



**HAL**  
open science

## **HOS-ocean: Open-source solver for nonlinear waves in open ocean based on High-Order Spectral method**

Guillaume Ducrozet, Félicien Bonnefoy, David Le Touzé, Pierre Ferrant

### ► **To cite this version:**

Guillaume Ducrozet, Félicien Bonnefoy, David Le Touzé, Pierre Ferrant. HOS-ocean: Open-source solver for nonlinear waves in open ocean based on High-Order Spectral method. *Computer Physics Communications*, 2016, 10.1016/j.cpc.2016.02.017 . hal-01299487

**HAL Id: hal-01299487**

**<https://hal.science/hal-01299487v1>**

Submitted on 8 Apr 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# HOS-ocean: Open-source solver for nonlinear waves in open ocean based on High-Order Spectral method

Guillaume Ducrozet\*, Félicien Bonnefoy, David Le Touzé, Pierre Ferrant

*Ecole Centrale de Nantes, LHEEA Lab. (ECN/CNRS),  
1, rue de la Noë - 44321 Nantes, France*

---

## Abstract

HOS-ocean is an efficient High-Order Spectral code developed to solve the deterministic propagation of nonlinear wavefields in open ocean. HOS-ocean is released as open-source, developed and distributed under the terms of GNU General Public License (GPLv3). Along with the source code, a documentation under wiki format is available which makes easy the compilation and execution of the source files. The code has been shown to be accurate and efficient.

*Keywords:* High-Order Spectral method, Nonlinear waves, Wave propagation, Wave kinematics, Ocean engineering

---

## PROGRAM SUMMARY

*Manuscript Title:* HOS-ocean: Open-source solver for nonlinear waves in open ocean based on High-Order Spectral method

*Authors:* G. Ducrozet, F. Bonnefoy, D. Le Touzé and P. Ferrant

*Program Title:* HOS-ocean

---

\*Corresponding author.

E-mail address: guillaume.ducrozet@ec-nantes.fr

Tel: +33 240 371 645

Fax: +33 240 372 523

*Preprint submitted to Elsevier*

*January 18, 2016*

*Journal Reference:*

*Catalogue identifier:*

*Licensing provisions:* GNU General Public License, version 3

*Programming language:* Fortran

*Computer:* Tested on Intel Xeon E5504 and Intel Core i7

*Operating system:* Any system with a Fortran compiler: tested on Linux, OS X and Windows7

*RAM:* From several MB up to several GB, depending on problem (512x512, M=3: 385 MB) 256x256, M=3: 99MB

*Keywords:* High-Order Spectral method, Nonlinear waves, Wave propagation, Wave kinematics, Ocean engineering

*Classification:* 4.12 (and eventually: 7.7 & 12)

*External routines/libraries:* FFTW 3.3.4 [17] and LAPACK [3]

*Nature of problem:*

HOS-ocean has been developed to study the propagation of highly nonlinear sea-states over large domains and long duration.

*Solution method:*

HOS-ocean is an implementation of the High-Order Spectral method, which solves the problem formulated on the free surface by means of a pseudo-spectral method.

*Restrictions:*

HOS-ocean is dedicated to the propagation of wave fields in infinite and finite constant depth, the evolution over variable bathymetry is not treated. Furthermore, simulations are restricted to non-breaking waves.

*Running time:*

2D simulation of irregular wavefield with  $N_x = 1024$  modes and an HOS order

$M = 5$ :  $t \simeq 2.0 \cdot 10^{-1}$  s per wave period

3D simulation of irregular wavefield with  $N_x = 256$ ,  $N_y = 256$  modes, an HOS order  $M = 3$ :  $t \simeq 10$  s per wave period

## Introduction

Accurate modeling of nonlinear wave propagation and induced kinematics is of particular concern in marine engineering, oceanography, security of marine transportation. . . Nowadays, the main challenge is to simulate large domains over long time periods with nonlinear features taken care of. This led in last decades to the development of several numerical methods able to solve the nonlinear evolution of sea-states. In this context, High-Order Spectral (HOS) model exhibits high efficiency and accuracy thanks to its pseudo-spectral formalism. When dealing with the propagation of open-ocean wavefields over constant water depth, HOS models have been shown to be more efficient than advanced models solving the volume problem with finite-differences discretization [10].

HOS models have been widely used and validated by several authors to study different physical mechanisms: *e.g.* in recent years nonlinear energy transfers [27], modulational instabilities [16], [28], bi-modal seas [29], freak waves [12, 32, 24] . . . among others. This method can consequently be considered mature and accessible to practical engineering applications.

HOS-ocean is an open-source HOS model developed at Ecole Centrale Nantes, LHEEA Lab. (ECN/CNRS). The software is available to download

and contribute on the GitHub platform [2]. The code is developed and re-distributed under the terms of the GNU General Public License as published by the Free Software Foundation. Along with the source code, documentation that describes the compilation and execution of the source files is also distributed. Note that HOS-ocean has been extensively validated on non-linear regular wave propagation [5] and applied to the study of freak waves modeling in [12].

One of the purposes of this code is to encourage other researchers to use HOS. Along with the core of models, post-processing is also delivered. This allows in particular the possibility to easily couple HOS-ocean with CFD (Computational Fluid Dynamics) software for wave-structure interactions modeling.

In the following sections, the HOS formulation implemented in HOS-ocean and associated numerical procedures are described. Sections describing how to compile and run the code are also provided and finally, several study cases are presented as typical applications of the presented numerical model.

## **1. High-Order Spectral method**

In this section, the High-Order Spectral method used in HOS-ocean code is presented. HOS models have been widely used for the study of wave propagation in open domains starting from the original work of [31] & [8] up to the analysis of complex wave phenomena [27, 28, 29, 16], including freak waves [12, 32, 24], among others. Extensions of the original HOS model to include wind forcing effects [30, 19] and jet current effects [25] are also worth to mention. In this section, unless specified, the method is presented in one

horizontal direction (*i.e.* 2DV problem) for conciseness, extension to 3D being straightforward (see [5, 12]).

### 1.1. Formulation

We consider a rectangular fluid domain  $D$  of horizontal dimension  $L_x$  and constant water depth  $h$  associated with a Cartesian coordinate system. Its origin  $O$  is located at one corner of the domain with  $Ox$  representing the horizontal axis and  $Oz$  the vertical one oriented upward with  $z = 0$  located at the mean free surface. We are working under the potential flow theory (assuming the fluid to be incompressible and inviscid and the flow irrotational). Then, continuity equation reduces to the Laplace equation for the velocity potential  $\phi$

$$\nabla^2 \phi + \frac{\partial^2 \phi}{\partial z^2} = 0 \quad \text{in } D \quad (1)$$

with  $\nabla$  denoting the horizontal gradient operator. The boundary conditions will close the system of equations to solve. At first, we consider the free surface boundary conditions which are written, following [33], using surface quantities namely the free surface elevation  $\eta$  and the free surface velocity potential  $\tilde{\phi}(x, t) = \phi(x, z = \eta(x, t), t)$ .  $z = \eta(x, t)$  describes the free surface position, assuming no wave breaking occurs (*i.e.* the free surface is a single valued function of  $x$ ). The kinematic and dynamic free surface boundary conditions then read

$$\frac{\partial \eta}{\partial t} = (1 + |\nabla \eta|^2) W - \nabla \tilde{\phi} \cdot \nabla \eta \quad (2)$$

$$\frac{\partial \tilde{\phi}}{\partial t} = -g\eta - \frac{1}{2} |\nabla \tilde{\phi}|^2 + \frac{1}{2} (1 + |\nabla \eta|^2) W^2 \quad (3)$$

$W(x, t) = \frac{\partial \phi}{\partial z}(x, z = \eta, t)$  is the vertical velocity at the free surface, the only quantity beyond the system (2)-(3) which needs solution in the water bulk. Latter  $W$  will be evaluated thanks to the order consistent HOS scheme of West *et al.* [31] described hereafter (Sec. 1.2). Once the vertical velocity is evaluated, it is possible to advance in time the two unknowns  $\eta$  and  $\tilde{\phi}$  (Sec. 1.4).

In this model, periodic lateral boundary conditions are used, assuming the domain to be infinite

$$\left( \eta, \tilde{\phi}, W \right) (x = 0, t) = \left( \eta, \tilde{\phi}, W \right) (x = L_x, t) \quad (4)$$

The latter associated to Laplace equation (1) and bottom boundary condition

$$\frac{\partial \phi}{\partial z}(x, z = -h, t) = 0 \quad (5)$$

allow us to define a spectral basis  $\Psi_m$  on which the potential is expanded.

$$\begin{aligned} \phi(x, z, t) &= \sum_m A_m(t) \Psi_m(x, z) \\ &= \sum_m A_m(t) \frac{\cosh [k_m (z + h)]}{\cosh (k_m h)} \exp (ik_m x) \end{aligned} \quad (6)$$

with  $k_m = m\Delta k_x = m\frac{2\pi}{L_x}$  the wavenumbers. Similarly, surface quantities are also expressed on a spectral basis allowing the use of Fast Fourier Transforms (FFTs).

$$\tilde{\phi}(x, t) = \sum_m B_m^{\tilde{\phi}}(t) \exp (ik_m x) \quad (7)$$

$$\eta(x, t) = \sum_m B_m^{\eta}(t) \exp (ik_m x) \quad (8)$$

Expansions (7)-(8) are truncated to a given number of modes which defines the corresponding number of points  $N$  in physical space. Some computing operations will be done in physical space (typically products of quantities) and others in transformed (Fourier) space (evaluation of spatial derivatives). Then, the resulting numerical method is pseudo-spectral and exhibits very interesting convergence properties. The corresponding HOS model features high efficiency and accuracy compared to other advanced methods for wave propagation, see [10].

### 1.2. HOS scheme

Knowing the surface quantity  $\eta$  and  $\tilde{\phi}$ , the HOS scheme intends to evaluate the vertical velocity at the free surface  $W(x, t)$ . This procedure relies on a series expansion in wave steepness  $\epsilon$  up to the so-called HOS order  $M$

$$\phi(x, z, t) = \sum_{m=1}^M \phi^{(m)}(x, z, t) \quad (9)$$

with  $\phi^{(m)}$  quantities of order  $\epsilon^m$ . Associated with a Taylor series around  $z = 0$  and collecting terms at each order in wave steepness lead to a triangular system for the different  $\phi^{(m)}$ .

$$\phi^{(1)}(x, 0, t) = \tilde{\phi}(x, t) \quad (10)$$

$$\phi^{(m)}(x, 0, t) = - \sum_{k=1}^{m-1} \frac{\eta^k}{k!} \frac{\partial^k \phi^{(m-k)}}{\partial z^k}(x, 0, t) \quad \text{for } m > 1 \quad (11)$$

This way, the complicated Dirichlet problem for  $\phi(x, z, t)$  on  $z = \eta(x, t)$  is transformed into  $M$  simpler Dirichlet problems for  $\phi^{(m)}(x, z, t)$  on  $z = 0$ . Similar series expansion done on the velocity potential (Eq. (9)) is applied to the vertical velocity  $W$ , leading to another triangular system (for  $W^{(m)}$ )



solved iteratively.

$$W^{(m)}(x, t) = \sum_{k=0}^{m-1} \frac{\eta^k}{k!} \frac{\partial^{k+1} \phi^{(m-k)}}{\partial z^{k+1}}(x, 0, t) \quad (12)$$

Previous set of equations consequently allows to evaluate the free surface vertical velocity of interest

$$W(x, t) = \sum_{m=1}^M W^{(m)}(x, z, t) \quad (13)$$

### 1.3. Dealiasing

HOS-ocean code is based on a pseudo-spectral method meaning that certain operators are treated in the physical space instead of spectral space. This is typically the case of all nonlinear products appearing in the free-surface boundary conditions Eqs. (2)-(3) or in the HOS procedure. These transforms between physical and spectral space are efficient thanks to the use of FFTs but leads to the well-known aliasing phenomena. In order to take care of this problem, we use zero-padding on spectral description of quantities. The number of collocation points has to be chosen to remove aliasing errors on multiple products. Those are at most products of order  $M$ . The extended description of quantities uses the half-rule and is defined on a number of point

$$N_d = \frac{p+1}{2} N \quad (14)$$

Full dealiasing is recommended and is achieved by choosing  $p = M$ . This ensures that the description with  $N$  modes is free from any aliasing errors. However, if needed (especially for computational effort reduction), partial dealiasing is available in the code by choosing lower value for  $p$ . Note that it appears crucial to perform careful dealiasing to preserve method convergence and accuracy as shown in [5]. Finally, we indicate that in 3D (*i.e.* 2

horizontal dimensions), dealiasing has to be performed along both directions  $x$  and  $y$ .

#### *1.4. Time integration*

Free surface boundary conditions Eqs. (2)-(3) allow the time-stepping of the two unknowns  $\eta$  and  $\tilde{\phi}$ . This system of equations appear very stiff for high frequencies, which may lead to small time-steps. Following [18], thanks to an adequate change of variables, it is possible to solve the nonlinear part of the equations as a problem in itself and not as an adjustment of the linear solution. The linear part of the equations is then analytically integrated while nonlinear evolution of the system is computed numerically. This is achieved thanks to an efficient 4<sup>th</sup> order Runge-Kutta Cash-Karp scheme with adaptive step size [6]. A desired accuracy (or so-called tolerance) is specified and the time-step is automatically chosen to achieve this accuracy. Typical values of the tolerance are in the range  $[10^{-5}; 10^{-7}]$  to achieve convergence. This choice depends mainly on the nonlinearity of the wave field and the duration of the simulation.

#### *1.5. Initial conditions*

With such an approach, the key point lays in the definition of an appropriate initial condition. The initial free surface elevation  $\eta$  and free surface velocity potential  $\tilde{\phi}$  have to be determined before propagating the corresponding wave field in the HOS-ocean model. Different possibilities are offered within the model.

### 1.5.1. Regular waves

The nonlinear regular wave solution of Rienecker & Fenton [23] (stream function theory) may be specified to initialize the HOS-ocean solution. This is mainly used for validation and to characterize the accuracy of the solution as well as the convergence properties of the method. Especially, a benchmarking procedure is set up in HOS-ocean code, that uses this initialization. Note that within HOS-ocean, the stream function solution is stored in a specific file for each wave condition.

### 1.5.2. Irregular waves

For completeness, the initialization of irregular wavefield is presented in 3D configuration, *i.e.* directional sea. For the definition of irregular sea-states, we use a classical superposition of linear components, each component having its own frequency and direction. The amplitude of each component is determined from a directional spectrum  $S(\omega, \theta)$ . The relation between the amplitude in the discrete Fourier domain  $B_{mn}$  and the spectrum is

$$\frac{1}{2} |B_{mn}^\eta|^2 = S(k_x, k_y) \Delta k_x \Delta k_y = \frac{\partial \omega}{\partial k} \frac{1}{k} S(\omega, \theta) \Delta k_x \Delta k_y \quad (15)$$

with  $\Delta k_x$  and  $\Delta k_y$  the modal discretization in both directions. This equation allows to calculate the norm  $|B_{mn}^\eta(t = 0)|$  for the desired directional spectrum  $S(\omega, \theta)$ . Then, the phase of each component  $B_{mn}^\eta(t = 0)$  is determined by a random number so that the phases assume uniform distribution over the range  $[0, 2\pi]$ . The free surface potential is linearly evaluated from the elevation  $B_{mn}^{\tilde{\phi}} = -i \frac{\omega_{mn}}{g} B_{mn}^\eta$  where  $\omega_{mn}$  is given by the adequate dispersion relation. Finally, the initial wavefield is constructed with an inverse Fourier transform.

HOS-ocean provides the possibility to use a directional spectrum form  $S(\omega, \theta) = F(\omega).G(\theta)$  defined with:

- i) a JONSWAP spectrum, specifying the peak period  $T_p = \frac{2\pi}{\omega_p}$ , the significant wave height  $H_s$  and the shape factor  $\gamma$

$$F(\omega) = \alpha_J H_s^2 \omega_p^4 \omega^{-5} \exp \left[ -\frac{5}{4} \left( \frac{\omega}{\omega_p} \right)^{-4} \right] \gamma^{\exp \left[ -\frac{(\omega - \omega_p)^2}{2\sigma^2 \omega_p^2} \right]} \quad (16)$$

with  $\sigma = \begin{cases} 0.07 & \text{for } \omega < \omega_p \\ 0.09 & \text{for } \omega \geq \omega_p \end{cases}$  and  $\alpha_J$  chosen to obtain the correct significant wave height.

- ii) a specific directional spreading:

$$G(\theta) = \frac{1}{\beta} \left[ \cos \left( \frac{\pi\theta}{2\beta} \right) \right]^2 \quad (17)$$

### 1.5.3. Relaxation period

It has been demonstrated in [7], that the definition of linear initial wavefields for fully-nonlinear computations can lead to numerical instabilities. In order to overcome this problem, a transition period is introduced through the use of a relaxation scheme, allowing a smooth transition from linear initial conditions to fully nonlinear computation. The transition period is described by its duration  $T_a$  and the parameter  $n$ . The free surface boundary conditions described by Eqs. (2)-(3) are rewritten as

$$\frac{\partial \tilde{\phi}}{\partial t} + g\eta = F \quad , \quad \frac{\partial \eta}{\partial t} - W^{(1)} = G \quad (18)$$

with  $W^{(1)}$  the linearized free surface vertical velocity. Then, the relaxation

period is applied on the nonlinear terms  $F$  and  $G$  as follows

$$\frac{\partial \tilde{\phi}}{\partial t} + g\eta = F \left( 1 - \exp \left[ - \left( \frac{t}{T_a} \right)^n \right] \right) \quad (19)$$

$$\frac{\partial \eta}{\partial t} - W^{(1)} = G \left( 1 - \exp \left[ - \left( \frac{t}{T_a} \right)^n \right] \right) \quad (20)$$

### 1.6. Post-processing

This section describes briefly the different possibilities for post-processing of HOS-ocean outputs offered within open-source release of the code. For a fine analysis of physical processes at play during the nonlinear wave propagation, it appears useful to compute complementary information in addition to the two unknowns solved by HOS model  $\eta$  and  $\tilde{\phi}$ .

#### 1.6.1. Wavefield analysis

HOS-ocean enables the study of the evolution in time of the free surface elevation  $\eta$ . It appears interesting during post-processing to give access to some basic information useful for any analysis, such as the different moments of this free surface elevation: mean, standard deviation, skewness, kurtosis.

At the same time, several studies need the extraction of individual waves inside the computed sea state (we refer for instance to [12] for the study of freak waves). Thus, post-processing of 2D wave fields allows a wave-by-wave analysis using zero-crossing to extract individual waves and possible useful information:  $H_{1/3}$ ,  $H_{max}$ , mean wavelength  $\lambda_0 \dots$

In three dimensions, we choose to define the height of a wave as its height in the direction of propagation. We first perform the same kind of analysis than in two dimensions (zero-crossing along  $x$ -axis which is the mean direction of propagation of our wavefield). This is performed recursively with

respect to the  $y$ -direction to get the height at the wave peak location in  $y$ . Thus, we obtain the characteristics of each wave and can deduce those of the whole wavefield:  $H_{1/3}$ ,  $H_{max}$ ,  $\lambda_0 \dots$ . At the same time, we have also access to transverse wavelength  $\lambda_y$ , height  $H_y$  that are interesting in terms of incidence of waves on a structure for example.

### 1.6.2. Kinematics computation within fluid domain

HOS models use a surface formulation where only quantities on the free surface are advanced in time, namely  $\eta$  and  $\tilde{\phi}$ . This allows a reduction of numerical complexity and high efficiency. However, it appears necessary in some configurations to access to the volume information, usually the kinematics/pressure inside domain. For the efficiency of this evaluation, preliminary computations are done during the main HOS-ocean execution. Then, post-processing enables the reconstruction of those quantities.

The main idea is to use a modal description of the volume information (for instance modal coefficients  $A_m$  for the velocity potential  $\phi(x, z, t)$  in Eq. (6)). Once they are known, it is possible to reconstruct easily this quantity wherever needed in the fluid domain. Different strategies exist to compute from the quantity known at the free surface the corresponding modal coefficients of the volume field. For instance, the HOS scheme solves this problem for  $\phi(x, z, t)$ . But, it has been shown in [4] that in order to achieve a good accuracy for velocity fields, more advanced procedures need to be used. HOS-ocean provides the possibility to perform a direct inversion of the full system. However, the approximate solution using the so-called  $H_2$  operator developed in the context of Dirichlet to Neumann Operator (DNO) models

is preferred (especially in 3D where the direct inversion is not possible) [4]. The accuracy of the strategy has been validated in details in [9].

This way, it is possible to compute easily during post-processing the kinematics induced by the nonlinear waves simulated in HOS-ocean. This enables the possible coupling between HOS-ocean and CFD solvers for the evaluation of wave-structure interactions. Note that the methodology has been used at ECN for coupling between HOS and RANS solver with the SWENSE method (Spectral Wave Explicit Navier-Stokes Equations). Details about the SWENSE coupling strategy are given in [20, 21]. HOS-ocean has been also coupled to a Smooth Particle Hydrodynamics (SPH) model to study violent wave-structure interactions inducing wave breaking, surface reconexions. . . see [22].

Note that the proposed post-processing allows to compute velocity and pressure maps in a given domain. For a specific coupling of HOS-ocean, the quantities to reconstruct have to be adapted for the problem solved but the core is the one proposed in this post-processing program.

## **2. Program documentation**

Figure 1 presents a schematic view of the structure of HOS-ocean code. The details of the different source-files are provided in next sections.

### *2.1. Source files*

#### *2.1.1. HOS-ocean*

A set of Fortran files need to be compiled in order to create HOS-ocean binary. Here, all the files are listed. Table 1 presents the different source files

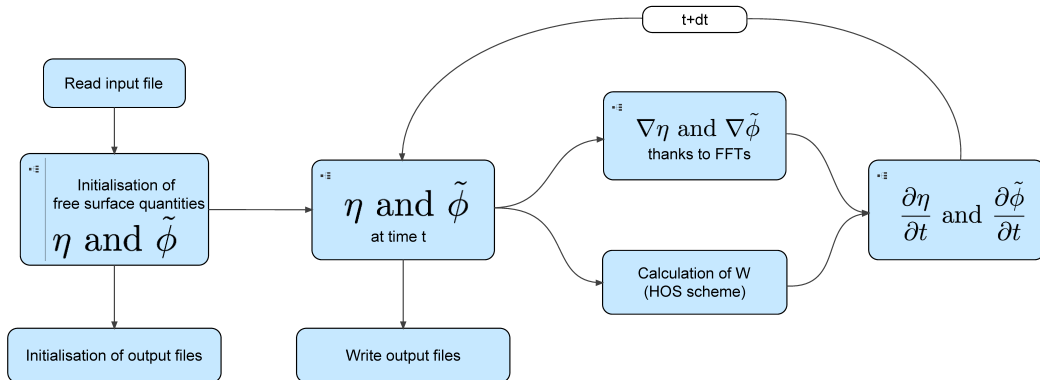


Figure 1: Schematic view of time-stepping in HOS-ocean

present in HOS-ocean code and their brief description.

In order to have a clear view of the different dependencies for HOS-ocean program, we provide a dependency matrix in Tab. 2. Value of 1 indicates that the source file on the line depends on the source file in the column and 0 is allocated to the diagonal of the matrix.

### 2.1.2. Post-processing & Benchmarking

Tables 3 & 4 give the list of the specific files used for the post-processing and the benchmarking respectively.

### 2.2. Compilation

The code can be compiled on any computer architecture. One only needs a Fortran compiler (for instance GNU's gfortran). Makefiles are provided with the source code for compiling HOS-ocean (main program, benchmarking and post-processing) in Linux/Unix environment. These make use of the



|                              |   |
|------------------------------|---|
| HOS-ocean source files       |   |
| <b>HOS_ocean.f90</b>         | Main program for HOS-ocean computations   |
| <b>energy_calc.f90</b>       | Evaluation of free-surface wave fields energy   |
| <b>Bivar.f90</b>             | Bivariate interpolation of irregular data   |
| <b>fftw.f03</b>              | FFTW automatically generated parameter file   |
| <b>fourier_r2c_FFTW3.f90</b> | Necessary features for the use of FFTW 3.3.4 in HOS-ocean with comprehensive interface and normalization  |
| <b>filters.f90</b>           | Definition of function used for dealiasing and filtering of highest modes   |
| <b>initial_condition.f90</b> | Initialization of HOS-ocean computation   |
| <b>input_HOS.f90</b>         | Reading of input files  |
| <b>linear_wave.f90</b>       | Computation of linear dispersion relation   |
| <b>maths.f90</b>             | Mathematical functions  |
| <b>nrutil_tmp.f90</b>        | Utility functions to swap two elements  |
| <b>output.f90</b>            | Initialization of output files and writing at specified time-step   |
| <b>ramp.f90</b>              | Definition of ramp functions  |
| <b>resol_HOS.f90</b>         | Computation of vertical velocity (HOS method) and free surface boundary conditions  |
| <b>RF_solution.f90</b>       | Computation of the solution of Rienecker & Fenton using a set of coefficients describing $\eta$ and $\tilde{\phi}$ obtained from a specific program from LHEEA Lab. |
| <b>runge_kutta.f90</b>       | Time integration with Runge-Kutta 5(4) Cash & Karp scheme. Linear part of equations is integrated analytically  |
| <b>type.f90</b>              | Definitions of variables types and useful constants   |
| <b>variables_3D.f90</b>      | Definition of number of points/nodes, HOS order and all different global variables used in HOS-ocean  |

|                       | HOS-ocean.f90 | type.f90 | variables_3D.f90 | output.f90 | runge_kutta.f90 | energy_calc.f90 | fourier_r2c_FFTW3.f90 | ramp.f90 | input_HOS.f90 | initial_condition.f90 | filters.f90 | linear_wave.f90 | Bivar.f90 | RF_solution.f90 | maths.f90 | nrutil_tmp.f90 | velocities.f90 | resol_HOS.f90 |
|-----------------------|---------------|----------|------------------|------------|-----------------|-----------------|-----------------------|----------|---------------|-----------------------|-------------|-----------------|-----------|-----------------|-----------|----------------|----------------|---------------|
| HOS-ocean.f90         | 0             | 1        | 1                | 1          | 1               | 1               | 1                     | 1        | 1             | 1                     | 1           | 1               | .         | .               | .         | .              | .              | .             |
| type.f90              | .             | 0        | .                | .          | .               | .               | .                     | .        | .             | .                     | .           | .               | .         | .               | .         | .              | .              | .             |
| variables_3D.f90      | .             | 1        | 0                | .          | .               | .               | .                     | 1        | .             | .                     | .           | .               | .         | .               | .         | .              | .              | .             |
| output.f90            | .             | 1        | 1                | 0          | .               | .               | 1                     | .        | 1             | .                     | .           | .               | .         | .               | .         | .              | .              | .             |
| runge_kutta.f90       | .             | 1        | 1                | .          | 0               | .               | 1                     | .        | .             | .                     | .           | .               | .         | .               | .         | .              | 1              | 1             |
| energy_calc.f90       | .             | 1        | 1                | .          | .               | 0               | 1                     | .        | .             | .                     | .           | .               | .         | .               | .         | .              | .              | .             |
| fourier_r2c_FFTW3.f90 | .             | 1        | 1                | .          | .               | .               | 0                     | .        | .             | .                     | .           | .               | .         | .               | .         | .              | .              | .             |
| ramp.f90              | .             | 1        | .                | .          | .               | .               | .                     | 0        | .             | .                     | .           | .               | .         | .               | .         | .              | .              | .             |
| input_HOS.f90         | .             | 1        | 1                | .          | .               | .               | .                     | .        | 0             | .                     | .           | .               | .         | .               | .         | .              | .              | .             |
| initial_condition.f90 | .             | 1        | 1                | .          | 1               | 1               | 1                     | .        | 1             | 0                     | .           | 1               | 1         | 1               | .         | .              | .              | .             |
| filters.f90           | .             | 1        | 1                | .          | .               | .               | .                     | .        | .             | .                     | 0           | .               | .         | .               | .         | .              | .              | .             |
| linear_wave.f90       | .             | 1        | .                | .          | .               | .               | .                     | .        | .             | .                     | .           | 0               | .         | .               | 1         | .              | .              | .             |
| Bivar.f90             | .             | .        | .                | .          | .               | .               | .                     | .        | .             | .                     | .           | .               | 0         | .               | .         | .              | .              | .             |
| RF_solution.f90       | .             | 1        | 1                | .          | .               | .               | .                     | .        | .             | .                     | .           | .               | .         | 0               | .         | .              | .              | .             |
| maths.f90             | .             | 1        | .                | .          | .               | .               | .                     | .        | .             | .                     | .           | .               | .         | .               | 0         | .              | .              | .             |
| nrutil_tmp.f90        | .             | 1        | .                | .          | .               | .               | .                     | .        | .             | .                     | .           | .               | .         | .               | .         | 0              | .              | .             |
| velocities.f90        | .             | 1        | 1                | .          | .               | .               | 1                     | .        | .             | .                     | .           | .               | .         | 1               | .         | .              | 0              | 1             |
| resol_HOS.f90         | .             | 1        | 1                | .          | .               | .               | 1                     | .        | .             | .                     | 1           | .               | .         | .               | .         | .              | .              | 0             |

Table 2: Dependency matrix for HOS-ocean

| Specific post-processing source files |  |
|---------------------------------------|--|
| <b>Post_processing.f90</b>            | Main program for post-processing HOS-ocean results                   |
| <b>analysis_wavefield.f90</b>         | Analysis (wave-by-wave) of the wavefield and freak waves detection   |
| <b>input_post_process.f90</b>         | Reading of input file for post-processing                            |
| <b>output_post_process.f90</b>        | Initialization of output files and writing at each time-step         |
| <b>read_files.f90</b>                 | Reading of files that will be post-processed                         |
| <b>reconstruction.f90</b>             | Reconstruction of the volumic wavefield informations                 |
| <b>variables_post_process.f90</b>     | Definition of the different global variables used in post-processing |

Table 3: List of source files used for post-processing

| Specific benchmarking source files |   |
|------------------------------------|---|
| <b>check_DYue.f90</b>              | Main program for benchmarking HOS core routines                   |
| <b>variables_3D_DYue.f90</b>       | Definition of the different global variables used in benchmarking |

Table 4: List of source files used for benchmarking

binary `makedepf90` to extract automatic dependencies. Following commands can be used

- Main program: `make` command executed at the root of the project will create the binary `HOS-ocean`
- Benchmarking: `make -f makefile_test_DYue` will generate the binary `check_W_DYue`
- Post-processing: `make` command executed in directory `sources/PostProcessing` will create the binary `Post_Processing`

Compilation has been tested on different Unix/Linux platforms as well as in Windows environment.

### *2.3. Running HOS-ocean*

HOS-ocean has been developed for command-line run with an input file containing all specifications needed. All output files will be created in a directory `Results` that has to be created before the run by the user.

#### *2.3.1. Important parameters*

At first, for the compilation, user has to set values of integers `M` (the HOS order), `n1`, `n2` (the number of points in  $x$  and  $y$  directions respectively), `p1` and `p2` (the parameters used for dealiasing (as seen in Sec. 1.3) in  $x$  and  $y$  directions respectively) in `variables_3D.f90`:

- For a **2D simulation**,
  - Compile with `n2=1` and `p2=1` to adjust the memory allocation to minimum

- If partial dealiasing is used, compile with `p1` set to maximal required value (total dealiasing is obtained with `p1=M` but it can be reduced if `p1` is further set to a value below `M`), see Sec. 1.3
- For a **3D simulation**,
  - Compile with `n2 ≠ 1` and `p2` set to required value
  - If partial dealiasing is used in x-direction, compile with `p1` set to maximal required value (total dealiasing is obtained with `p1=M` but it can be reduced if `p1` is further set to a value below `M`)
  - If partial dealiasing is used in y-direction, compile with `p2` set to maximal required value (total dealiasing is obtained with `p2=M` but it can be reduced if `p2` is further set to a value below `M`)

### 2.3.2. *Input file*

The input file is assumed to be named `input_HOS.dat`. Table 5 describes the different parameters accessible in this input file.

### 2.3.3. *Output files*

Depending on the choices made in the input file (see Tab. 5), different output files are created. They are located in a specific folder `Results` with a specific format dedicated to Tecplot visualization. Input file also defines if outputs are dimensional or non-dimensional quantities. Following files may be created

- `3d.dat` gives the 3D free surface quantities ( $\eta$  and  $\tilde{\phi}$ ) as function of time

|                                       |   |
|---------------------------------------|---|
| <b>i_case</b>                         | Choice of initial condition (regular wave, irregular ...)   |
| Geometry of the horizontal domain     |   |
| <b>xlen</b>                           | Length of the domain along x-direction  |
| <b>ylen</b>                           | Length of the domain along y-direction (if 3D case)   |
| Time integration                      |   |
| <b>T_stop</b>                         | Duration of the simulation  |
| <b>f_out</b>                          | Sampling frequency for output files   |
| <b>toler</b>                          | Tolerance of Runge-Kutta Cash & Karp scheme (adaptive time-stepping)  |
| <b>n</b>                              | $n$ parameter in relaxation scheme on nonlinear terms in free surface boundary conditions ( <i>cf.</i> Dommermuth [7])              |
| <b>Ta</b>                             | $T_a$ parameter (duration) in relaxation scheme on nonlinear terms in free surface boundary conditions ( <i>cf.</i> Dommermuth [7]) |
| Physical dimensional parameters       |   |
| <b>grav</b>                           | Gravity acceleration ( $m.s^{-2}$ )   |
| <b>depth</b>                          | Water depth ( $m$ )   |
| Irregular waves definition (i_case=3) |   |
| <b>Tp_real</b>                        | Peak period ( $s$ )   |
| <b>Hs_real</b>                        | Significant wave height ( $m$ )   |
| <b>gamma</b>                          | $\gamma$ factor in JONSWAP spectrum (Eq. (16))  |
| <b>beta</b>                           | $\beta$ factor in the definition of directional spreading (Eq. (17))  |
| Definition of output files            |   |
| <b>tecplot</b>                        | Tecplot version used for output files format  |
| <b>i_out_dim</b>                      | Output are dimensional (=1) or non-dimensional (=0)   |
| <b>i_3d</b>                           | File 3d.dat created if equal to 1   |
| <b>i_a_3d</b>                         | File a_3d.dat created if equal to 1   |
| <b>i_2d</b>                           | Files 2d.dat or 2dpt.dat created if equal to 1 or 2 respectively  |
| <b>i_prob</b>                         | File probes.dat created if equal to 1   |
| <b>i_sw</b>                           | File modes_HOS_SWENSE.dat created if equal to 1   |

Table 5: Description of input parameters for HOS.

- `a_3d.dat` gives the modal amplitudes of free surface quantities ( $\eta$  and  $\tilde{\phi}$ ) as function of time
- `2d.dat` gives the 2D free surface quantities ( $\eta$  and  $\tilde{\phi}$ ) as function of time
- `2dpt.dat` gives the 2D+t free surface elevation
- `vol_energy.dat` gives the temporal evolution of volume and energy during the simulation
- `modes_HOS_SWENSE.dat` gives the file containing modal information of volumic HOS field for visualization, physical interpretation of wave-field...or possible coupling between HOS and wave-structure interactions model for instance (SWENSE method): see post-processing section
- `probes.dat` gives the free surface elevation at specific location given in file `prob.inp`
- `phase_shift.dat` gives the phase shift between computed solution and reference Rienecker & Fenton solution

#### *2.4. Post-processing*

`Post_processing` binary will analyze the HOS-ocean outputs and perform some analysis with a given input file named `input_post_process.dat`. This input file is detailed in Tab. 6 which describes the different constants to specify.

| Choice of post-processing options |  |
|-----------------------------------|--|
| <b>i_ana</b>                      | Wavefield general features (=1) with 2D wave-by-wave analysis (=2) or 3D (=3)                                      |
| <b>i_card</b>                     | Evaluation of velocity and pressure maps inside fluid domain with cartesian grid (=1) or boundary fitted mesh (=2) |
| <b>T_start</b>                    | Starting time of post-processing   |
| <b>T_stop</b>                     | Stopping time of post-processing   |
| Information on kinematics output  |  |
| <b>x_min</b>                      | Minimum x location of velocity/pressure maps   |
| <b>x_max</b>                      | Maximum x location of velocity/pressure maps   |
| <b>y_min</b>                      | Minimum y location of velocity/pressure maps (if 3D)   |
| <b>y_max</b>                      | Maximum y location of velocity/pressure maps (if 3D)   |
| <b>z_min</b>                      | Minimum z location of velocity/pressure maps   |
| <b>z_max</b>                      | Maximum z location of velocity/pressure maps (if cartesian grid)   |
| <b>i_zvect</b>                    | Number of points in the z direction  |
| Input files information           |  |
| <b>tecplot</b>                    | Version of tecplot used to write following input files   |
| <b>file_3d</b>                    | Path defining the file used for wavefield analysis ( <b>3d.dat</b> generated by HOS-ocean)                         |
| <b>file_mod</b>                   | Path defining the file used for velocity/pressure map ( <b>modes_HOS_SWENSE.dat</b> generated by HOS-ocean)        |

Table 6: Description of input file parameters for post-processing



### 2.4.1. Wavefield analysis

The wavefield analysis is performed on the `3d.dat` files generated by HOS-ocean code. This analysis intends to provide interesting information about the computed sea state. Different output files are created that are described hereafter.

- `Analysis.dat` contains the general information about the wave field simulated with HOS-ocean
  - Different moments of the free surface are always computed
  - `i_ana = 2` or `i_ana = 3` provides in addition in this file  $H_{1/3}$ ,  $H_{max}$ ,  $A_{max}$  (the maximum crest amplitude) and the number of freak waves detected
- `freak_waves.dat` gives the free surface profile of each detected freak wave in the domain
- `Caract_freak_waves.dat` gives the characteristics of detected freak waves (size, amplitude ...)

### 2.4.2. Kinematics inside fluid domain

HOS-ocean is able to create a modal description of some volume information in the fluid domain. All necessary information to compute velocities, pressure, acceleration (or whatever derivatives of these quantities) are included in the file named `modes_HOS_SWENSE.dat` generated by HOS-ocean code. Different output files are created and give examples of representation of velocities and pressure field (`i_card`  $\neq$  0).

- Cartesian grid, `i_card = 1`
  - `data_VP_card.dat` gives useful informations about the map created in `VP_card.dat`. This is used at ECN as the file containing coupling information.
  - `VP_card.dat` contains the velocities and pressure map. The mesh in  $z$  direction is defined as constant  $z$  for all  $x$  and  $y$ . This file is adapted to Tecplot visualization.
- Boundary-fitted mesh, `i_card = 2`
  - `VP_card_fitted.dat` contains the velocities and pressure map in the case `i_card=2`, i.e. when a boundary fitted (free surface) mesh is used. This file is adapted to Tecplot visualization.

### 3. Results

This section presents different results obtained with HOS-ocean code. At first, validation results obtained with the benchmarking procedure included in the code release are presented. They are dealing with the accuracy of the model when studying a nonlinear regular wave propagation. Then, an example of application of the HOS-ocean model to the propagation of 3D irregular sea-states is provided. In the end, the computation of kinematics inside the fluid domain is demonstrated for a 2D irregular wave field exhibiting an extreme event.

#### 3.1. Benchmarking-validation

The nonlinear potential flow solver HOS-ocean presented in this paper uses fully nonlinear free surface boundary conditions Eqs. (2)-(3) to advance

in time the two unknowns  $\eta$  and  $\tilde{\phi}$ . With the chosen formulation, the accuracy is quantified by the one achieved on the vertical velocity at the free surface, evaluated by the so-called HOS method.

Following the test made by Dommermuth and Yue [8] (only the highest steepness  $ka = 0.4$  is presented), we provide the reference  $\eta$  and  $\tilde{\phi}$  from the model of Rienecker and Fenton [23] and evaluate the vertical velocity with the HOS scheme. The absolute error  $\epsilon = \max |W - W_{RF}|$  is computed ( $W_{RF}$  is the reference vertical velocity obtained from [23]). Fig. 2 presents the results of  $\log_{10}(\epsilon)$  as function of the number of points  $N$  and HOS order  $M$  with full dealiasing. For a more comprehensive reading of the figure, note that  $N$  is varying in the range  $N \in [4; 128]$  with powers of two values and  $M \in [2; 18]$  with steps of two.

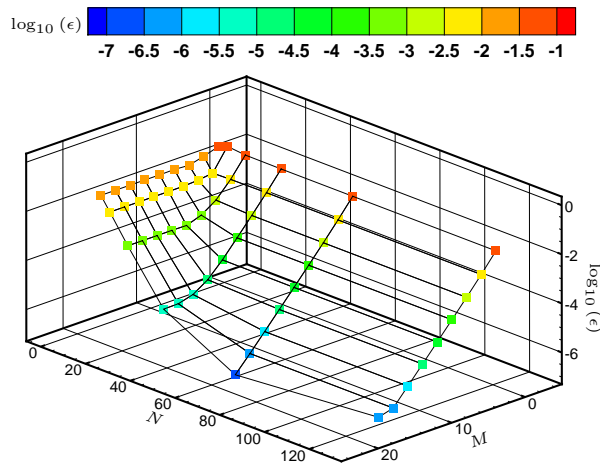


Figure 2: Convergence on the evaluation of vertical velocity  $W$  for highly nonlinear regular wave  $ka = 0.4$

It is clear from this figure that HOS-ocean code allows an accurate evalu-

ation of the vertical velocity at the free surface, even with a very steep wave ( $ka = 0.4$ ), close to the wave-breaking limit. Exponential convergence rate is observed with respect to the two parameters  $N$  and  $M$ , thanks to the pseudo-spectral formalism of the model. Furthermore, note that thanks to the careful dealiasing procedure used in HOS-ocean, saturation observed in [8, 26] is removed and better accuracy is achieved.

Finally, benchmarks allow to test the configuration with waves defined along x-axis, y-axis or 3D wave propagating at  $\pi/4$  angle. Exact same results are obtained in the three set-ups.

### *3.2. Application: 3D irregular wave field*

This section presents a typical application case of the simulation of non-linear 3D irregular wave field. Physical and numerical parameters of the simulation are the following

- JONSWAP spectrum,  $H_s = 11$  m,  $T_p = 12.5$  s (mean steepness  $s = k_p \bar{a} = \frac{k_p H_s}{2\sqrt{2}} = 0.1$ ),  $\gamma = 5$
- Directional spreading defined as  $\beta = 0.14$
- Infinite water depth
- Lengths  $L_x \times L_y = 20\lambda_p \times 40\lambda_p \simeq 50$  km<sup>2</sup>
- Duration of simulation  $T_{stop} = 250 T_p \simeq 50$  min
- Discretization  $N_x \times N_y = 256 \times 256$
- HOS order  $M = 3$

- Time tolerance  $10^{-7}$
- Relaxation used with  $T_a = 10 T_p$  and  $n = 4$

An example of free surface elevation at a given time-step is presented in Fig. 3. The 3D wave-field simulated has a small angular spreading ( $\beta = 0.14 \text{ rad}$ ), as can be seen with the free surface elevation. At the same time, frequency spreading is also limited ( $\gamma = 5$ ). This is a typical configuration where the Benjamin-Feir instability is important during the propagation of such sea state.

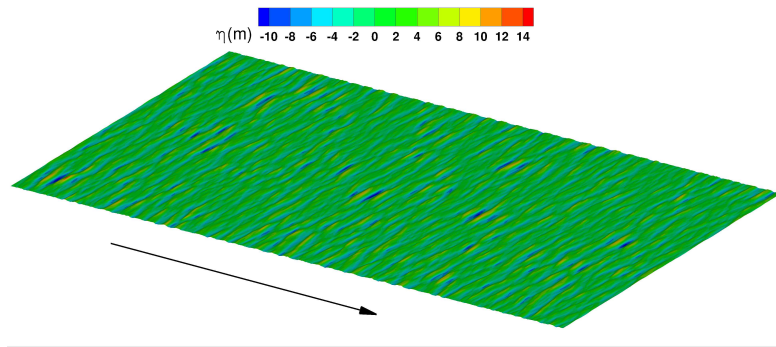


Figure 3: Initial free-surface elevation

Figure 4 presents the initial wave spectrum as well as the final spectrum after a propagation of  $250 T_p$ . Dashed lines indicates the most unstable Benjamin-Feir directions, along which spectrum evolution is expected to be preponderant [14].

This is a result with one realization (*i.e.* one random phase generation). Thus, this is maybe not very clear that we recover correctly those instabilities. However, it has been shown in [9] that HOS-ocean solves correctly

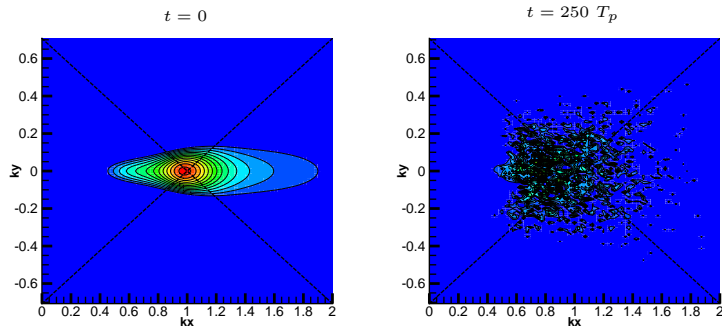


Figure 4: Evolution of wave spectrum during nonlinear propagation

this phenomenon. Consequently, for detailed analysis of physical processes, several realizations have to be made.

Note that post processing enables the analysis of the free surface elevation to extract evolution of interesting properties that characterize the nonlinear propagation of large-scale sea-states. For instance, Fig. 5 presents the evolution of kurtosis of free-surface elevation, as well as the normalized maximum wave height  $\frac{H_{max}}{H_s}$  observed during the simulation (defined as the maximum of each 3D wave detected with wave-by-wave analysis along mean and transverse direction). Note that this increase of kurtosis (*i.e.* the increase of the probability of freak waves' occurrence) is a typical nonlinear effect in such configuration due to third-order quasi-resonant interactions, see *e.g.* [15].

### 3.3. Kinematics and pressure inside the domain

The study of the kinematics induced by a wave field is of particular importance for in-depth analysis of the wave phenomena as well as for the wave-structure interactions. Here is given an example of the kinematics and

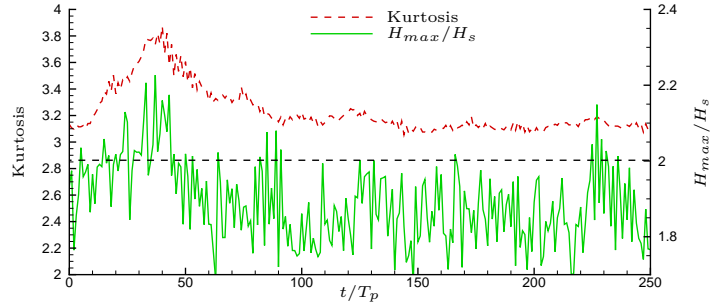


Figure 5: Evolution of kurtosis and normalised maximum wave height  $\frac{H_{max}}{H_s}$  during simulation

pressure induced by an extreme event appearing in a 2D irregular wave field. This sea-state is characterized by a JONSWAP spectrum with  $H_s = 4$  m,  $T_p = 10$  s and  $\gamma = 3.3$ . Water depth is equal to  $h = 200$  m. Similarly to the previous application of 3D irregular sea-states, freak waves may be monitored with a wave-by-wave analysis. Then, it is possible to identify such an extreme event and to study the induced flow in the whole fluid domain. In this example, the rogue wave exhibits a wave height  $H = 2.57 H_s = 10.3$  m. Figure 6 presents the dynamic pressure field and the horizontal velocity field at the moment of the extreme event.

## Conclusions and future work

HOS-ocean code has been developed to study the propagation of large-scale nonlinear sea states. HOS-ocean is an open-source code, redistributed under the terms of the GNU General Public License (version 3) as published by the Free Software Foundation. Along with the source code, a documentation under wiki format is available which makes compilation and execution of

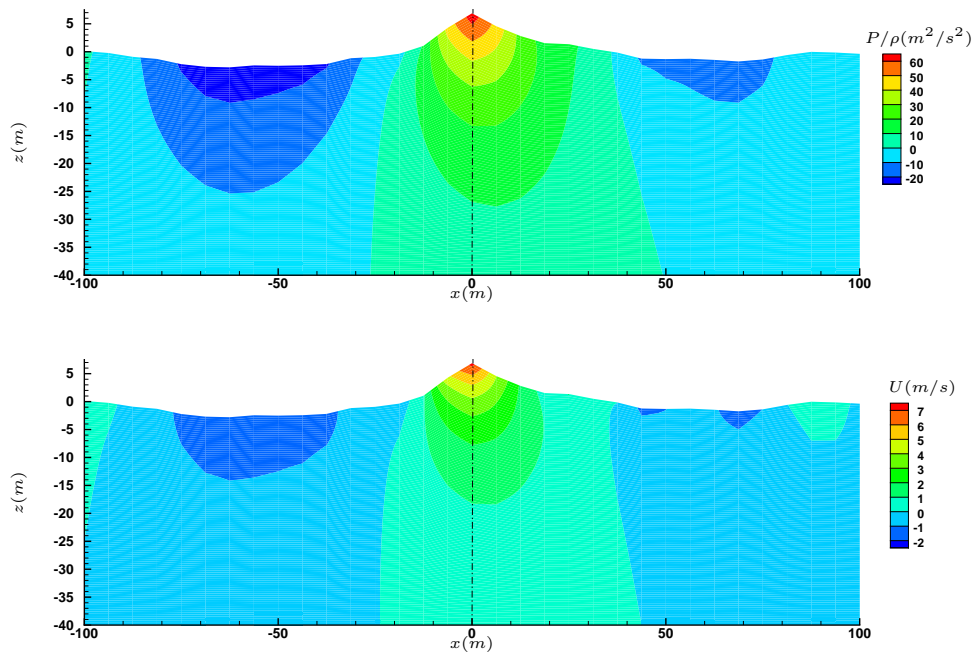


Figure 6: Dynamic pressure (top) and horizontal velocity (bottom) of an extreme wave appearing in open ocean.



the source files easy. In addition, benchmarking procedures as well as typical post-processing examples are provided with the main code.

The code has been shown to be accurate and highly efficient thanks to the interesting convergence properties of pseudo-spectral methods. It has been highly validated on several configurations and one of the purpose of the open-source release is to encourage researchers to use it. Particularly, thanks to the post-processing provided, the coupling between HOS-ocean and CFD software for wave-structure interactions modeling should be straightforward.

HOS-ocean, based on a pseudo-spectral formalism, uses FFTs which are performed with the FFTW library [17]. Parallelization of the program using MPI is consequently quite simple since FFTW provides also parallel FFT implementations. Preliminary works have already been done in [11] and show promising results. This will be included in the open-source version in the next future.

Finally, note that a HOS model dedicated to the simulation of a Numerical Wave Tank has been developed at ECN. It intends to reproduce a wave basin and its specificities: lateral reflective walls, wavemaker and absorbing beach. The resulting model named HOS-NWT has been presented in [13] and is also available as open-source release [1].

## References

- [1] Open-source release of HOS-NWT. <https://github.com/LHEEA/HOS-NWT/wiki>.
- [2] Open-source release of HOS-ocean. <https://github.com/LHEEA/HOS-ocean/wiki>.

- [3] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.
- [4] W.J.D. Bateman, C. Swan, and P.H. Taylor. On the calculation of the water particle kinematics arising in a directionnally-spread wavefield. *J. Comp. Phys.*, 186:70–92, 2003.
- [5] F. Bonnefoy, G. Ducrozet, D. Le Touzé, and P. Ferrant. *Advances in Numerical Simulation of Nonlinear Water Waves*, volume 11 of *Advances in Coastal and Ocean Engineering*, chapter Time domain simulation of non linear water waves using spectral methods. World Scientific, 2010.
- [6] J.R. Cash and A.H. Karp. A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides. *ACM Trans. Math. Softw.*, 16(3):201–222, 1990.
- [7] D.G. Dommermuth. The initialization of nonlinear waves using an adjustment scheme. *Wave Motion*, 32:307–317, 2000.
- [8] D.G. Dommermuth and D.K.P. Yue. A high-order spectral method for the study of nonlinear gravity waves. *J. Fluid Mech.*, 184:267 – 288, 1987.
- [9] G. Ducrozet. *Modélisation des processus non-linéaires de génération et de propagation d'états de mer par une approche spectrale*. PhD thesis, École Centrale de Nantes.

- [10] G. Ducrozet, H. B. Bingham, A. P. Engsig-Karup, F. Bonnefoy, and P. Ferrant. A comparative study of two fast nonlinear free-surface water wave models. *Int. J. Numer. Meth. Fl.*, 69(11):1818–1834, 2012. doi: 10.1002/flid.2672.
- [11] G. Ducrozet, F. Bonnefoy, and P. Ferrant. Analysis of freak waves formation with large scale fully nonlinear high order spectral simulations. In *Proc. 18<sup>th</sup> Int. Symp. on Offshore and Polar Engng.*, Vancouver, Canada, July 2008.
- [12] G. Ducrozet, F. Bonnefoy, D. Le Touzé, and P. Ferrant. 3-D HOS simulations of extreme waves in open seas. *Nat. Hazards Earth Syst. Sci.*, 7(1):109–122, 2007.
- [13] G. Ducrozet, F. Bonnefoy, D. Le Touzé, and P. Ferrant. A modified high-order spectral method for wavemaker modeling in a numerical wave tank. *Eur. J. Mech. - B-Fluid*, 34:19 – 34, 2012.
- [14] K.B. Dysthe, K. Trulsen, H.E. Krogstad, and H. Socquet-Juglard. Evolution of a narrow-band spectrum of random surface gravity waves. *J. Fluid Mech.*, 478:1–10, 2003.
- [15] F. Fedele. On the kurtosis of deep-water gravity waves. *J. Fluid Mech.*, 782:25–36, 2015.
- [16] L. Fernandez, M. Onorato, J. Monbaliu, and A. Toffoli. Modulational instability and wave amplification in finite water depth. *Nat. Hazards Earth Syst. Sci.*, 14:705–711, 2014. doi:10.5194/nhess-14-705-2014.

- [17] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on “Program Generation, Optimization, and Platform Adaptation”.
- [18] D. Fructus, D. Clamond, J. Grue, and Ø. Kristiansen. An efficient model for three-dimensional surface wave simulations. Part I: Free space problems. *J. Comp. Phys.*, 205:665–685, 2005.
- [19] C. Kharif, J.-P. Giovanangeli, J. Touboul, L. Grare, and E. Pelinovski. Influence of wind on extreme wave events: experimental and numerical approaches. *J. Fluid Mech.*, 594:209–247, 2008.
- [20] R. Luquet, G. Ducrozet, L. Gentaz, P. Ferrant, and B. Alessandrini. Applications of the SWENSE method to seakeeping simulations in irregular waves. In *Proc of the 9<sup>th</sup> Int. Conf. on Num. Ship Hydro.*, Ann Arbor, Michigan, 2007.
- [21] C. Monroy, G. Ducrozet, F. Bonnefoy, A. Babarit, L. Gentaz, and P. Ferrant. Rans simulations of a calm buoy in regular and irregular seas using the SWENSE method. *Int. J. Offshore Polar*, 21(4):264–271, 2011.
- [22] G. Oger, D. Le Touzé, P.M. Guilcher, and M. de Lefte. Advances in SPH for naval hydrodynamics. In *Proc. of the 30<sup>th</sup> Symposium on Naval Hydrodynamics*, Hobart, Tasmania, Australia, 2014.
- [23] M. M. Rienecker and J. D. Fenton. A Fourier approximation method for steady water waves. *J. Fluid Mech.*, 104:119–137, 1981.
- [24] A. Sergeeva and A. Slunyaev. Rogue waves, rogue events and extreme

- wave kinematics in spatio-temporal fields of simulated sea states. *Nat. Hazards Earth Syst. Sci.*, 13:1759–1771, 2013.
- [25] V.I. Shrira and Slunyaev A.V. Nonlinear dynamics of trapped waves on jet currents and rogue waves. *Phys. Rev. E*, 89(041002), 2014.
- [26] C. Skandrani, C. Kharif, and J. Poitevin. Nonlinear evolution of water surface waves: the frequency down-shift phenomenon. *Contemporary Mathematics*, 200:157–171, 1996.
- [27] M. Tanaka. Verification of Hasselmann’s energy transfer among surface gravity waves by direct numerical simulations of primitive equations. *J. Fluid Mech.*, 444:199–221, 2001.
- [28] A. Toffoli, O. Gramstad, K. Trulsen, J. Monbaliu, E. M. Bitner-Gregersen, and M. Onorato. Evolution of weakly nonlinear random directional waves: laboratory experiments and numerical simulations. *J. Fluid Mech.*, 664:313–336, 2010.
- [29] A. Toffoli, M. Onorato, E. M. Bitner-Gregersen, and J. Monbaliu. Development of a bimodal structure in ocean wave spectra. *J. Geophys. Res.*, 115(C03006), 2010. doi:10.1029/2009JC005495.
- [30] J. Touboul. On the influence of wind on extreme wave events. *Nat. Hazards Earth Syst. Sci.*, 7:123–128, 2007.
- [31] B. J. West, K. A. Brueckner, R. S. Janda, M. Milder, and R. L. Milton. A new numerical method for surface hydrodynamics. *J. Geophys. Res.*, 92:11803 – 11824, 1987.

- [32] W. Xiao, Y. Liu, G. Wu, and D.K.P. Yue. Rogue wave occurrence and dynamics by direct simulations of nonlinear wave-field evolution. *J. Fluid Mech.*, 720:357–392, 2013.
- [33] V. Zakharov. Stability of periodic waves of finite amplitude on the surface of a deep fluid. *J. Appl. Mech. Tech. Phys.*, pages 190–194, 1968.