DIMPACT

Slamming waves implementation into OpenFAST

Summary

OpenFAST

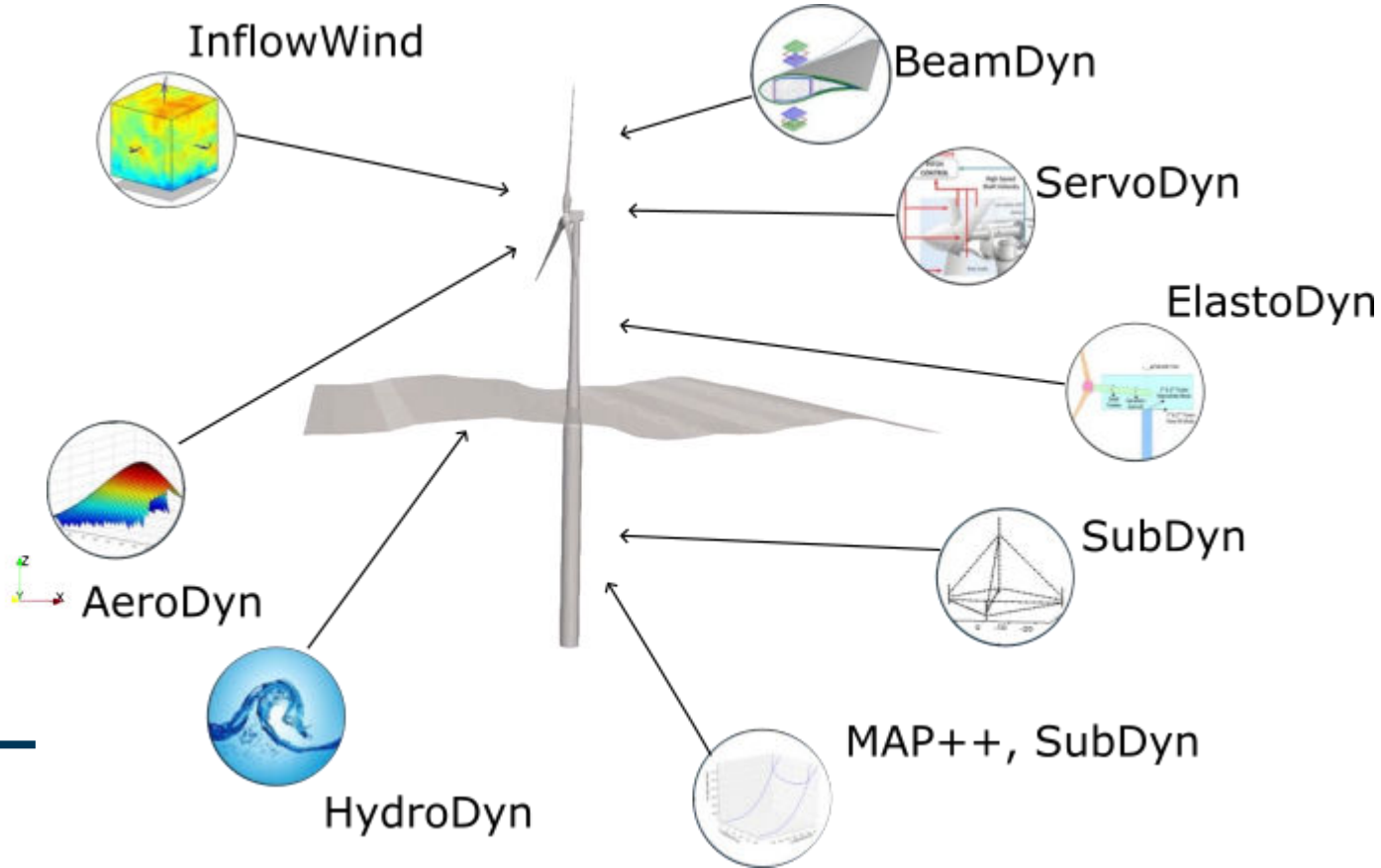
- OpenFAST: An Open-Source Tool for Wind Turbine Physics-Based Engineering Modeling
- Developed by NREL laboratory
- Detailed representation of the wind turbine and exterior solicitations

OpenFAST

- OpenFAST: An Open-Source Tool for Wind Turbine Physics-Based Engineering Modeling

- Detailed representation of the wind turbine and exterior solicitations

- Coupled simulation
  - Aero
  - Hydro
  - Elastic
  - Servo



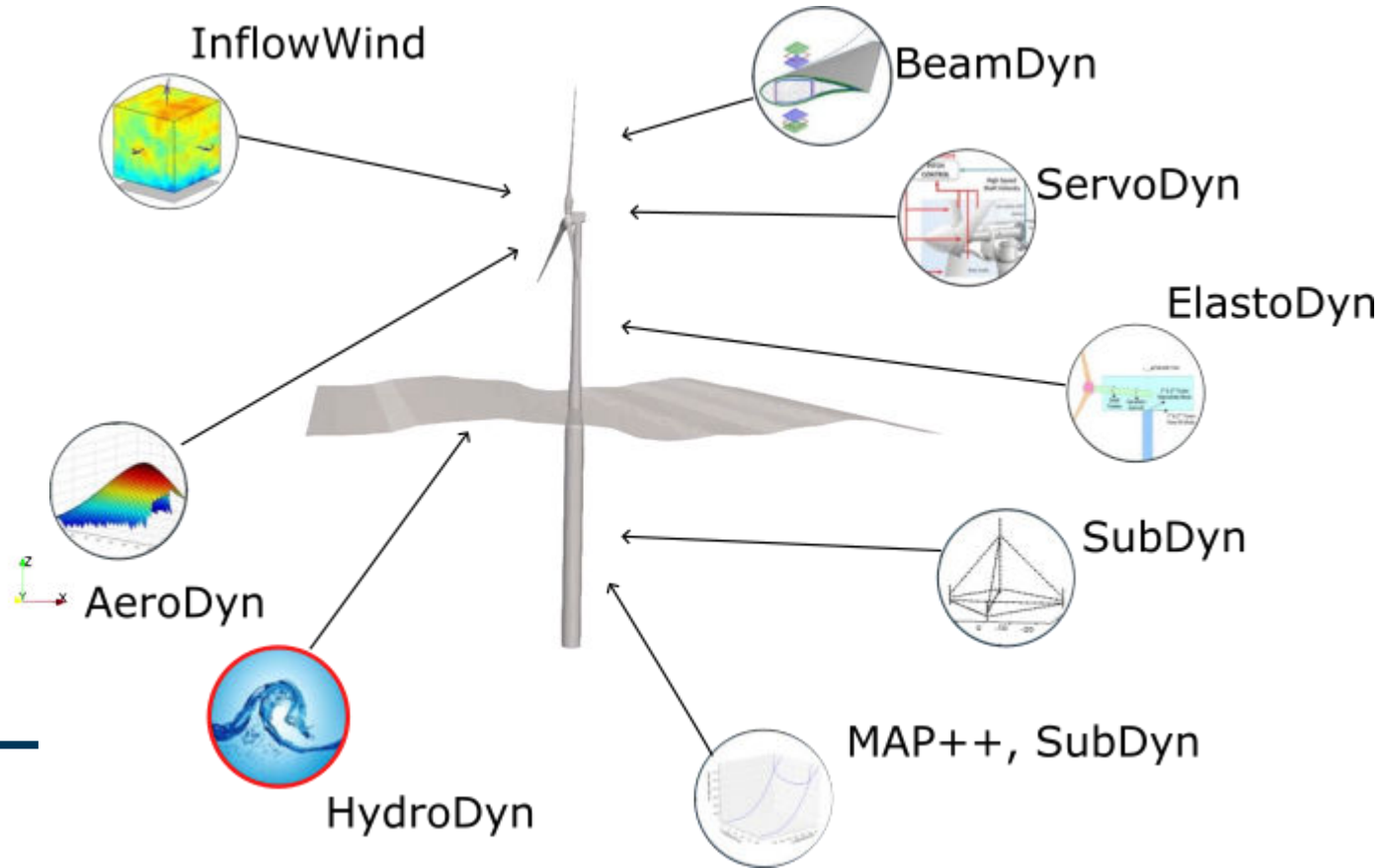
OpenFAST

- OpenFAST: An Open-Source Tool for Wind Turbine Physics-Based Engineering Modeling

- Detailed representation of the wind turbine and exterior solicitations

- Coupled simulation
  - Aero
  - Hydro
  - Elastic
  - Servo

- Modification of HydroDyn

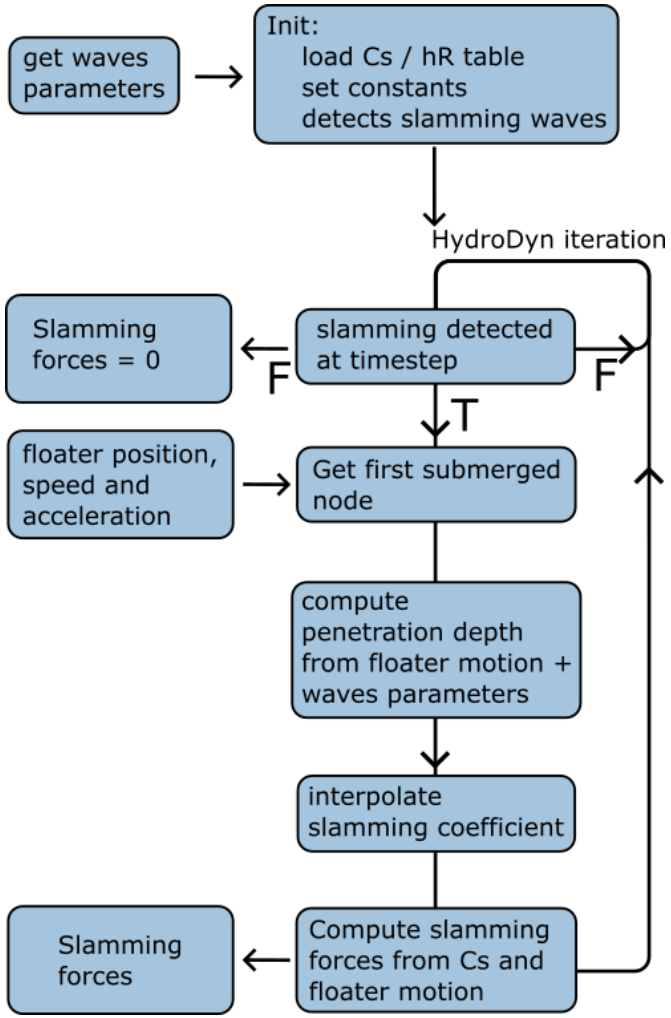


Implementation of slamming forces

- Development of the DIMPACT module integrated into OpenFAST
  - Gets internal variables
  - Incorporates slamming forces

- At initialization
  - Get waves parameter
  - Computes full wave-kinematics
  - Detect slamming waves

- At each iteration
  - Get submerged node
  - Computes penetration depth and slamming coefficient
  - Integrate slamming forces
  - Add those forces to hydrodynamic forces



Implementation of slamming forces

- Slamming waves detection from linear waves
  - Get ‘U’ fluid velocity from internal calculations
  - Computes C the phase velocity from wave number and water depth

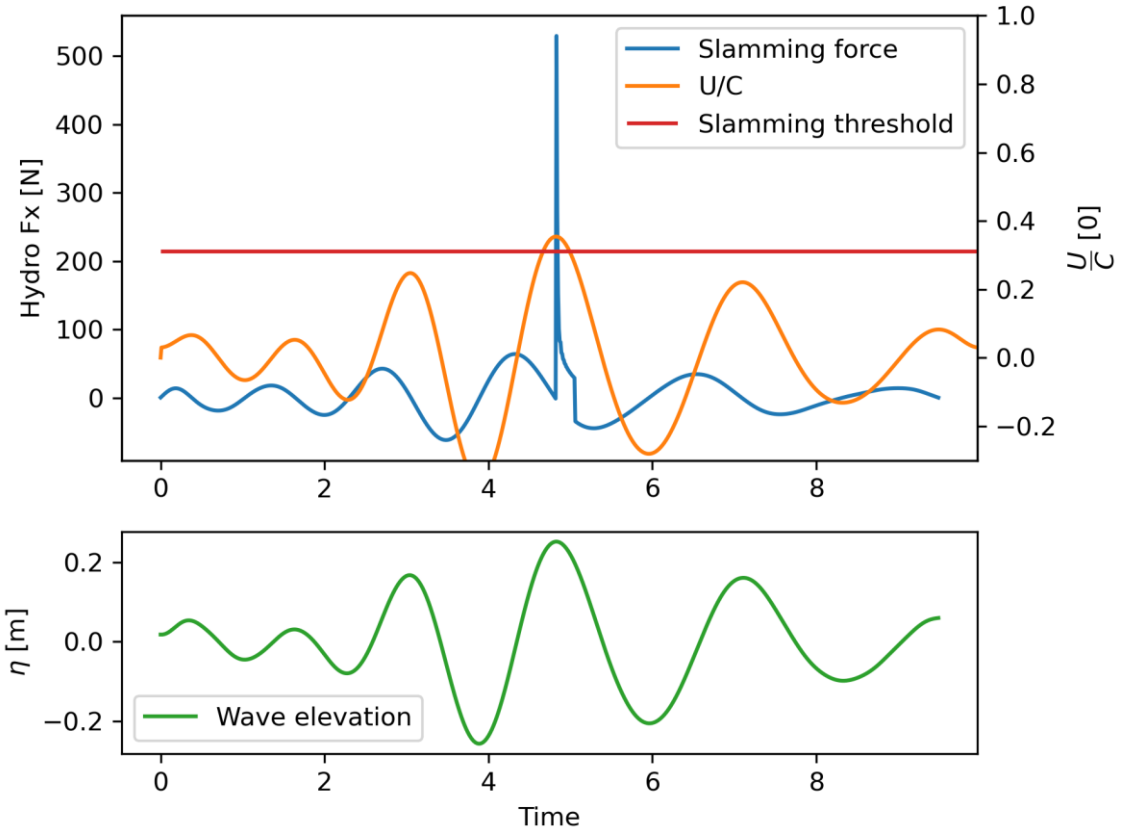
$$C = \sqrt{\frac{g}{k}\tanh(kh)}$$

  - Detection of slamming waves with:

$$\frac{U}{C} > S_c(k)$$

- Implement Slamming forces to HydroDyn

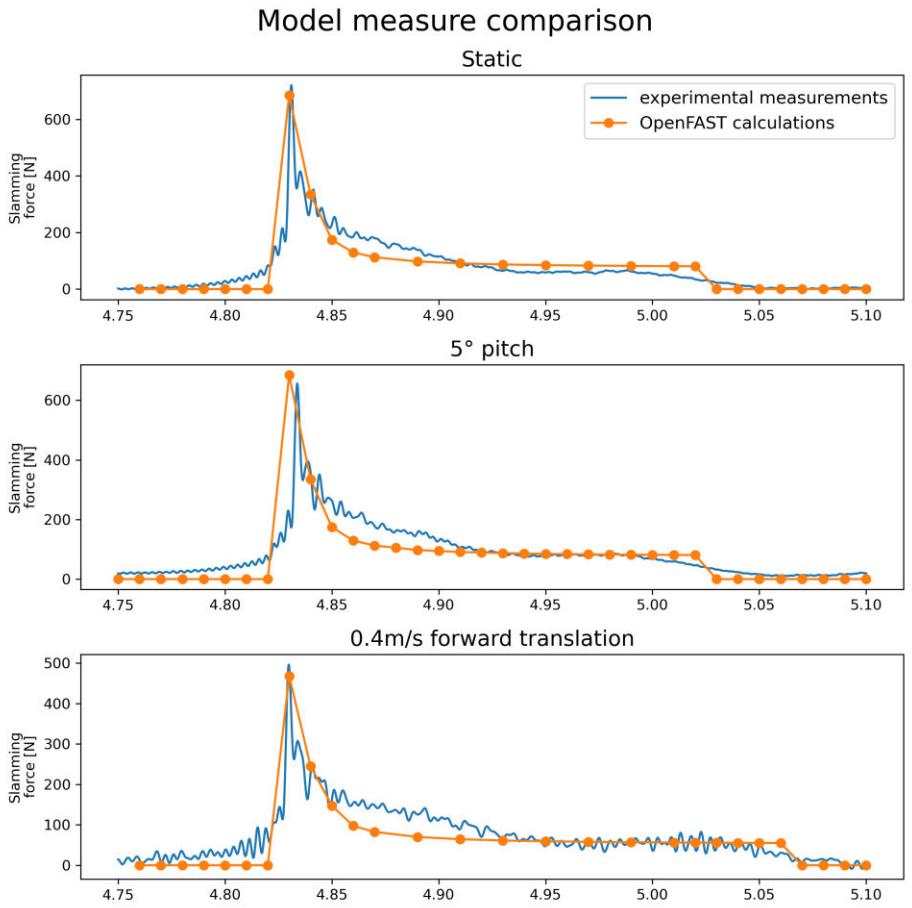
$$F_s = \rho R \int_{(1-\lambda)_{\eta\nu}}^{\eta\nu} C_s(h)(U(z) - U_b)^2 \cos^2\theta dz$$



Implementation of slamming forces

- Validation against experimental data
- Demonstration with HydroDyn standalone reproducing wave basin experiments
- Implementation is a success
- Improvements are still possible
  - Refining computations
  - Test and validate on a fully coupled model
  - Integrate multidirectional sea-states

➢ Observe impact on FOWT and compare with literature



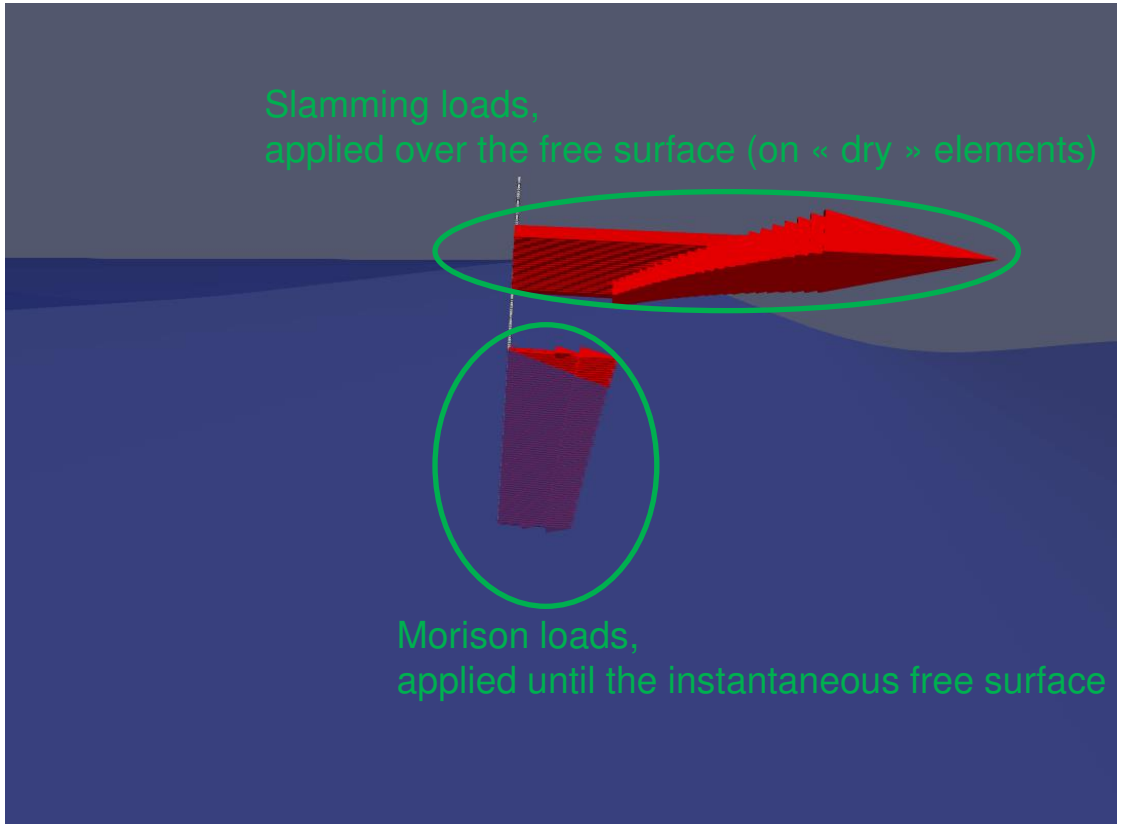
EDF DIEGO implementation

- Main differences:
  - Loads applications
  - Analyze the incoming wave field
  - Considers if the crest is close enough

- Next slides from a presentation by Christophe Peyrard

DIMPACT – Breaking waves loads

Typical distribution of the loads (ex : case DIMPACT-wave 3)



View of the distribution of wave loads at the impact instant

Slamming loads :

- Applied in a « ghost » nonlinear crest
- Applied before the wave crest reaches the centre of the column
- Follow the power law of the velocities based on C
- Depend on the sharp Cs evolution
- Are short in time

Morions loads :

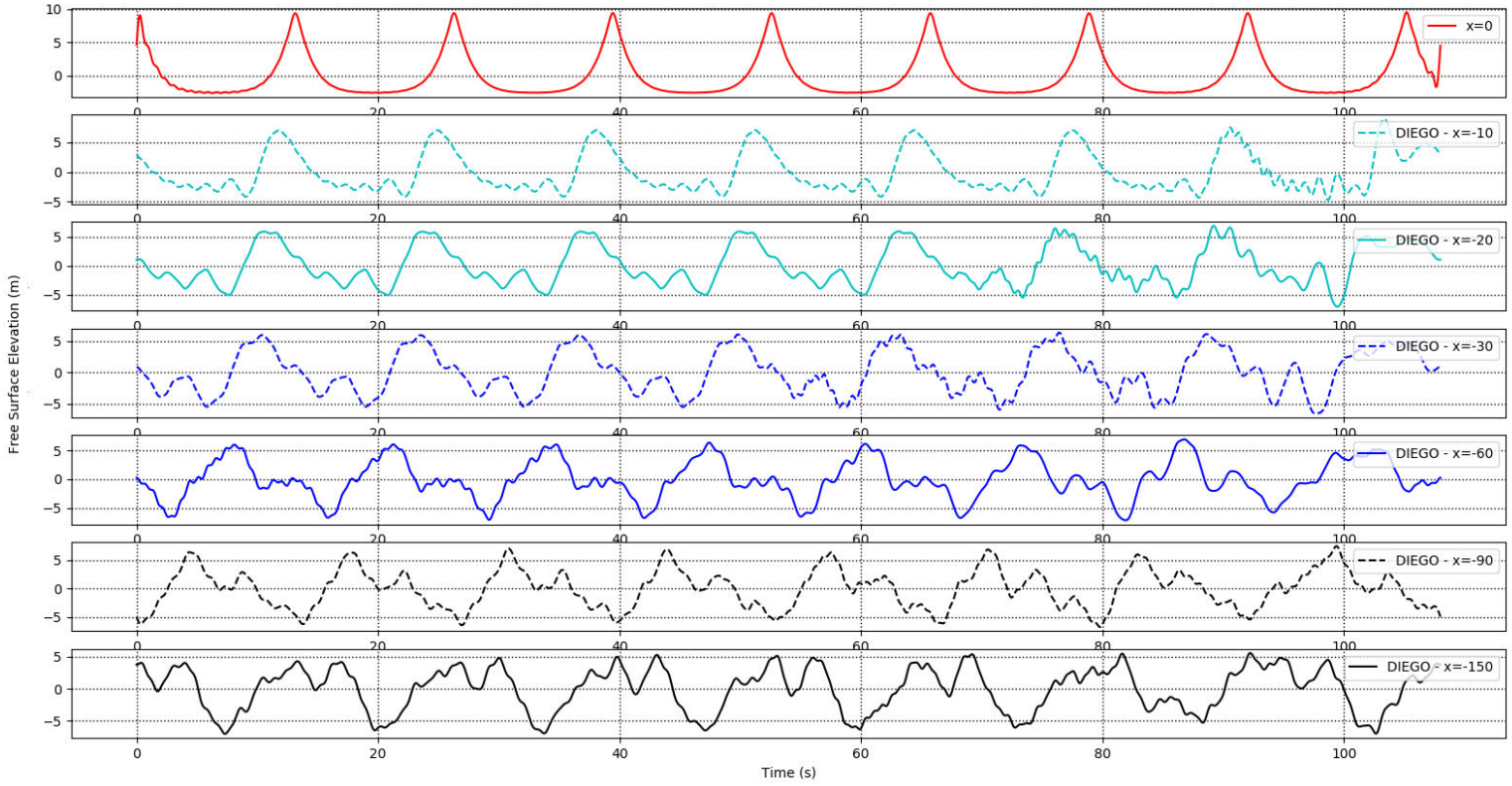
- Applied on the instantaneous wetted elements
- Depend on the local wav kinematics
- Follow the exponential law of the wave kinematics
- Depend on Cd and Ca
- Follow the wave periods (+harmonics)

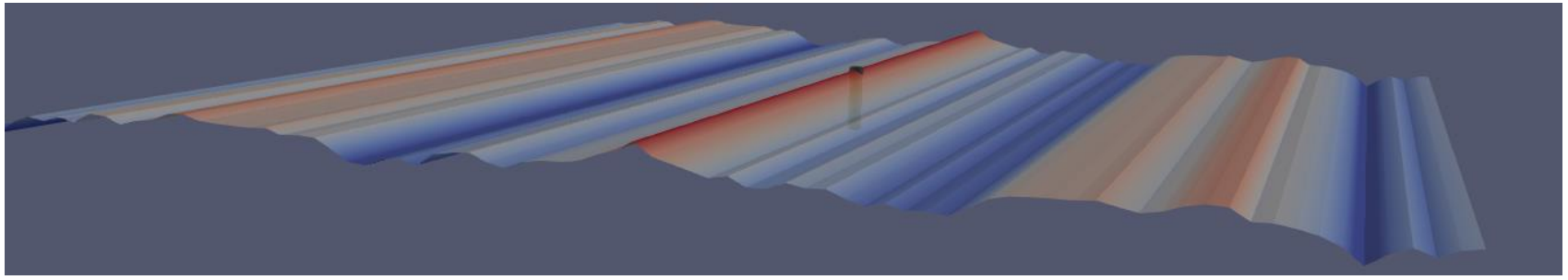
Note : it is important to get a correct phasing between slamming and Morison loads, as engineering models need both

DIMPACT – Breaking waves loads

Case 2 : Benoit 2004

The free surface elevation obtained is the same as the Stream Function at the pile location



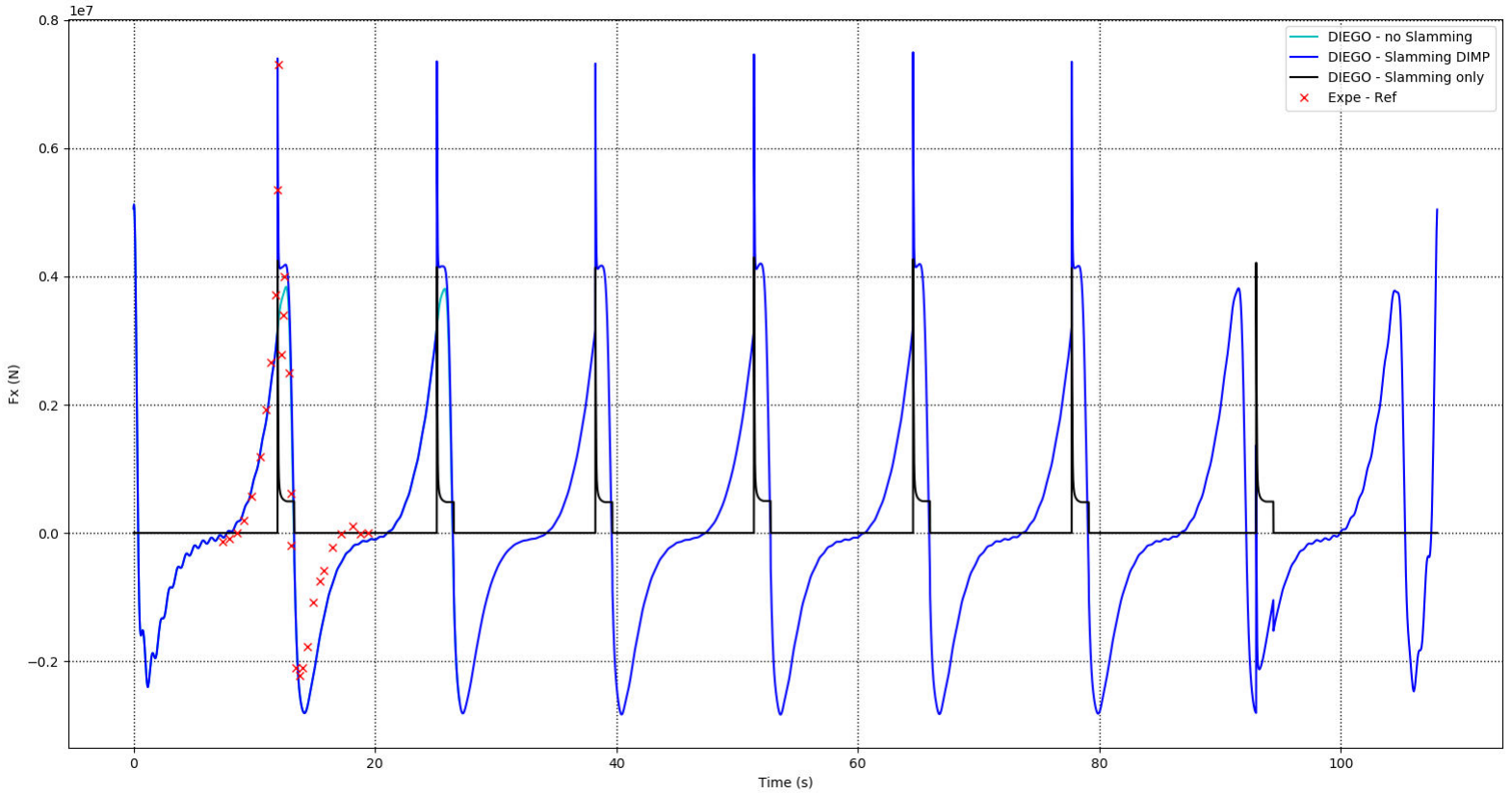


DIMPACT – Breaking waves loads

Case 2 : Benoit 2004

Wave loads :

- Nice prediction of the impact loads
- Repeatability of the loads from one wave to another (was not systematically observed during the development)



Note : I had to change a bit my setting to phase the breaking loads.
Setup : Apply the loads when the wave crest reaches the center of the cylinder – D.
=> I need to implement Paul's methodo