BEMRosetta: An open-source hydrodynamic coefficients converter and viewer integrated with Nemoh and FOAMM

Iñaki Zabala, Yerai Peña-Sanchez, Thomas Kelly, João C. C. Henriques,
Markel Penalba, Nicolás Faedo, John V. Ringwood and Jesús M. Blanco

***Abstract*—Boundary Element Method (BEM) solvers are extensively used to obtain the hydrodynamic coefficients required to model hydrodynamic forces in floating marine structures. BEM solvers require the discretization of the submerged device surface as a mesh to compute the hydrodynamic coefficients as radiation damping and added mass, response amplitude operators and linear and second-order exciting forces. Each of these solvers need particular input files and mesh formats, and save the results in specific file formats. Typically, the input and output files are incompatible between different solvers. Researchers handle this problem by converting model results through homemade spreadsheets or macros made in scripting languages. BEMRosetta was created to allow loading and saving the input files, mesh geometries and the hydrodynamic coefficients, in different formats. Furthermore, it also includes a mesh viewer. Additionally, BEMRosetta can calculate different parameters from the mesh and the hydrodynamic coefficients. Through its integration with the Finite-Order hydrodynamic Approximation by Moment-Matching (FOAMM) toolbox, BEMRossetta allows the state-space model of the radiation convolution term for the desired degrees of freedom be obtained.**

***Index Terms*—BEM, Nemoh, FOAMM, Capytaine, HAMS.**

I. Introduction

FLOATING marine devices such as ships, offshore wind platforms or wave energy converters are usually modelled using hydrodynamic coefficients obtained from potential flow BEM solvers. Based on these frequency domain coefficients comprising added mass, radiation damping and excitation forces, frequency domain, or normally time domain simulations can be created to estimate the dynamic loads and power output for the marine energy converters in question. All the solvers are made by different companies and open source projects and use different formats for a) the geometry mesh definition, b) the BEM case definition including, for example, the set of frequencies, wave headings, water depth, degrees of freedom to do the calculation, and c) to set the hydrodynamic coefficients (Fig. 1).

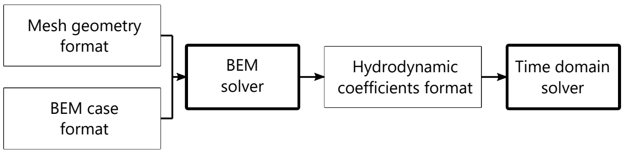


Fig. 1: Basic outline of the BEM related section of the floating body simulation workflow.

These different file formats make it difficult to share cases and results between research groups, and limit the use of some formats that are less supported than other formats.

Expert programmers solve this problem by implementing filters and conversion routines with reduced error checking and sometimes low re-usability. Because of this, researchers with limited programming skills or economic resources to access software licenses, may not be able to convert the results obtained from open-source codes, like Nemoh [1], Capytaine [2] or HAMS [3], into the specific formats required by other software.

BEMRosetta is a software tool designed to solve these drawbacks, allowing easy sharing of files between researchers and using as much as possible open source tools for the workflow of developers without economic resources. It includes an intuitive, window-based user interface that allows the use of its features in an agile way and without programming knowledge. Using the built-in graphics viewer, BEMRosetta allows for the visual comparison between the results obtained with different software packages, as well as the results obtained with the same software but with different discretization levels. Additionally, BEMRosetta can calculate the infinite frequency added mass, the impulse response function of the radiation force, and a wide set of geometric parameters like the hydrodynamic stiffness matrix, submerged volume and centre of buoyancy. BEMRosetta also includes a wizard for handling Nemoh, which eases the use of such BEM solver, especially for beginners.

This paper, with ID 70/2124/TRT/1392, has been submitted in the EWTEC 2021 conference track *Wave hydrodynamic modelling.*

I. Zabala (e-mail: inaki.zabala@sener.es) is at SENER Ingeniería y Sistemas, S.A., Av. Zugazarte 56, 48930 Getxo, Spain

Y. Peña-Sanchez (e-mail: yerai.pena.2017@mumail.ie), and J. V. Ringwood (e-mail: John.Ringwood@mu.ie) are at the Centre for Ocean Energy Research (COER), Maynooth University, Co. Kildare, Ireland

M. Peñalba-Retes (e-mail: mpenalba@mondragon.edu) was at the Centre for Ocean Energy Research (COER), Maynooth University, Co. Kildare, Ireland. He is now with the Mondragon University, Loramendi Kalea, 4. 20500 Arrasate, Spain

T. Kelly (e-mail: thomas.kelly@dkit.ie) was at the Centre for Ocean Energy Research (COER), Maynooth University, Co. Kildare, Ireland. He is now with the Dunkalk Institute of Technology, Marshes Upper, Dundalk, Co. Louth, A91 K584, Ireland

N. Faedo (e-mail: nicolas.faedo.2017@mumail.ie) was at the Centre for Ocean Energy Research (COER), Maynooth University, Co. Kildare, Ireland. He is now with the Marine Offshore Renewable Energy Lab, Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Italy

J.C.C. Henriques (e-mail: joaochenriques@tecnico.ulisboa.pt) is at IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, 1049-001 Lisboa, Portugal

Jesús M. Blanco (e-mail: jesusmaria.blanco@ehu.eus) is at the University of the Basque Country (UPV/EHU), Plaza Ingeniero Torres Quevedo 1, 48130 Bilbao, Spain

There is special support to a branch of Nemoh v. 115 that apart from handling meshes of over 12,000 panels, has stronger irregular frequencies functionalities than previous versions [4]. Apart from this, the graphical user interface allows a seamless interaction with the FOAMM toolbox [5] without the need to have any version of Matlab® installed except the one included freely in the FOAMM installer.

BEMRosetta runs on both Windows and Linux operating systems, it is open-source released in GitHub[1], its portable (no installation is required for the Windows version), and it includes a graphical user interface, a command-line version a dynamic library and a Python interface. File formats supported by BEMRosetta include those used by Nemoh, Capytaine, HAMS, Wamit® and Ansys AQWA®, which can be used as input for Orcaflex®, WEC-Sim[2], OpenFAST[3] or DNV Bladed®. Because of this, the conversion routines are able to handle some poorly documented file formats from proprietary solvers.

The following is a summary of BEMRosetta's main contributions:

- Load and save different BEM mesh and hydrodynamic coefficients formats
- Simple, compactly arranged tools which are used like a Swiss army knife: All files, mesh, Nemoh or BEM files, can be loaded by drag and drop or copy and paste from file explorer in Windows and Linux.
- Foster interoperability between different research teams: Meshes, cases definition and hydrodynamic coefficients can be shared.
- Makes these tools available to researchers without funds or programming knowledge.
- Multi-platform: Versions in Windows and in Linux, GUI or command line.

Examples of its use are given below:

- Load the hydrodynamic coefficients for the same geometry from different software and compare the results
- Load the hydrodynamic coefficients for the same geometry with different levels of discretisation to check for convergence.
- Load the hydrodynamic coefficients for different geometries to compare them
- Mesh loading, combining them for visual comparison

BEMRosetta is programmed in the C++ programming language for fast processing, using the U++[4] libraries to ease the development, and Anboto[5] and Eigen[6] libraries for its advanced numerical tools. In addition, some calculations have been carried out using GNU Octave[7] and Maxima[8].

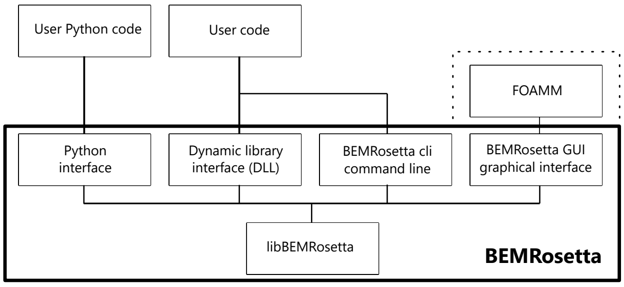

User Python code
User code
FOAMM
Python interface
Dynamic library interface (DLL)
BEMRosetta cli command line
BEMRosetta GUI graphical interface
libBEMRosetta
BEMRosetta

Fig. 2: BEMRosetta internal architecture. The solid line indicates the packages included in BEMRosetta. The dashed line indicates the possibility for the user to install the FOAMM software to be used transparently by BEMRosetta.

The examples included in this paper are based on the semisubmersible platform designed by the DeepCwind Consortium, led by the University of Maine [6].

This paper is organised in six parts. The first section gives an outline of the architecture. The second part describes the mesh processing. In the third part, the definition of cases to be processed with BEM solvers is explained. The fourth part defines the processing of hydrodynamic coefficients. In the fifth part it is described the integration with FOAMM. The sixth part includes other features not directly related with BEM solving but that can be of value for the researchers, possible areas for future development of the tool, and some comments indicating problems in the development of such a tool and asking for community support in different ways.

II. Architecture

BEMRosetta structure, detailed in Fig. 2, is based in a back-end called libBEMRosetta and different front-ends:

- A graphical user interface, included in bemrosetta binary. It includes all available features in a user friendly desktop application. These features include the possibility to use the FOAMM software, described in section VI, in a way that is transparent to the user.
- Command line interface (CLI), included in bemrosetta_cl binary. It makes some of the BEMRosetta functions available to users, to be used from command line or from simple batch files. The number of functions available has been expanded in response to user requests.
- Dynamic library (DLL): This binary allows any software to directly access the functions available in the CLI interface without any performance penalty.
- Python interface: It makes the functions provided in the CLI interface available to Python programmers in a simple and transparent way. It includes the necessary Python glue code, and it relies in ctypes[9] Python built-in library.

III. Mesh processing

BEMRosetta includes a basic wireframe mesh viewer that allows to load and convert different formats related to BEM solvers. The formats supported are:

[1] https://github.com/BEMRosetta/BEMRosetta
[2] https://github.com/WEC-Sim/WEC-Sim
[3] https://github.com/OpenFAST/openfast
[4] https://www.ultimatepp.org
[5] https://github.com/Anboto/anboto
[6] https://eigen.tuxfamily.org
[7] https://www.gnu.org/software/octave
[8] https://maxima.sourceforge.io/
[9] https://docs.python.org/3/library/ctypes.html

- Load and view
  - Wamit: .gdf, pan.dat
  - Nemoh: .dat
  - Ansys AQWA: .dat
  - HAMS: .pnl
  - STL format: .stl (binary and text)
- Save
  - Wamit: .gdf
  - HAMS: .pnl (hull and waterplane)
  - STL format: .stl (binary and text)

The mesh visualisation options allow the display of panel normals in the form of arrows, body boundaries, centre of gravity, centre of buoyancy, the underwater part of the mesh, panels detected as erroneous in the mesh healing process, and the list of panels and nodes.

Meshes are loaded into an internal generic mesh scheme that allows to be saved to any mesh format.

Before saving a mesh file, BEMRosetta analyses its structure looking for symmetry in x and y axis. Furthermore, the user is able to save a file forcing symmetries. For this purpose, BEMRosetta examines the mesh and makes the necessary cuts in the panels. This reduces the file size and may optimise solver simulation speed as it may take advantage of symmetries.

In addition to a number of options for making, for example, simple axisymmetric or polygonal meshes, it is possible to perform different operations on the meshes:

- General: Translation, rotation, mirror image, mesh simplify, mesh healing, panel normals reorienting.
- BEM specific: Obtain underwater mesh, obtain waterplane mesh, obtain the mesh hull without waterplane and generate waterplane mesh based on how the hull crosses the waterplane.

The mesh management includes a table for parallel comparison of the parameters of different geometries loaded at the same time, like the volume (Eq. (1)) or the centre of buoyancy (Eq. (2)). In addition it includes the calculation and comparison of the different hydrostatic stiffness matrices (Eq. (3)) [7].

$$V = -\iint_{S_b} n_x dS = -\iint_{S_b} n_y dS = -\iint_{S_b} n_z dS \tag{1}$$

$$\boldsymbol{X_b} = \begin{pmatrix} -\frac{1}{2V}\iint_{S_b} n_x x^2 dS \\ -\frac{1}{2V}\iint_{S_b} n_y y^2 dS \\ -\frac{1}{2V}\iint_{S_b} n_z z^2 dS \end{pmatrix} \tag{2}$$

$$\boldsymbol{C} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \rho g \iint_{S_b} n_z dS & \rho g \iint_{S_b} y n_z dS & -\rho g \iint_{S_b}^{x} n_z dS & 0 \\ 0 & 0 & \rho g \iint_{S_b} y n_z dS & \begin{matrix}\rho g \iint_{S_b} y^2 n_z dS + \\ \rho g V z_b - m g z_g\end{matrix} & \rho g \iint_{S_b} x y n_z dS & \rho g V x_b + m g x_g \\ 0 & 0 & -\rho g \iint_{S_b}^{x} n_z dS & 0 & \begin{matrix}\rho g \iint_{S_b} x^2 n_z dS + \\ \rho g V z_b - m g z_g\end{matrix} & -\rho g V y_b + m g y_g \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3}$$

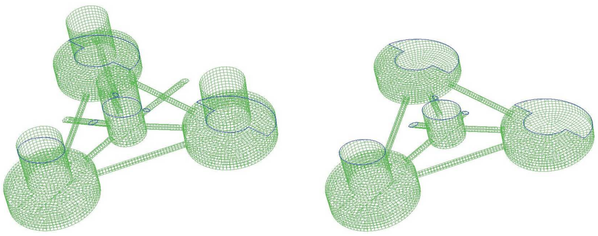
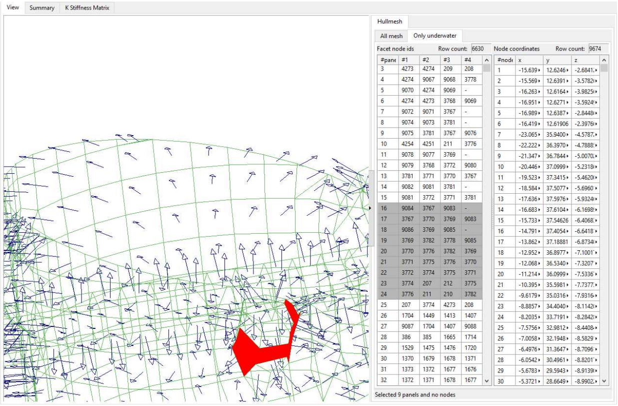


Fig. 3: Hull mesh with waterline (left) and after cutting at the waterline (right).



Facet node ids Row count: 6630

#pane	#1	#2	#3	#4
3	4273	4274	209	208
4	4274	9067	9068	3778
5	9070	4274	9069	-
6	4274	4273	3768	9069
7	9072	9071	3767	-
8	9074	9073	3781	-
9	9075	3781	3767	9076
10	4254	4251	211	3776
11	9078	9077	3769	-
12	9079	3768	3772	9080
13	3781	3771	3770	3767
14	9082	9081	3781	-
15	9081	3772	3771	3781
16	9084	3767	9083	-
17	3767	3770	3769	9083
18	9086	3769	9085	-
19	3769	3782	3778	9085
20	3770	3776	3782	3769
21	3771	3775	3776	3770
22	3772	3774	3775	3771
23	3774	207	212	3775
24	3776	211	210	3782
25	207	3774	4273	208
26	1704	1449	1413	1407
27	9087	1704	1407	9088
28	386	385	1665	1714
29	1529	1475	1476	1720
30	1370	1679	1678	1371
31	1373	1372	1677	1676
32	1372	1371	1678	1677

Node coordinates Row count: 9674

#node	x	y	z
1	-15.639▸	12.6246▸	-2.6841▸
2	-15.569▸	12.6391▸	-3.5782▸
3	-16.263▸	12.6164▸	-3.9825▸
4	-16.951▸	12.6271▸	-3.5924▸
5	-16.989▸	12.6387▸	-2.8448▸
6	-16.419▸	12.61906	-2.3976▸
7	-23.065▸	35.9400▸	-4.5787▸
8	-22.222▸	36.3970▸	-4.7888▸
9	-21.347▸	36.7844▸	-5.0070▸
10	-20.446▸	37.0999▸	-5.2318▸
11	-19.523▸	37.3415▸	-5.4620▸
12	-18.584▸	37.5077▸	-5.6960▸
13	-17.636▸	37.5976▸	-5.9324▸
14	-16.683▸	37.6104▸	-6.1698▸
15	-15.733▸	37.54626	-6.4068▸
16	-14.791▸	37.4054▸	-6.6418▸
17	-13.862▸	37.18881	-6.8734▸
18	-12.952▸	36.8977▸	-7.1001▸
19	-12.068▸	36.5340▸	-7.3207▸
20	-11.214▸	36.0999▸	-7.5336▸
21	-10.395▸	35.5981▸	-7.7377▸
22	-9.6179▸	35.0316▸	-7.9316▸
23	-8.8857▸	34.4040▸	-8.1142▸
24	-8.2035▸	33.7191▸	-8.2842▸
25	-7.5756▸	32.9812▸	-8.4408▸
26	-7.0058▸	32.1948▸	-8.5829▸
27	-6.4976▸	31.3647▸	-8.7096▸
28	-6.0542▸	30.4961▸	-8.8201▸
29	-5.6783▸	29.5943▸	-8.9139▸
30	-5.3721▸	28.6649▸	-8.9902▸

Selected 9 panels and no nodes

Fig. 4: Detail of the interface with panels selected and normals viewed.

where ρ is the water density, g is the acceleration of gravity, V is the volume, S_b is the mean body wetted surface, x_b, y_b and z_b are the components of the centre of buoyancy, x_g, y_g and z_g are the components of the centre of gravity, and for each panel of the mesh, x, y and z are the coordinates, n_x, n_y and n_z are the components of the normal, and S is the surface.

IV. Definition of cases to be processed with BEM solvers

Each BEM solver defines the case parameters to be calculated with one or more input files and the body meshes to be used as the basis for the calculation (Fig. 1). These input files include similar information in the different solvers, like the degrees of freedom of each body involved, the frequencies to calculate and the headings of the wave exciting forces. In addition, they include the ability to include information to perform additional calculations, such as mass, centre of gravity and inertia matrix needed to obtain the RAO, or duration and sampling time for the calculation of the impulse response function.

BEMRosetta allows the loading and saving of calculation cases in different formats like Nemoh-Capytaine (.cal files) and HAMS (.in files), including the necessary folder structure. In addition it allows for the creation of new cases by editing the parameters, setting up the necessary binaries and input files in a folder structure with batch files ready to launch the hydrodynamic coefficients calculation.

In case that either the solver is not able to parallelise the work on the different cores of the microprocessor, if this parallelisation is inefficient in certain circumstances, or if it is desired to spread the calculation over different computers, it is possible to split the calculation into multiple sub cases, by automatically dividing the total frequency range into different sub-ranges. In this way, each sub case can be executed in different command windows, thus using the different cores and can be solved more quickly. Once the work has been done, BEMRosetta allows all sub cases to be joined together into one single simulation as if it had been calculated in only one run.

V. Hydrodynamic coefficients processing

BEMRosetta facilitates the reading, showing, comparing, joining, processing and converting of various BEM formats, such as:

- Load-View-Process
  - Wamit: .out, .3sc, 3fk, .1, .3, .4, .hst, .12s, .12d
  - HAMS: .1, .3, .4, .hst
  - Nemoh/Capytaine: Nemoh.cal, Mesh/Hydrostatics*.dat, Mesh/KH*.dat, RadiationCoefficients.tec, ExcitationForce.tec, DiffractionForce.tec, FKForce.tec, IRF.tec
  - FAST-Wamit: HydroDyn.dat, .1, .3, .hst, .ss, .12s, .12d
  - SeaFEM-Nemoh: .flavia.inf, RadiationCoefficients.tec, ExcitationForce.tec, DiffractionForce.tec, FKForce.tec
  - Ansys AQWA: .lis, .ah1, .qtf
  - FOAMM: .mat
- Save
  - Wamit: .out, .1, .3, .hst, .4, .12s, .12d
  - FAST-Wamit: HydroDyn.dat, .1, .3, .hst, .ss, .12s, .12d

Other solvers that may use these files are Bladed or Orcaflex.

This allows the handling of frequency and time domain information like added mass, added mass in infinite frequency, radiation damping, impulse response function (IRF), response amplitude operator (RAO), first order and second order wave exciting forces and state-space matrices.

Simulations including different frequencies and headings processed separately can be joined and saved in one only case. Simulations with forces calculated for a set of headings in one direction, can be mirrored.

A. The management of the dimensions

Hydrodynamic coefficients can be handled in dimensional or non-dimensional form. On this issue there is no agreement among solvers, using either one or the other form. BEMRosetta stores the coefficients in the native format, including if available the density and gravity with which they have been calculated. When displaying this information, the user can choose to display them in one way or another. Since one of the uses is for comparison between solvers, care must be taken that the dimensional data are displayed with respect to the same density. It is common for some research teams to work with fresh water density, while others work with salt water density. Therefore, since two solvers have been able to perform their calculations on the same geometry, but with different densities, the data are shown in BEMRosetta considering the same reference density, which is stored in the Options menu and which, due to its critical nature, is always available together with the gravity in the upper right part of the program window. The conversion equations used are shown below. Adimensional parameters have a bar symbol on them, like in C and $\bar{C}$: [7]:

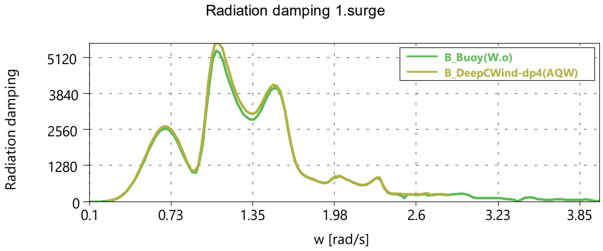


Fig. 5: Comparison of the radiation damping in surge of the same mesh modelled with two different solvers (Wamit and AQWA).

- Hydrostatic stiffness matrix

$$\boldsymbol{C} = \bar{\boldsymbol{C}} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \rho g L^2 & \rho g L^3 & \rho g L^3 & 0 \\ 0 & 0 & \rho g L^3 & \rho g L^4 & \rho g L^4 & \rho g L^4 \\ 0 & 0 & \rho g L^3 & \rho g L^4 & \rho g L^4 & \rho g L^4 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{4}$$

- Hydrodynamic coefficients

$$\begin{aligned} A_{ij} &= \bar{A}_{ij} \rho L^m \\ B_{ij} &= \bar{B}_{ij} \rho L^m \omega \end{aligned} \tag{5}$$

where L is the reference length (usually set to 1), and exponent m is got for each degree of freedom from next matrix:

$$\begin{bmatrix} 3 & 3 & 3 & 4 & 4 & 4 \\ 3 & 3 & 3 & 4 & 4 & 4 \\ 3 & 3 & 3 & 4 & 4 & 4 \\ 4 & 4 & 4 & 5 & 5 & 5 \\ 4 & 4 & 4 & 5 & 5 & 5 \\ 4 & 4 & 4 & 5 & 5 & 5 \end{bmatrix}$$

- Wave exciting forces F and RAO (only represented F in equation)

$$F = \bar{F} \rho g L^m \tag{6}$$

where $m = 2$ for $i = 1, 2, 3$ and $m = 3$ for $i = 4, 5, 6$.

B. Density correction

It is possible that the designs to be compared have been calculated non-dimensional, or with different water densities. To resolve this, the density considered for dimensional conversions is always the one indicated in BEMRosetta Options and shown, together with the density, in the top right corner of the window interface. In addition, Equations (7), (8) are used to set default density:

- For added mass $A(\omega)$ and radiation damping $B(\omega)$ (only represented A in equation):

$$A_{\text{shown}} = A_{\text{loaded}} \frac{\rho_{opt}}{\rho_{load}} \tag{7}$$

- For the impulse response function IRF k_{r}, hydrostatic stiffness matrix C, excitation forces F and response amplitude operator RAO (only represented k_{r} in equation):

$$k_{\text{r shown}} = k_{\text{r loaded}} \frac{\rho_{opt} g_{opt}}{\rho_{load} g_{load}} \tag{8}$$

C. Added mass at infinite frequency and IRF calculation

In the event that the solver has not obtained $k_{\text{r}}(t)$ and μ_∞, it is possible to calculate them following Ogilvie's formulas [8], which are defined as follows:

$$B(\omega) = \int_0^\infty k_{\text{r}}(t)\cos(\omega t)dt \tag{9}$$

$$A(\omega) = \mu_\infty - \frac{1}{\omega}\int_0^\infty k_{\text{r}}(t)\sin(\omega t)dt \tag{10}$$

Therefore, from Eq. (9) the radiation IRF $k_{\text{r}}(t)$ can be obtained as a function of the frequency domain hydrodynamic parameter $B(\omega)$ as

$$k_{\text{r}}(t) = \frac{2}{\pi}\int_0^\infty B(\omega)\cos(\omega t)\, d\omega \tag{11}$$

And since we know $k_{\text{r}}(t)$ and $A(\omega)$, with equation Eq. (10) it is possible to obtain μ_∞.

It should be noted that μ_∞ obtained from Eq. (10) would have to give the same value regardless of ω. However to reduce numerical inaccuracies the average for all available ω values is obtained [9].

VI. Integration with FOAMM

In the literature, the motion of floating devices is often described using the well-known Cummins' equation [10], where the radiation force is characterised by a convolution operation. Such a convolution term often represents a drawback for several applications such as, for example, optimisation studies (due to the computational effort associated with the solution of the convolution) or optimal control (since they usually require a state-space model describing the system). To overcome this disadvantage, and given that the convolution operator intrinsically defines a dynamical system, the convolution term is commonly approximated using finite-order parametric models. To this end, the Centre for Ocean Energy Research (COER) developed a moment-matching based identification method for the radiation force subsystem termed Finite Order Approximation by Moment-Matching (FOAMM). Even though FOAMM was originally distributed as an application for Matlab, BEMRosetta provides an interactive user interface to use FOAMM with no need for any other application.

A. FOAMM mathematical description

This section provides a brief summary of the theory behind the radiation convolution term approximation mode of FOAMM. The interested reader is referred to [5] for an extensive discussion on the underlying mathematical principles, or to [11] for the definition of all the identification modes and options comprised of FOAMM. Since the radiation convolution term represents a Linear Time Invariant (LTI) system by itself, it can be represented using a state space (SS) model, with the velocity of the device $v(t)$ as input, as

$$\begin{aligned} \dot{x}_{\text{ss}}(t) &= A_{\text{ss}}x_{\text{ss}}(t) + B_{\text{ss}}v(t), \\ y_{\text{ss}}(t) &= C_{\text{ss}}x_{\text{ss}}(t), \end{aligned} \tag{12}$$

where $y_{\text{ss}}(t) \approx f_{\text{r}}^*(t) = \int_0^t k_{\text{r}} v(t)dt$ is the output, $x_{\text{ss}}(t)$ the state vector, A_{ss} the system matrix, B_{ss} the input matrix, and C_{ss} the output matrix of the SS model. First, the input $v(t)$ is expressed as a signal generator, written in implicit form as

$$\begin{aligned} \dot{x}_v(t) &= A_{\text{sg}}x_v(t), \\ v(t) &= C_{\text{sg}}x_v(t). \end{aligned} \tag{13}$$

where C_{sg} is chosen so that $(A_{\text{sg}}, C_{\text{sg}})$ is observable, and A_{sg} is a block diagonal matrix defined as

$$A_{\text{sg}} = \bigoplus_{i=1}^{\beta} \begin{bmatrix} 0 & \omega_i \\ -\omega_i & 0 \end{bmatrix}, \tag{14}$$

the symbol $\bigoplus_{i=1}^{\beta}$ stands for the direct sum of β matrices, and β is the number of interpolation frequencies. It should be noted that each ω_i represents a frequency where the transfer function of the parametric model exactly matches the transfer function of the target system.

Following the theory developed in [5], a family of models, achieving moment-matching at a set of frequencies $\mathcal{F} = \{\omega_1, .., \omega_\beta\}$ and parametrised on G, can be described as

$$\begin{aligned} \dot{x}_{\text{ss}}(t) &= (A_{\text{sg}} - GC_{\text{sg}})x_{\text{ss}}(t) + Gv(t), \\ y_{\text{ss}}(t) &= Y_k x_{\text{ss}}(t). \end{aligned} \tag{15}$$

In Eq. (15), $Y_k = C_{\text{sg}}R$ is the moment-domain equivalent of the output (the radiation convolution term), and R can be obtained from the frequency domain hydrodynamic coefficients as

$$R = \bigoplus_{i=1}^{\beta} \begin{bmatrix} \text{real}(K_{\text{r}}(\omega_i)) & \text{imag}(K_{\text{r}}(\omega_i)) \\ -\text{imag}(K_{\text{r}}(\omega_i)) & \text{real}(K_{\text{r}}(\omega_i)) \end{bmatrix}, \tag{16}$$

with $K_{\text{r}}(\omega_i)$ the frequency domain analogous of the radiation convolution term defined as

$$K_{\text{r}}(\omega) = B(\omega) + j\omega[A(\omega) - \mu_\infty], \tag{17}$$

at the interpolation frequencies ω_i.

Thus, the additional flexibility provided by G is exploited to assign the eigenvalues of system (15). Such eigenvalues are chosen within an optimisation formulation, to minimise the Euclidean distance between the radiation convolution term frequency response $K_\mathrm{r}(\omega)$, and the frequency response of the parametric family of Eq. (15) $\tilde{K}_\mathrm{r}(\omega)$, which for the set of frequencies $\mathcal{F}$, can be obtained from the transfer function

$$\tilde{K}_{\mathrm{r}_\mathcal{F}} = Y_k \left[s\mathrm{I}_{n_\mathrm{ss}} - (A_\mathrm{sg} - GC_\mathrm{sg}) \right]^{-1} G, \tag{18}$$

where $\mathrm{I}_{n_\mathrm{ss}}$ is an identity matrix of size $n_\mathrm{ss} = 2\beta$, which is the size of the final SS model. It should be noted that the device parameters $\mathrm{Ar}(\omega)$ and $\mathrm{Br}(\omega)$ are calculated at a finite number of user-defined frequencies using BEM solvers. However, the approximation of the parametric model can be computed on a reduced (also user-selected) frequency range, with frequency step of Δ_ω and lower and upper bound frequencies given by ω_l and ω_u, respectively. Thus, defining the complex-valued vectors K_fr and $\tilde{K}_\mathrm{fr}$ as

$$\begin{aligned} K_\mathrm{fr} &= \begin{bmatrix} K_\mathrm{r}(\omega_\mathrm{l}) & K_\mathrm{r}(\omega_\mathrm{l} + \Delta_\omega) & \cdots & K_\mathrm{r}(\omega_\mathrm{u}) \end{bmatrix}, \\ \tilde{K}_\mathrm{fr} &= \begin{bmatrix} \tilde{K}_{\mathrm{r}_\mathcal{F}}(\omega_\mathrm{l}) & \tilde{K}_{\mathrm{r}_\mathcal{F}}(\omega_\mathrm{l} + \Delta_\omega) & \cdots & \tilde{K}_{\mathrm{r}_\mathcal{F}}(\omega_\mathrm{u}) \end{bmatrix}, \end{aligned} \tag{19}$$

respectively, the optimisation procedure to assign the eigenvalues Σ_opt of the parametric model of system described in Eq. (15) can be formulated using least-squares optimisation as:

$$\Sigma_\mathrm{opt} = \arg \min_{\Sigma} ||K_\mathrm{fr} - \tilde{K}_\mathrm{fr}||_2^2, \tag{20}$$

where the elements of the set Σ are chosen to have a negative real part, so that the final SS model is internally stable.

B. Application case

In order to be able to use FOAMM in BEMRosetta, the location of the FOAMM executable file has to be specified in BEMRosetta's options. Additionally, the correct Matlab Runtime version needs to be installed on the computer. It should be noted that both the executable file and the Matlab Runtime installer can be downloaded for free from the COER website[10].

Once the frequency domain hydrodynamic coefficients of a geometry are loaded in BEMRosetta, a SS model approximating the radiation subsystem of the selected DoF can be computed from BEMRosetta's 'FOAMM State Space' tab (inside the 'Hydrodynamic Coefficients' tab). As Fig. 6 illustrates, BEMRosetta shows the radiation frequency response $K_r(\omega)$ of the loaded geometry and, using the graphical interface, the desired interpolation frequencies and frequency range for the optimisation procedure can be selected. After the selection of the desired options, pressing the button 'Identify' explicitly calls the FOAMM executable file and identifies an approximated model of the selected DoF. It should be noted that several DoFs can be selected, with different options for each one, and BEMRosetta will run FOAMM in an iterative fashion, obtaining a separate Single-Input Single-Output (SISO) SS model for each DoF.

Once the optimisation procedure of FOAMM finishes, illustrated with a wait-bar, BEMRosetta shows the frequency response of the obtained SS model along with the original frequency response in the 'State Space' tab, as illustrated in Fig. 7. In addition to the graphical results, the matrices of the obtained SS model, the value of the interpolation frequencies, the approximation frequency-range and the Mean Average Percentage Error (MAPE) is provided in the 'State Space' tab.

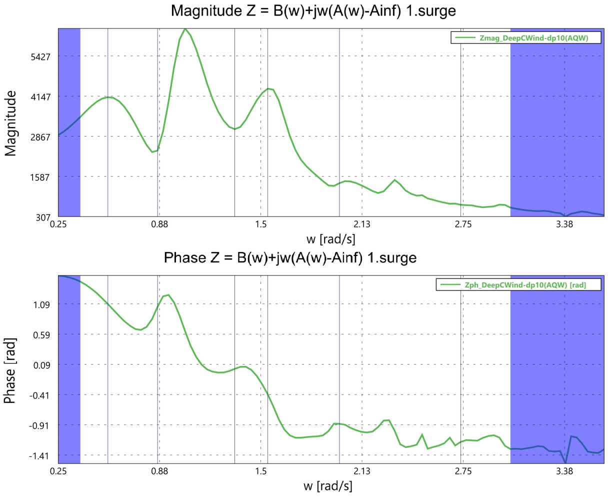


Fig. 6: Screenshot of BEMRosetta's graphical interface to select the desired interpolation frequencies and frequency range for the approximation of FOAMM. The plots represent the magnitude and phase of the K_r parameter (shown as Z) for surge degree of freedom of DeepCwind semisubmersible platform.

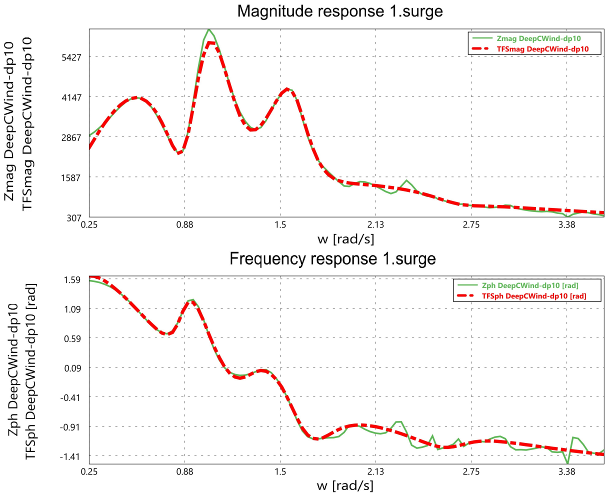


Fig. 7: Frequency response computed from the original hydrodynamic data (green solid line) and that computed from the SS model obtained by FOAMM (red dashed line).

VII. Other features, areas for development and comments

A. Other features

BEMRosetta includes an OpenFAST .out/.outb file reader with user interface. It includes an online mode for updating the plot while the simulation is in progress. Files may be opened by drag and drop, and parameters are filtered by name or units.

A new irregular frequencies removal module has been added. This makes use of filtering techniques, pattern

[10] https://coer.maynoothuniversity.ie/downloads/

matching and Ogilvie's relation to improve radiation damping $B(\omega)$, reconstruct added mass $A(\omega)$, and obtain a finer μ_∞, less influenced by irregular frequencies, discontinuities or distortions in added mass results,

B. Possible areas for development

There are some areas of improvement that are being considered:

- Mesh visualisation: Actual wire mesh visualisation can be changed with including a more featured 3D rendering engine showing surfaces with back-face culling and with interactive handling of the objects.
- RAO calculation: The response amplitude operator is currently read from files, stored, visualised and saved. And additional feature would be to calculate it. As the equation of motion in six degrees of freedom following matricial notation is

$$[-\omega^2(\mathbf{M}+\mathbf{A}(\omega))+i\omega\mathbf{B}(\omega)+\mathbf{C}]\mathbf{X}(\omega)= \zeta(\omega,h)\mathbf{F}(\omega,h) \quad (21)$$

  where $\mathbf{X}(\omega)$ is the vessel motion, ω is the frequency, $\mathbf{M}(\omega)$ is the mass and inertia matrix, $\mathbf{A}(\omega)$ is the added mass, $\mathbf{B}(\omega)$ is the linear damping, $\mathbf{C}$ is the stiffness matrix and $\mathbf{F}(\omega,h)$ is the excitation force per unit of incoming wave height of wave heading h, and $\zeta(\omega,h)$ is the amplitude of the wave. As the RAO is the motion amplitude related to the wave amplitude

$$\mathbf{RAO}(\omega)=\frac{\mathbf{X}(\omega)}{\zeta(\omega,h)} \quad (22)$$

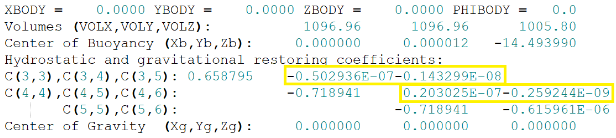
  Replacing Eq. (21) in (22) holds.

$$\mathbf{RAO}(\omega,h)= [-\omega^2(\mathbf{M}+\mathbf{A}(\omega))+j\omega\mathbf{B}(\omega)+\mathbf{C}]^{-1}\mathbf{F}(\omega,h) \quad (23)$$

  This equation can be solved for each frequency and heading with an assumable set of matrix algebra operations.
- Unit tests improving: To improve the robustness of the calculations and to avoid that code modifications may affect the already implemented features, a wide range of proven test cases should be included and always checked after code changes have been made and published.

C. Some comments

Handling different files not designed to be compatible between them, especially with proprietary formats, is usually tricky (as an example, see a special case in Fig. 8). Without additional information, it cannot be completely certain that the file parser and the converter are fully correct. Because of this, the user support is key to find and report problems. Software tools, unlike hardware tools, improve with use. Therefore, users are asked to report any problems or suggestions they may encounter. In this way, within a short period of time, the user that reported the problem and the whole community will be able to enjoy a better and more reliable tool.



```
XBODY =    0.0000 YBODY =    0.0000 ZBODY =    0.0000 PHIBODY =   0.0
Volumes (VOLX,VOLY,VOLZ):           1096.96      1096.96      1005.80
Center of Buoyancy (Xb,Yb,Zb):     0.000000     0.000012   -14.493990
Hydrostatic and gravitational restoring coefficients:
C(3,3),C(3,4),C(3,5): 0.658795    -0.502936E-07-0.143299E-08
C(4,4),C(4,5),C(4,6):             -0.718941     0.203025E-07-0.259244E-09
       C(5,5),C(5,6):                          -0.718941    -0.615961E-06
Center of Gravity  (Xg,Yg,Zg):     0.000000     0.000000     0.000000
```

Fig. 8: Example of a particular and valid case in Wamit where two fields are joined together.

VIII. Conclusion

Interoperability between work teams, and the facilitation of the use of open source tools, have been the main motivations behind the development of BEMRosetta. BEMRosetta can work with different mesh formats, calculation input files and hydrodynamics coefficients which may be obtained from various different BEM tools. In this way, BEMRosetta makes it possible to share information, compare results, and perform post-processes between organisations, and facilitate the work of both the researcher and the engineer. Other functionalities will be added to BEMRosetta in the future, always with the aims of ease of use, compactness and compatibility with scripting languages, while all the time minimising the need for expensive external resources. BEMRosetta is made for the community and by the community. As such the authors welcome all contributions from researchers who can provide new ideas, corrections or simply give moral support.

IX. Acknowledgments

The collaboration and support of the scientific community in the development of this tool is gratefully acknowledged, with special mention of the contribution of J. C. Portillo, M. J. Sanchez-Lara, J. Galván and A. Otter.

Some file parsing strategies are taken from the BEMIO project[11].

We would like to thank the École Centrale de Nantes researchers A. Babarit, G. Delhommeau and other developers for making NEMOH freely available under the Apache 2.0 license. With their work and their example they paved the way for other developers like M. Ancellin and Y. Liu to create more and more advanced tools like Capytaine and HAMS.

References

[1] A. Babarit and G. Delhommeau, "Theoretical and numerical aspects of the open source BEM solver NEMOH," in *11th European Wave and Tidal Energy Conference (EWTEC2015)*, ser. Proceedings of the 11th European Wave and Tidal Energy Conference, Nantes, France, Sep. 2015. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01198800

[2] M. Ancellin and F. Dias, "Capytaine: a python-based linear potential flow solver," *Journal of Open Source Software*, vol. 4, no. 36, p. 1341, 2019. [Online]. Available: https://doi.org/10.21105/joss.01341

[3] Y. Liu, "HAMS: A frequency-domain preprocessor for wave-structure interactions—theory, development, and application," *Journal of Marine Science and Engineering*, vol. 7, no. 3, p. 81, 2019.

[4] M. Penalba, T. Kelly, and J. Ringwood, "Using nemoh for modelling wave energy converters: A comparative study with wamit," in *12th European Wave and Tidal Energy Conference (EWTEC)*, 2017. [Online]. Available: http://mural.maynoothuniversity.ie/12466/

[11] http://wec-sim.github.io/bemio

[5] N. Faedo, Y. Peña-Sanchez, and J. V. Ringwood, "Finite-order hydrodynamic model determination for wave energy applications using moment-matching," *Ocean Engineering*, vol. 163, pp. 251–263, 2018.

[6] A. N. Robertson, F. Wendt, J. M. Jonkman, W. Popko, H. Dagher, S. Gueydon, J. Qvist, F. Vittori, J. Azcona, E. Uzunoglu, C. G. Soares, R. Harries, A. Yde, C. Galinos, K. Hermans, J. B. de Vaal, P. Bozonnet, L. Bouy, I. Bayati, R. Bergua, J. Galvan, I. Mendikoa, C. B. Sanchez, H. Shin, S. Oh, C. Molins, and Y. Debruyne, "OC5 project phase II: Validation of global loads of the DeepCwind floating semisubmersible wind turbine," *Energy Procedia*, vol. 137, pp. 38–57, oct 2017.

[7] J. N. Newman and C. H. Lee, "Wamit user manual version 7.4," available online at: www.wamit.com/manualupdate/v74_manual.pdf (Last accessed date 2021/02/07), 2020.

[8] T. F. Ogilvie, "Recent progress toward the understanding and prediction of ship motions," in *5th Symposium on Naval Hydrodynamics*, ser. Proceedings of the 5th Symposium on Naval Hydrodynamics, Bergen, Norway, 1964, pp. 3–80.

[9] P. Ricci, "Time-domain models," in *Numerical Modelling of Wave Energy Converters*, M. Folley, Ed. Elsevier, pp. 31–66.

[10] W. E. Cummins, "The impulse response function and ships motions," *Schiffstechnik*, vol. 9, pp. 101–109, 1962.

[11] Y. Peña-Sanchez, N. Faedo, M. Penalba, G. Giorgi, A. Mérigaud, C. Windt, D. García Violini, L. Wang, and J. V. Ringwood, "Finite-Order hydrodynamic Approximation by Moment-Matching (FOAMM) toolbox for wave energy applications," in *Proceedings of the 13th European Wave and Tidal Energy Conference (EWTEC) Naples, Italy*, 2019, pp. 1448–(1–9).