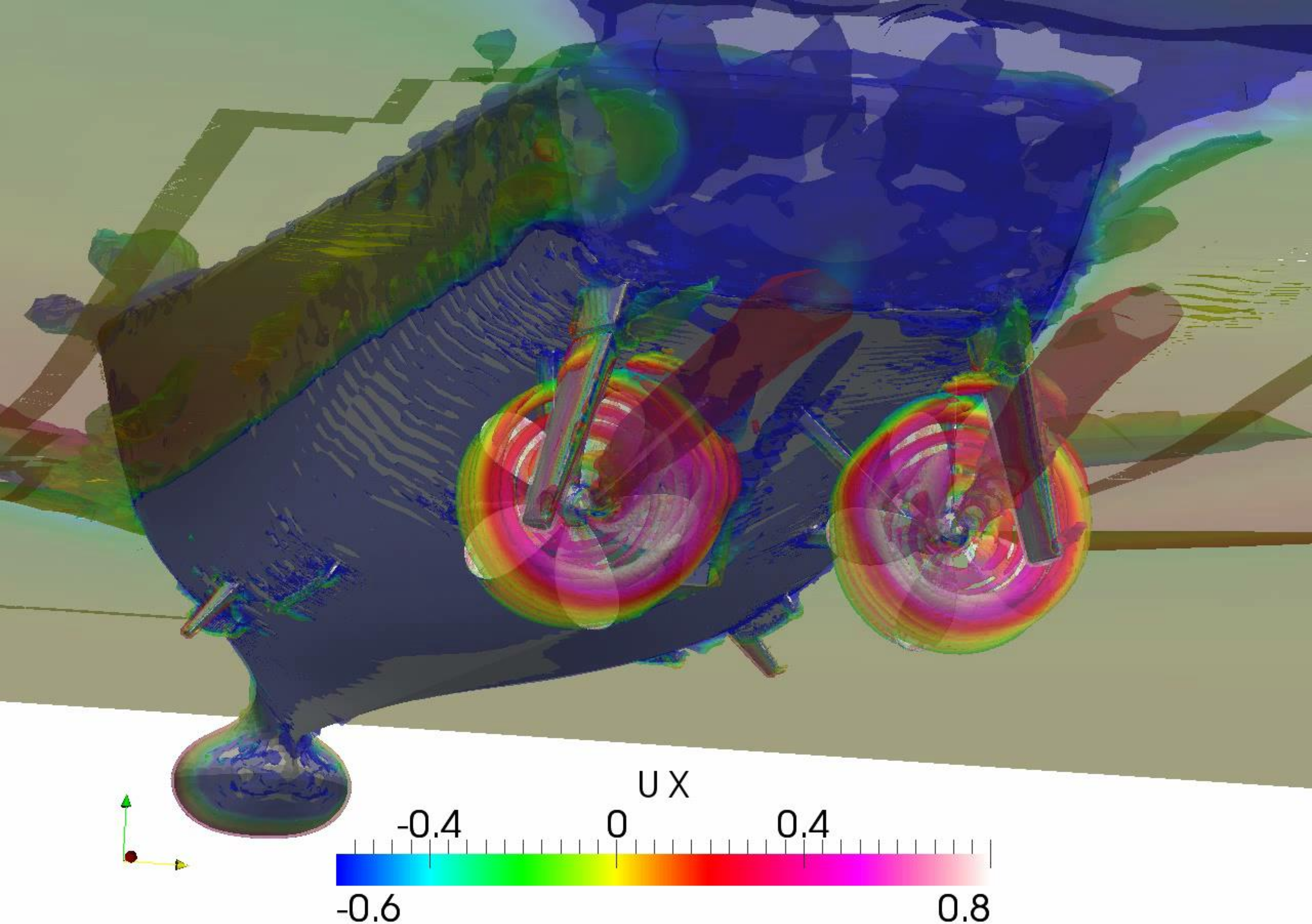


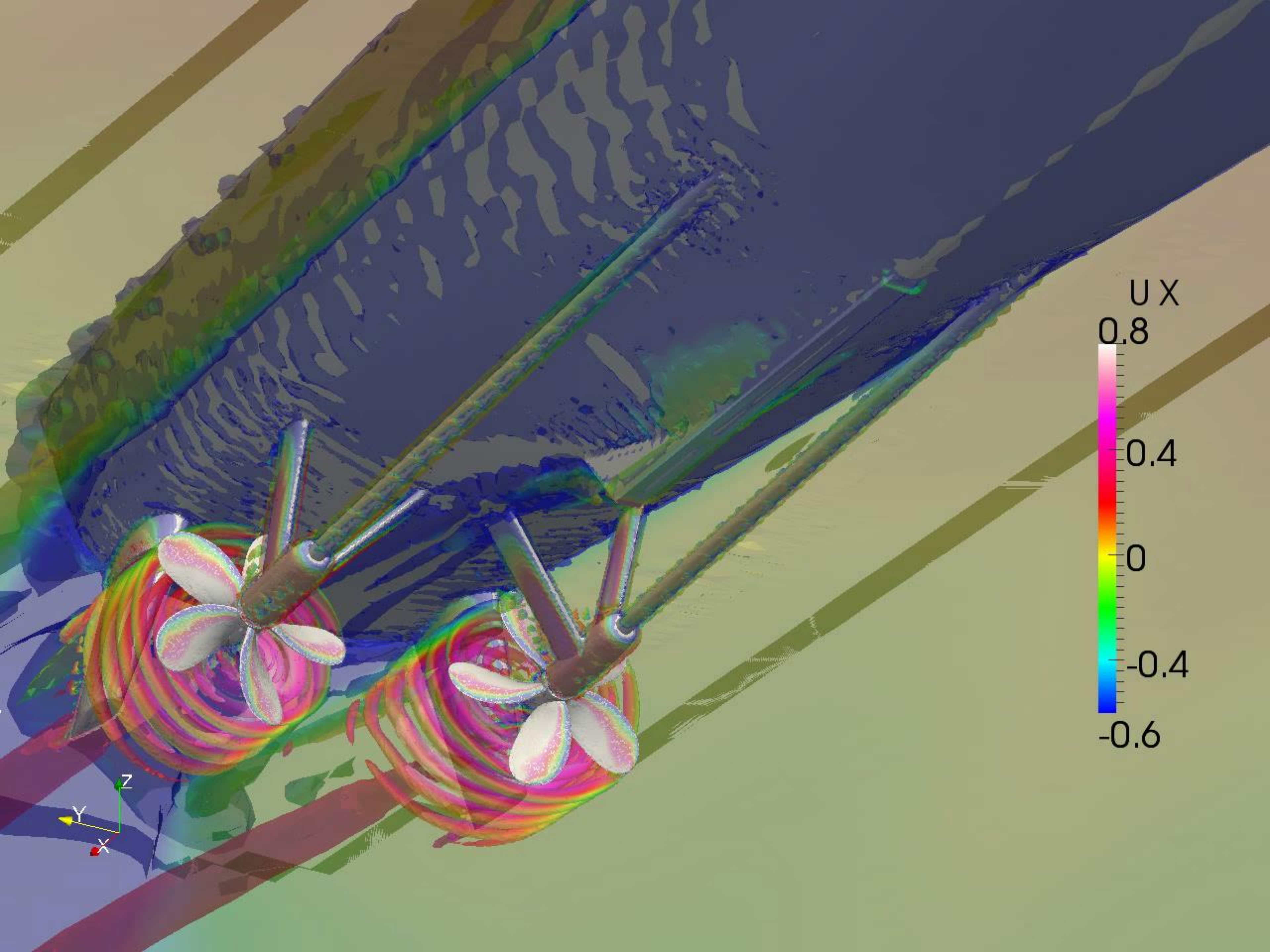


Z
X
Y

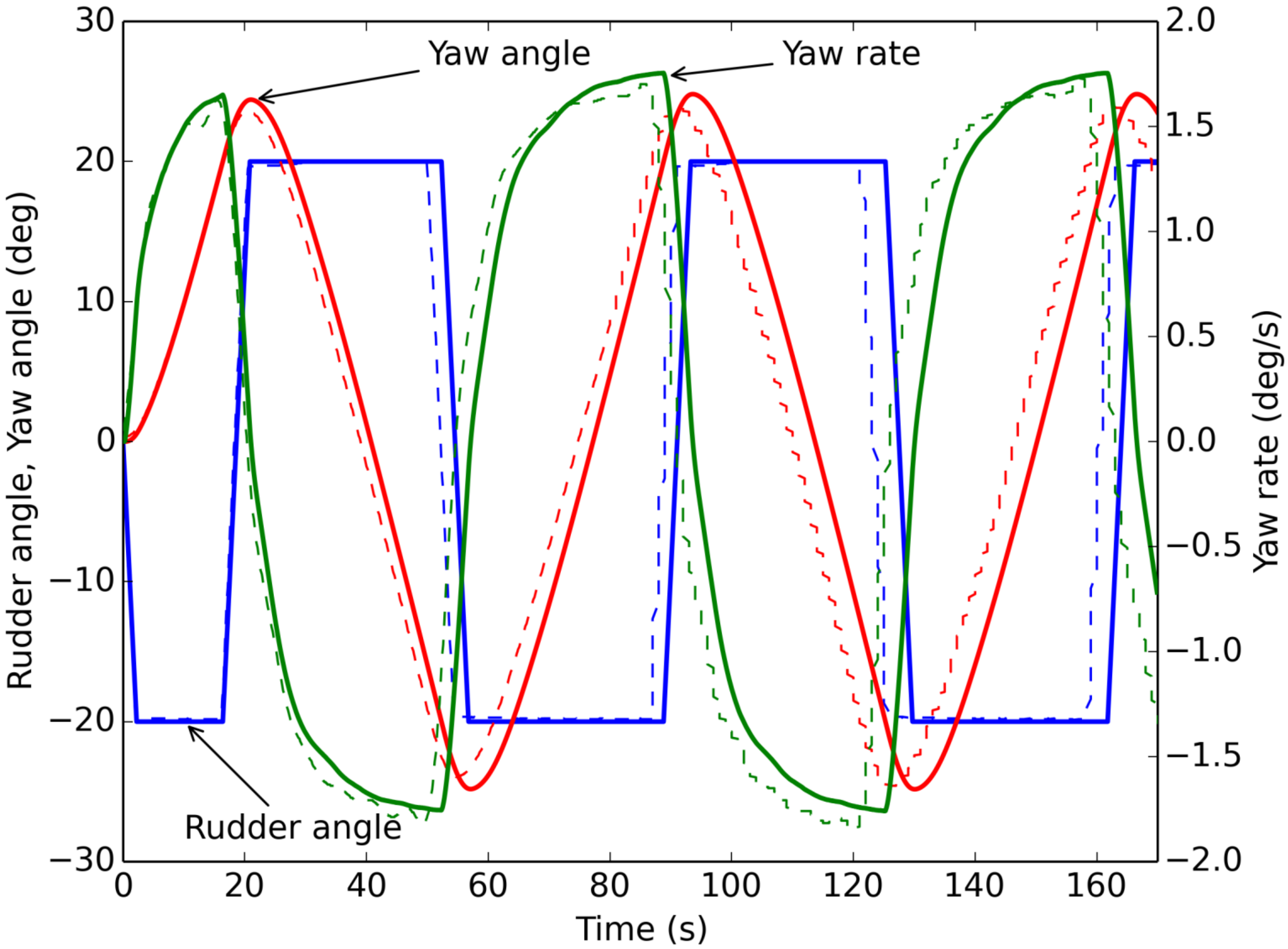


Z
X
Y


U X
-0.4
0
0.4
-0.6
0.8

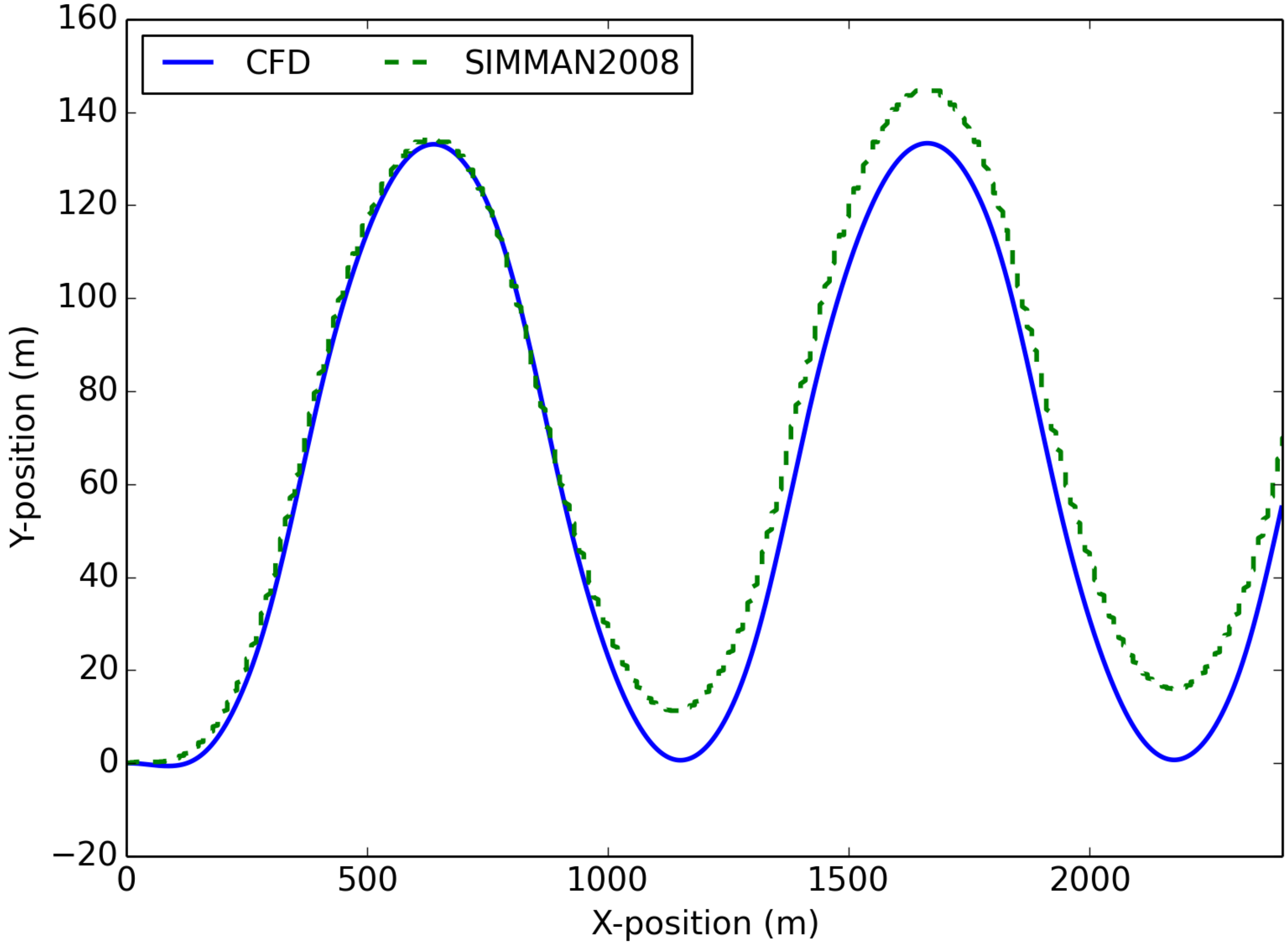

U X
0.8
0.4
0
-0.4
-0.6
Z
Y
X

Rudder and Yaw motion

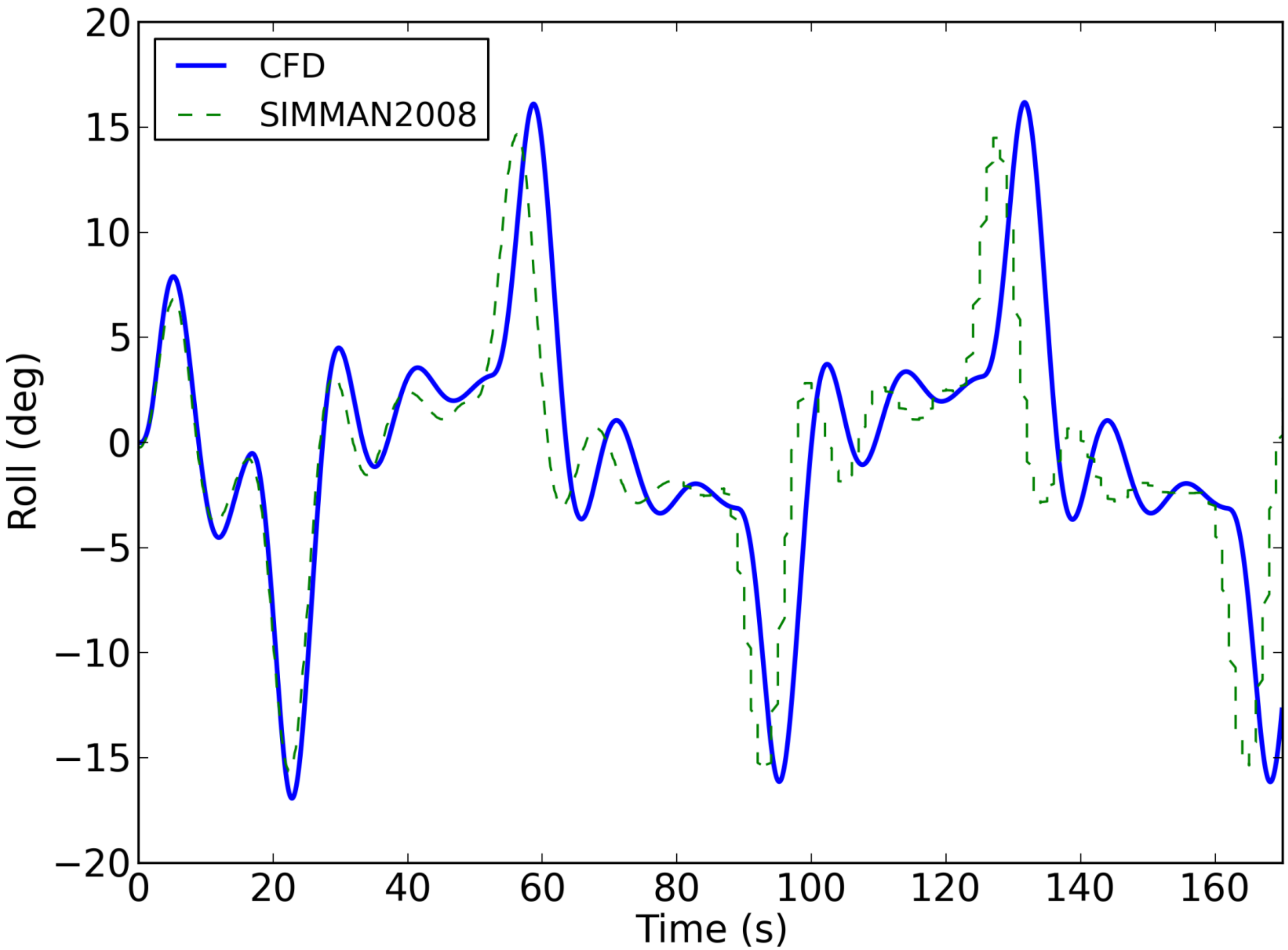


Solid line – CFD; Dashed line -- EXP

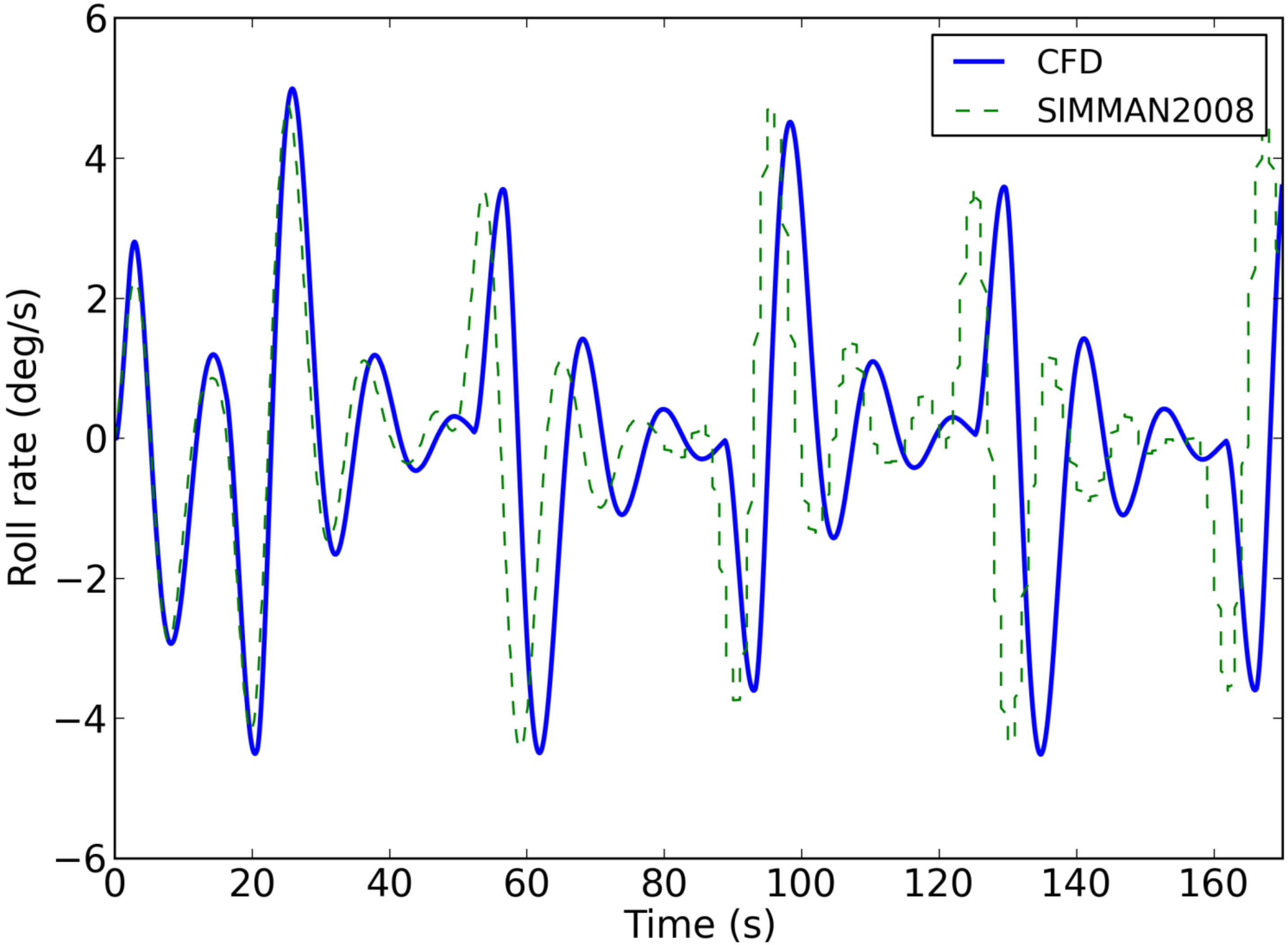
Trajectory


CFD
SIMMAN2008
Y-position (m)
X-position (m)
160
140
120
100
80
60
40
20
0
−20
0
500
1000
1500
2000

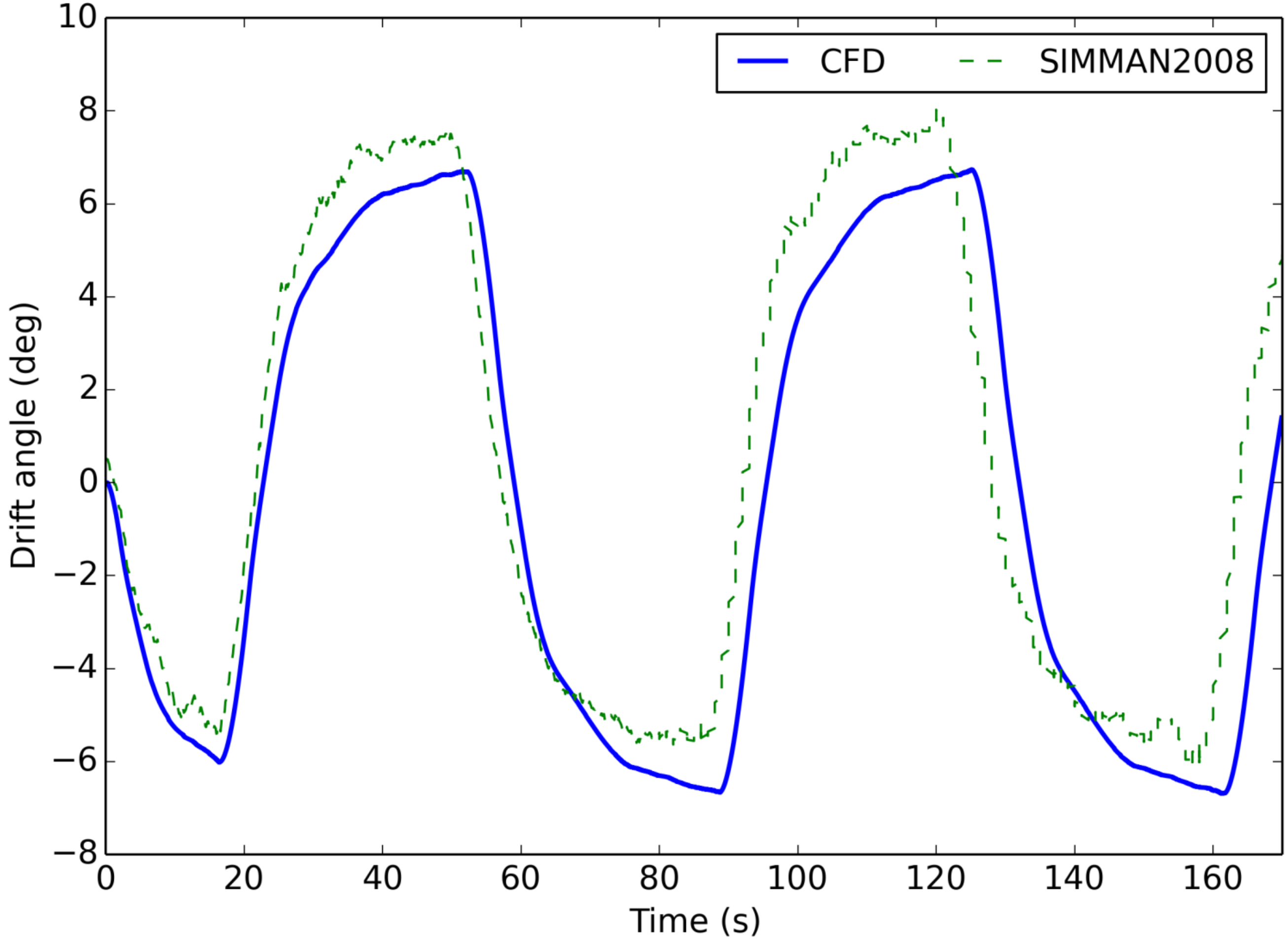
Roll motion

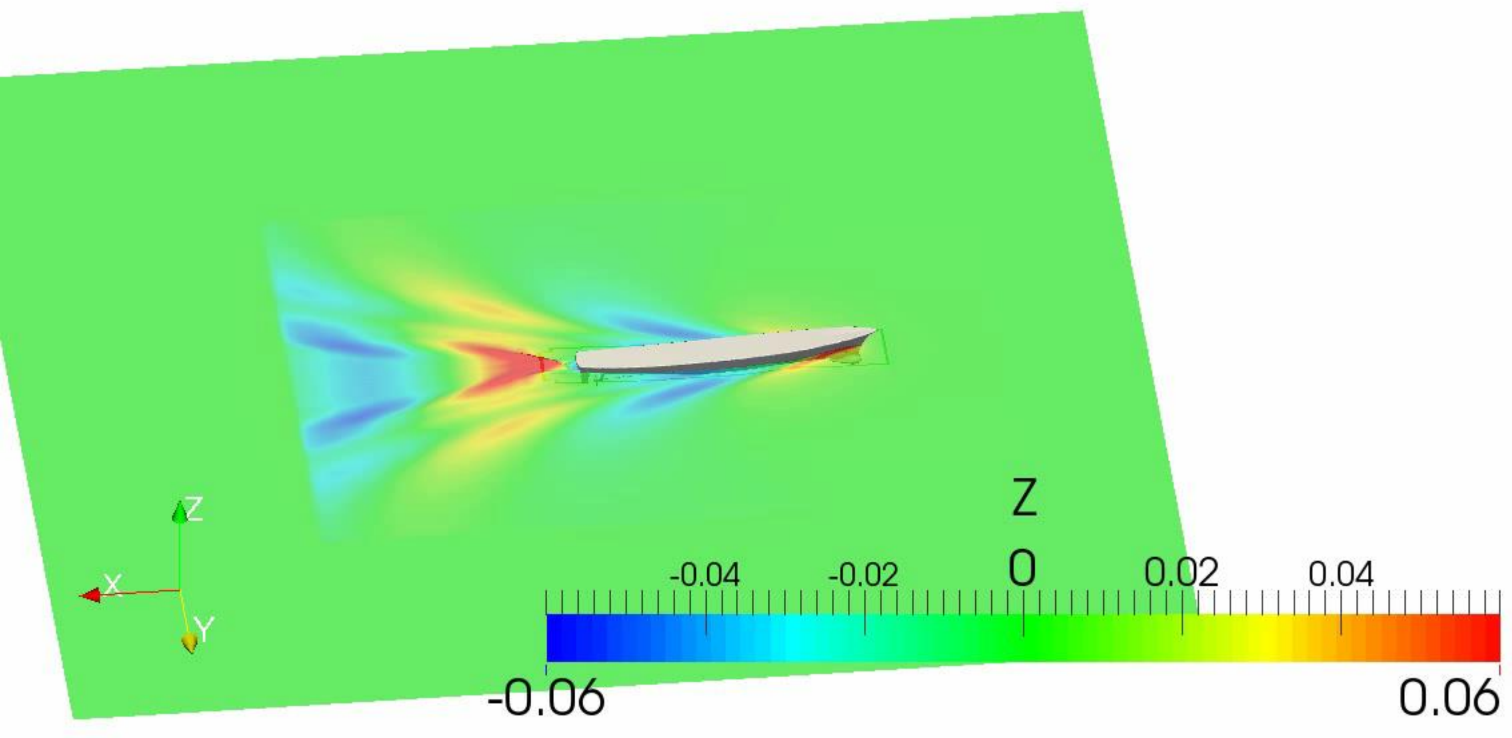

CFD
SIMMAN2008
Roll (deg)
Time (s)
20
15
10
5
0
−5
−10
−15
−20
0
20
40
60
80
100
120
140
160

Roll rate

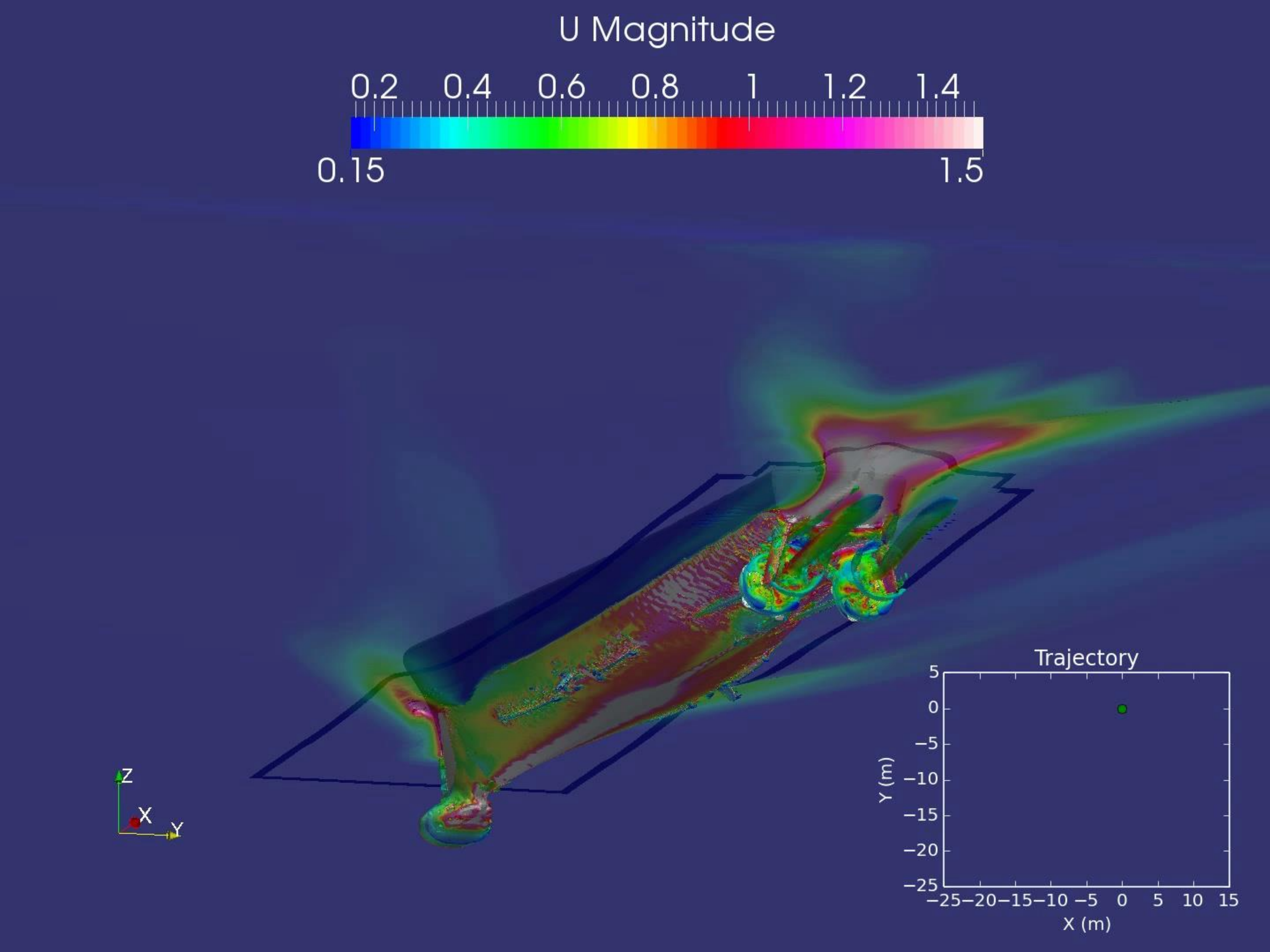

CFD
SIMMAN2008
Roll rate (deg/s)
Time (s)
6
4
2
0
−2
−4
−6
0
20
40
60
80
100
120
140
160

Drift angle

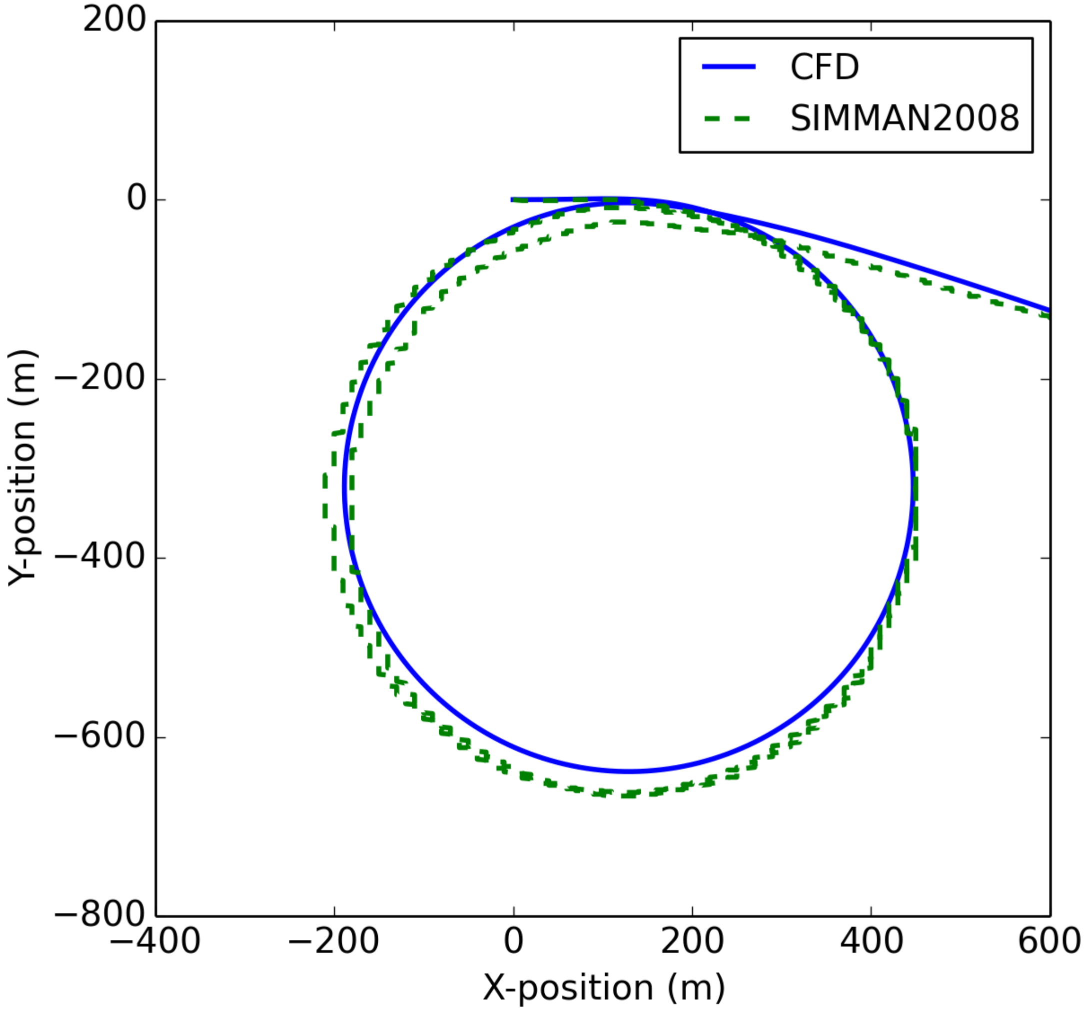

CFD
SIMMAN2008
Drift angle (deg)
Time (s)
10
8
6
4
2
0
−2
−4
−6
−8
0
20
40
60
80
100
120
140
160



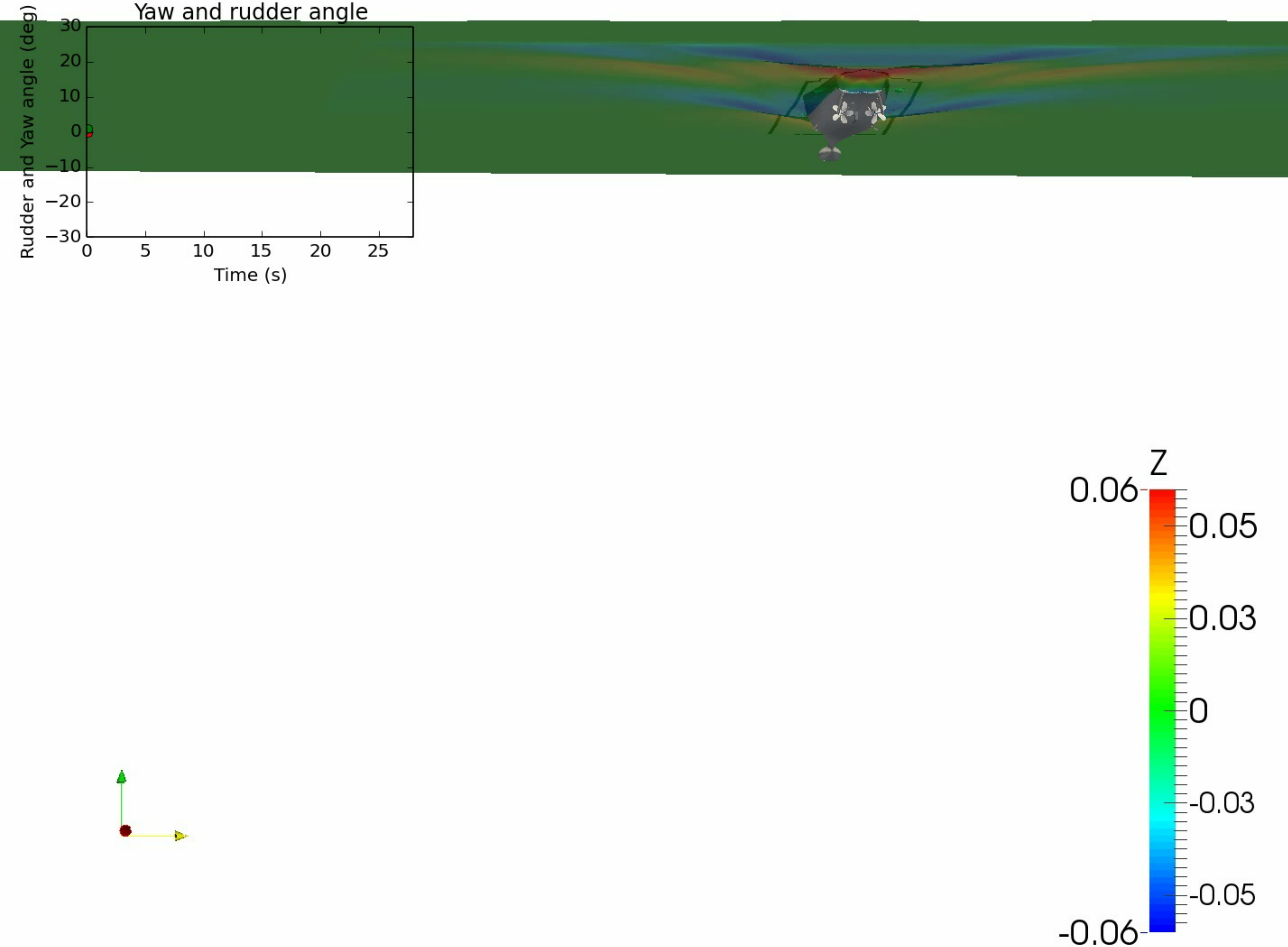
Z
X
Y
Z
-0.04
-0.02
0
0.02
0.04
-0.06
0.06


U Magnitude
0.2
0.4
0.6
0.8
1
1.2
1.4
0.15
1.5
Z
X
Y
Trajectory
5
0
−5
−10
−15
−20
−25
Y (m)
−25
−20
−15
−10
−5
0
5
10
15
X (m)

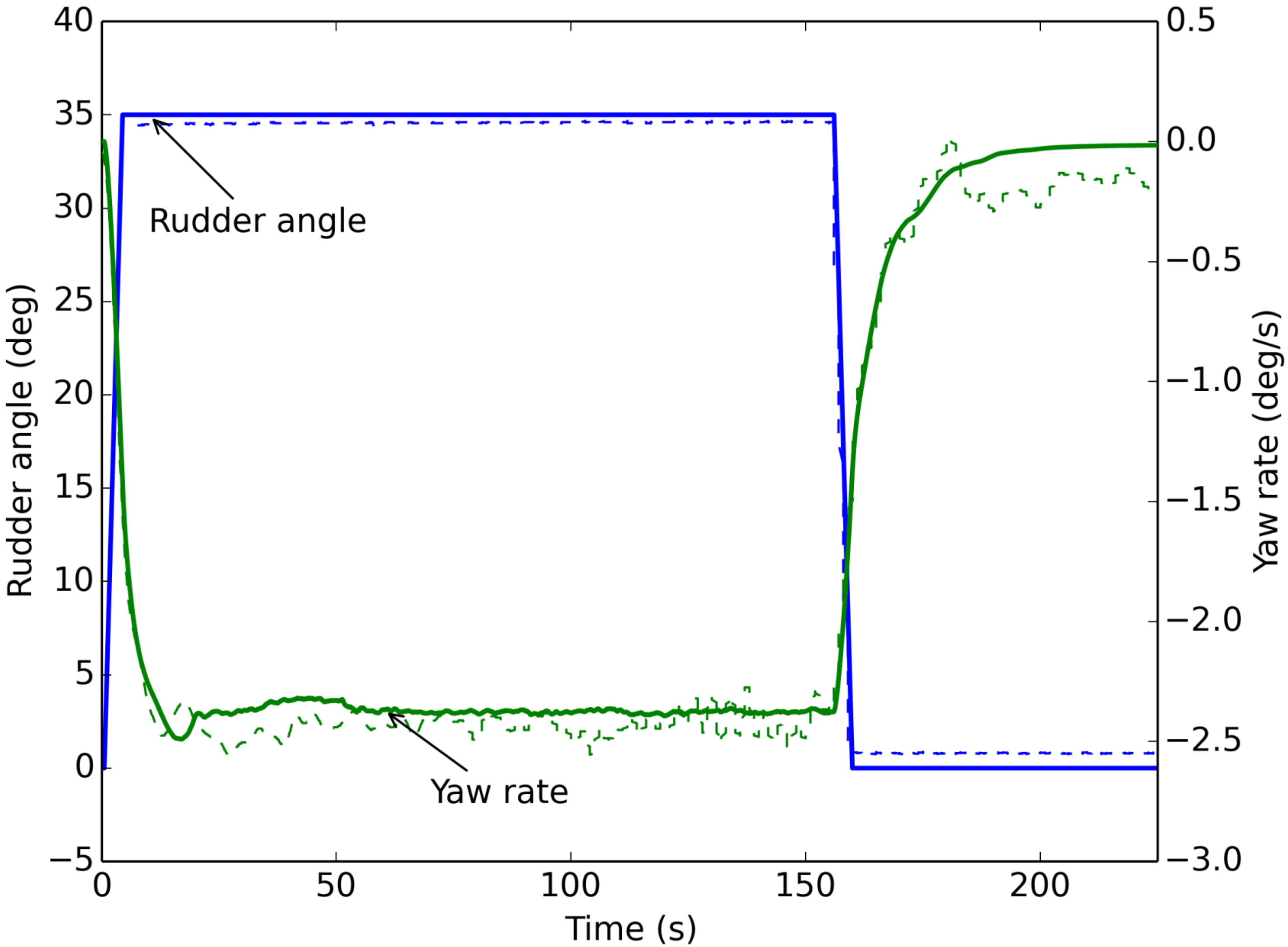
Trajectory



- CFD: 360 deg
- EFD: 720 deg

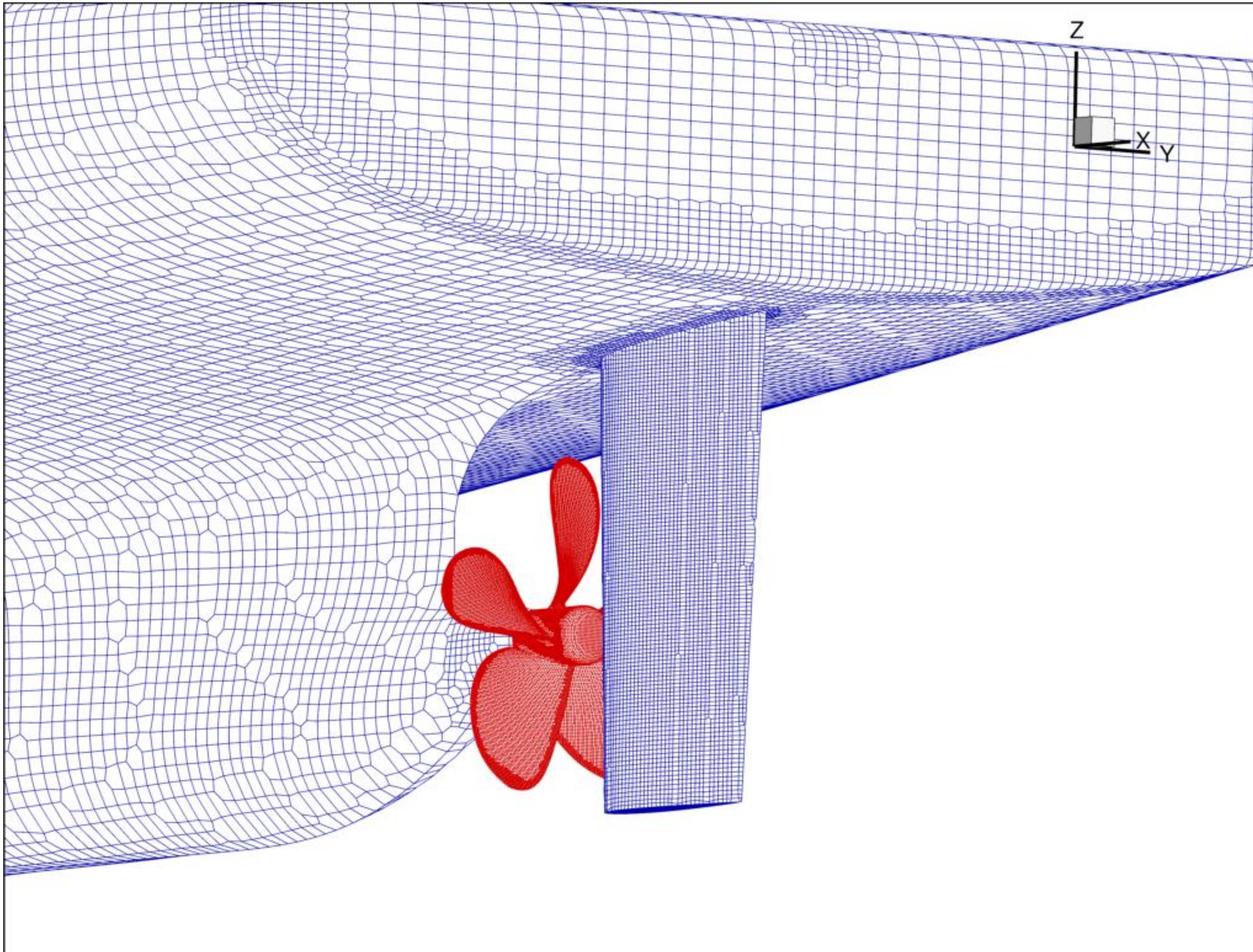

Yaw and rudder angle
Rudder and Yaw angle (deg)
30
20
10
0
−10
−20
−30
0
5
10
15
20
25
Time (s)
Z
0.06
0.05
0.03
0
-0.03
-0.05
-0.06

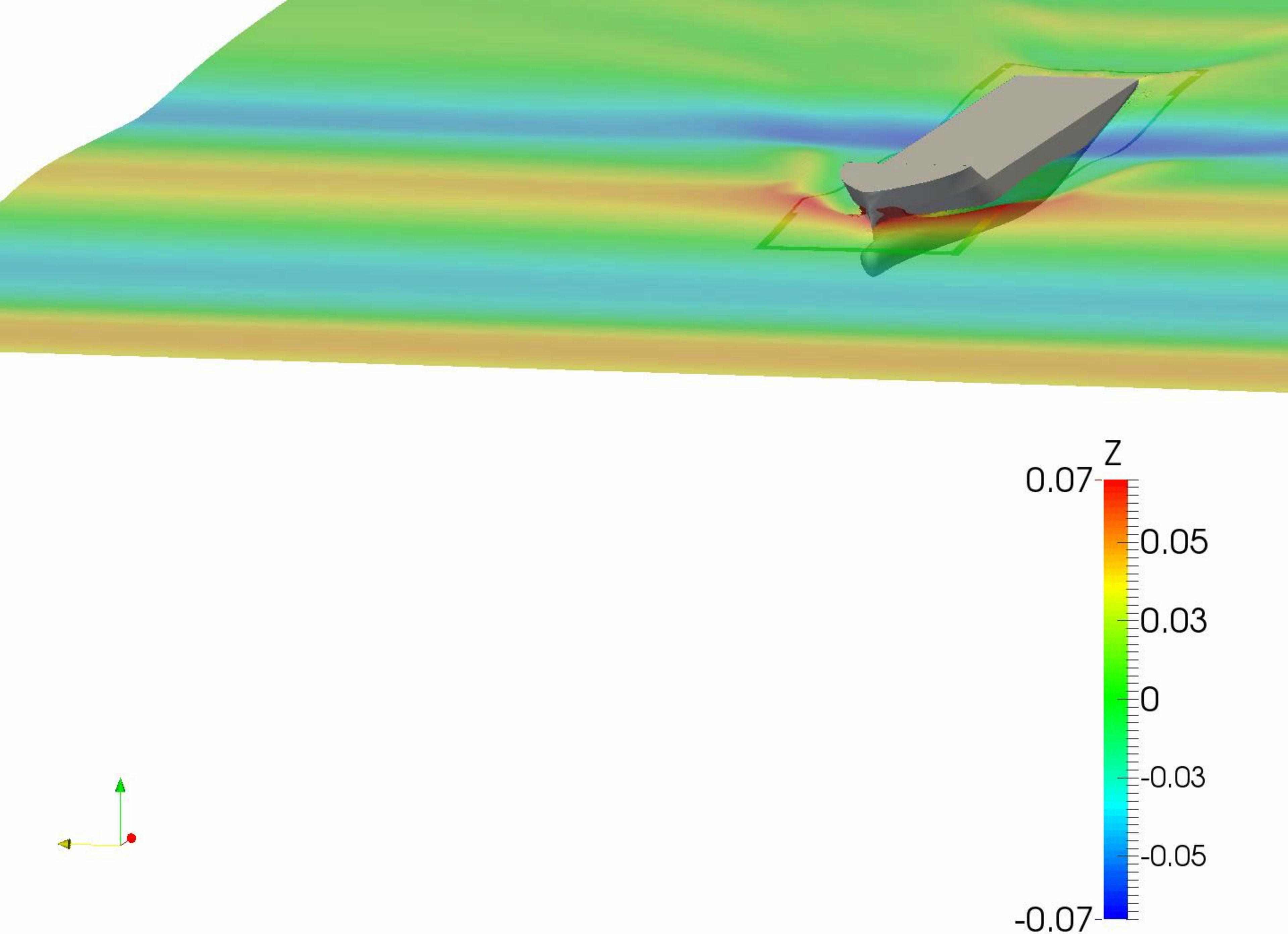
Rudder and Yaw rate

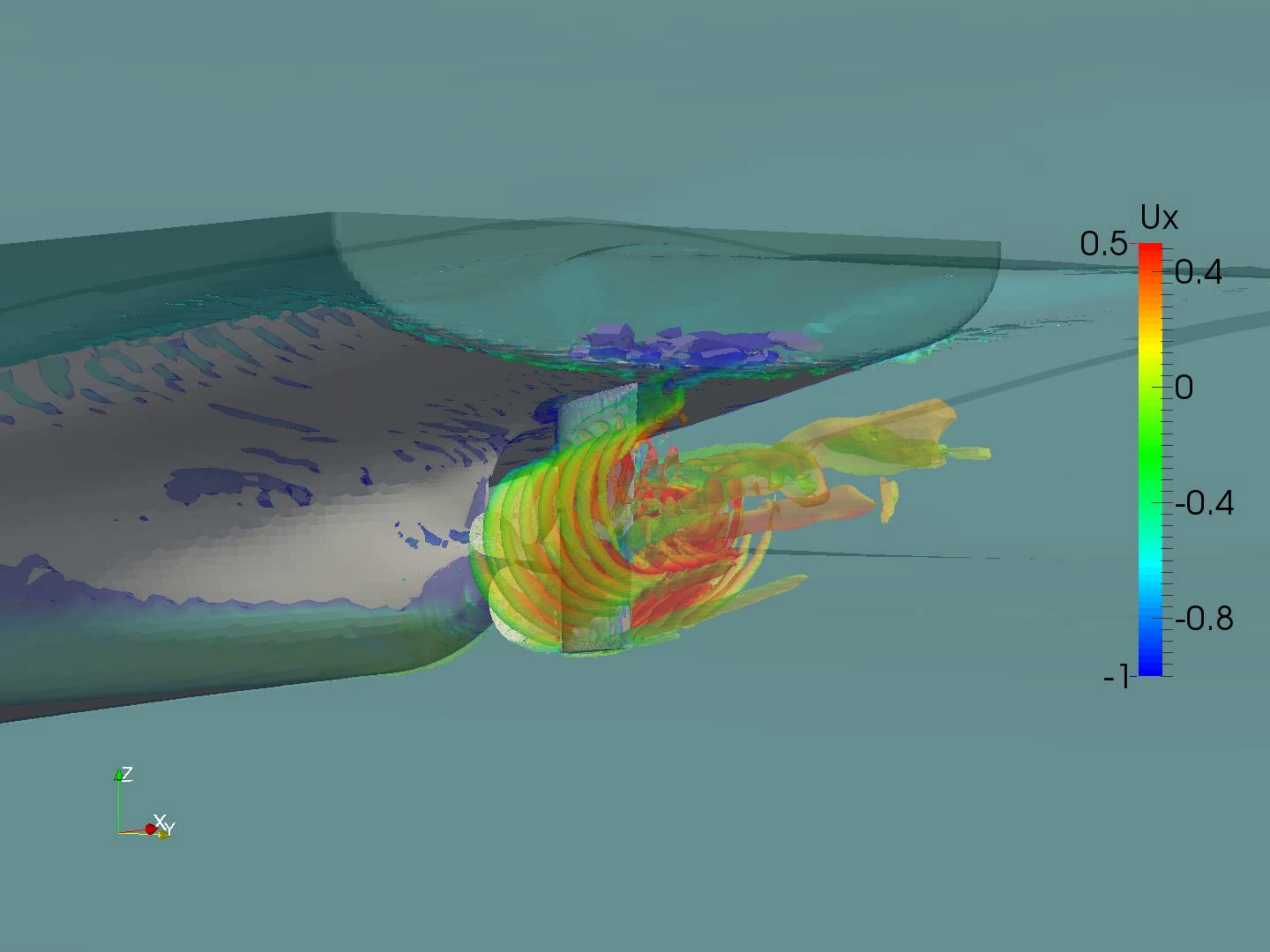


Solid line – CFD; Dashed line -- EXP

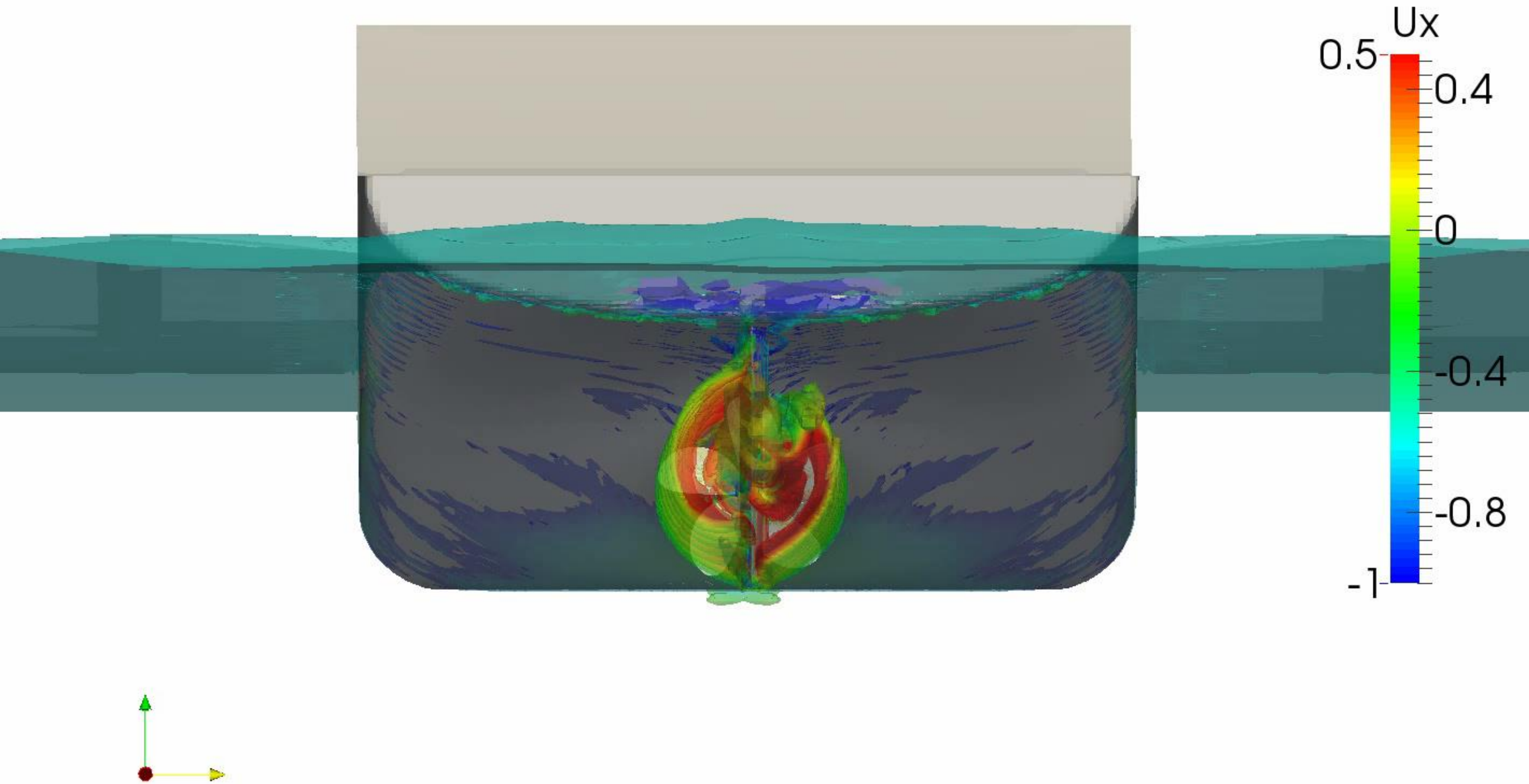
Ship self-propuslion motion in waves


Z
X
Y

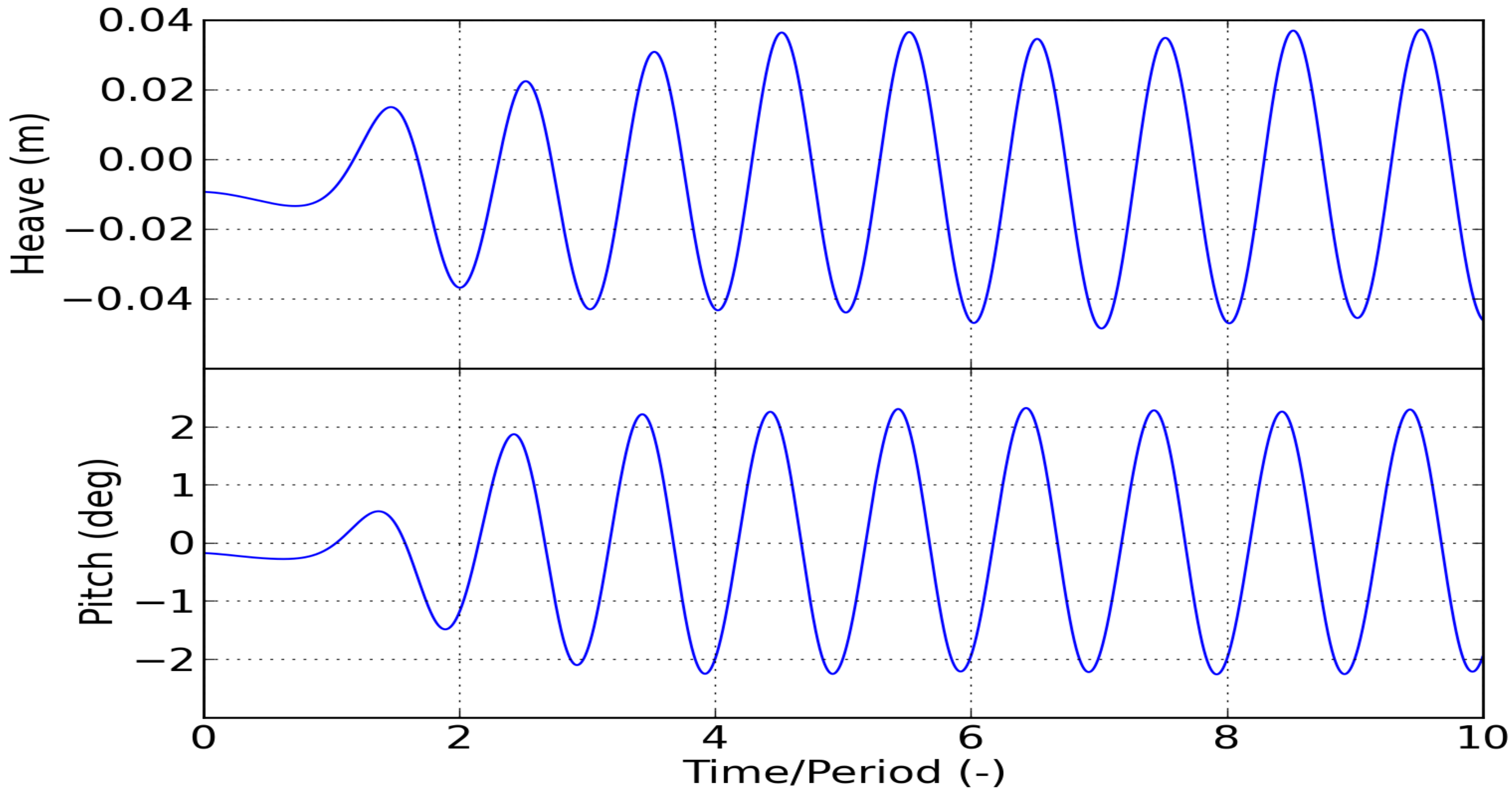

Z
0.07
0.05
0.03
0
-0.03
-0.05
-0.07



Ux
0.5
0.4
0
-0.4
-0.8
-1
Z
X
Y


Ux
0.5
0.4
0
-0.4
-0.8
-1

Motion histories



	TF3	TF5
CFD	0.9785	0.7406
EFD	1.039	0.669

Closing Remarks

- **A solver package naoeFAOM-os-SJTU based on the implementation of the overset grid technique into naoe-FAOM-SJTU is presented .**
- **A self-propulsion study of several ship models in both still water and waves was carried out. All self-propulsion factors were obtained through CFD computations. The results show good agreement between CFD and EFD.**

Thank You !