Programme Area: Marine

Project: PerAWAT

Title: Implementation of Wave Energy Converters in Spectral Wave Models

Abstract:

This deliverable is the first within the Wave sub-project. It describes the representation of wave energy converters in a third generation spectral wave model and begins by providing some background about the current state-of-the art for wave energy converter array representation. This is followed by the development of a general methodology for representing wave energy converters in a spectral-domain model, where the term spectral-domain model here refers to a numerical representation of the wave energy field which has been decomposed into frequency components. Emphasis is placed on the ability of a spectral domain model to capture the nonlinear characteristics of a wave energy converter. Initial cross validation is provided by comparison of the derived spectral-domain model with a time-domain model in two separate examples of nonlinear wave energy converter characteristics (wave force decoupling, and quadratic drag). Additional cross-validation of the method is demonstrated though comparison with physical wave tank experiments carried out at Queen's University Belfast. The deliverable subsequently focuses on the specific type of spectral-domain model that will be implemented in WG1 WG2, the spectral wave model, in which the wave energy converters will be represented.

Context:

The Performance Assessment of Wave and Tidal Array Systems (PerAWaT) project, launched in October 2009 with £8m of ETI investment. The project delivered validated, commercial software tools capable of significantly reducing the levels of uncertainty associated with predicting the energy yield of major wave and tidal stream energy arrays. It also produced information that will help reduce commercial risk of future large scale wave and tidal array developments.

Implementation of wave energy converters in spectral wave models

WG1 WP2 D2

DOCUMENT CONTROL SHEET

Client	Energy Technology Institute
Contact	Geraldine Newton-Cross
Project Title	PerAWaT
Document N°	QUB-101123
Classification	Not to be disclosed except in line with the terms of the Technology Contract
Date	17th February 2011

REV.	Issue date	Purpose of issues	Prepared by	Checked by
0.01	7/12/2010	Draft for review	KS	MF
0.02	12/12/2010	Changes for comments	KS	MF
0.03	14/12/2010	Draft for comments from GH	KS	MF
0.04	15/12/2010	Executive summary and references added	KS	MF
1.00	22/12/2010	Preliminary release for GH/ETI	KS	MF
2.00	29/12/2010	Release for GH/ETI	MF	
3.00	17/02/2011	Revised in response to comments from ETI	KS	MF

Approved for release by: Matt Folley

Contents

Executive Summary

This document addresses the practical aspects of the implementation of a wave energy converter representation in a third generation spectral wave model. The first section includes an introduction which contains the scope of the document and the relationship of this document to other deliverables. A summary of the closely related deliverable WP1 WG2 D1, which describes the physical representation of wave energy converters in a third generation spectral wave model, is provided, and finally the acceptance criteria for this deliverable are listed and addressed.

The second section contains an overview of the development strategy. First, methods for achieving the key desirable characteristics of the new software (flexibility, user-friendliness, and reliability) are described. This is followed by a discussion of how the software modification will be carried out on four separate key elements successively, and then the integration of these elements that will provide the final software tool. The key elements are identified as: representation of a wave energy converter, location of the wave energy converters, input of wave energy converter parameters, and output of wave energy converter power capture. The software development tools which will be used for the project are described and the reason for the specific tool choice is explained. The Microsoft Visual Studio program will be used for interactive development of the code, and the Git source code management (or revisioning) software will be used to manage changes to the code.

The third section addresses the choice of a suitable third generation spectral wave model for the project. Two open source models, SWAN and TOMAWAC, are identified as good candidates and a close comparison of the two models is carried out. The comparison includes an assessment of the physical processes represented in the models and their ease of use and modification. Two test cases are implemented in both models to aid in the comparison. Although the models solve the same equation in a totally different way, it is shown that the results are very similar, and that neither model can be eliminated as a potential candidate based on physical process representation. The final model choice is TOMAWAC, because it was developed at EDF which is associated with the PerAWaT project and support for interpretation of the source code is more readily available. However, both models were deemed suitable for the task, and therefore should there be an unforeseen problem with TOMAWAC, it would be possible to proceed with SWAN.

The fourth section describes each of the four core elements in detail, and outlines the development process for each of them. New subroutines, and existing subroutines and variables which require modification are identified. Additionally, for each core element the method for verification that the added code is working correctly is described.

The fifth section contains a discussion of the validation of the representation of wave energy converters in a spectral wave model. Two key parameters which will be addressed during the validation process are identified as wave energy converter density, and wave energy converter performance. Because there is no wave farm data available for the validation process, the spectral representation of wave energy converters will be compared with both the time-domain model WaveFarmer being developed at Garrad Hassan, and the wave tank experimental data which is due to be carried out at Queen’s University Belfast as part of the PerAWaT project.

1 Introduction

1.1 Scope of document

The purpose of this document (WG1 WP2 D2) is to describe the details for implementing a representation of wave energy converters (WECs) in a third generation spectral wave model. The physical representation of WECs in a spectral wave model has been described in detail in a complementary deliverable (WG1 WP2 D1). In this document, the practical aspects of the software code development are discussed. Throughout this document, the implemented representation of wave energy converters in a spectral wave model is referred to as the SpecWec tool.

Following this introduction Section 2 provides an overview of the software code development strategy, including the tools that will be used to support this process. Section 3 describes the choice of a third generation spectral wave model that will be the base software model in which the representation of WECs will be developed. This includes detailed descriptions of two candidate spectral wave models, TOMAWAC and SWAN, and the rationale for the selection of which of these will be the base software model. Section 4 explains the core elements of the modifications to the spectral wave model required for the representation of WECs to be implemented. Key existing subroutines and variables in the spectral wave model that need modification are identified, together with the requirement for new subroutines and variables. Finally, Section 5 outlines the procedures that will be used for both verification of the SpecWec tool with other software models and validation against wave tank experimental data to ensure that the SpecWec correctly represents WECs as defined in WG1 WP2 D1.

1.2 Relationship to other deliverables

This deliverable (WG1 WP2 D2) describes how wave energy converters will be implemented in a third generation spectral wave model. This is closely related to the concurrent deliverable (WG1 WP2 D1), which justifies the choice of spectral wave models for the task, and describes the physical representation of wave energy converters in a third generation spectral wave model. The final section of this deliverable looks forward towards WG1 WP2 D4, which will involve comparison of the SpecWec tool with the linear and nonlinear hydrodynamic models produced by Garrad Hass and the University of Oxford (WG1 WP1), as well as experimental validation of the model to be performed by Garrad Hassan and at Queen's University Belfast (WG2 WP1 & WG2 WP2).

1.3 WG1 WP2 D1-Spectral models of wave energy converters

Because WG1 WP2 D1 (*Representation of Wave Energy Converters in Spectral Wave Models*) is so closely linked to this document, a brief summary follows to inform the reader of its contents. The representation deliverable begins by providing some background about the current state of wave energy converter array representation in spectral wave models. This is followed by the development of a general methodology for representing wave energy converters in a spectral-domain model, where the term “spectral-domain model” here refers to a numerical representation of the wave energy field which has been decomposed into uncorrelated frequency components. Emphasis is placed on the ability of a spectral-domain model to capture the nonlinear characteristics of a wave energy converter. Verification is provided by comparison of the derived spectral-domain model with a time-domain model in two separate examples of nonlinear wave energy converter characteristics (wave force decoupling, and quadratic drag). Additionally, validation of the method is demonstrated though comparison with physical wave tank experiments carried out at Queen’s University, Belfast.

The deliverable next focuses in on the specific type of spectral-domain model that will be implemented in the spectral wave model. Background information is provided about the development of spectral wave models. Then, the wave action density equation, which is the main equation solved by spectral wave models, is described in detail. The relationship of spectral wave models to other types of wave models, including time-domain models such as mild slope equation and Boussinesq models, is discussed. In particular, a major difference between spectral wave models and other wave model types is the phase averaging assumption, which says that the phases of individual waves in a wave field are represented with a random distribution. Previous numerical studies have suggested that spectral wave models may be incapable of an accurate representation of wave energy converter because the phase averaging assumption doesn’t allow for capture of array interaction processes. However, these studies need careful interpretation because they generally do not represent realistic situations. A simple numerical model of two wave energy converter arrays is used to demonstrate that when a more realistic setup is used (i.e. one that accounts for uncertainties associated with the wave field and the device characteristics) array interaction factors are reduced to a level where individual interactions are insignificant; this provides justification that a phase averaging model can adequately represent a wave farm.

Next, the formulation of a wave energy converter in a spectral wave model is addressed. The few published studies which have implemented an array of wave energy converters in a spectral wave model have used a supra-grid representation. The strengths and weaknesses of this approach are discussed. A new, sub-grid, representation is described and justified for use in the WG1 WP2 work plan. This sub-grid method will allow wave energy converters to be represented as source terms in the wave action equation. The advantage of this method is the ability to easily represent the frequency, directional, and even wave state dependence of the wave energy converter effect on wave action.

Finally, the derivation of the specific framework to be used for incorporating a wave energy converter in a spectral wave model is presented. The source strength of wave energy converter in the wave action equation is broken down into three major physical mechanisms, including reflection of incident wave energy due to the presence of the wave energy converter, radiation of energy due to the motion of the wave energy converter and extraction of energy both by the wave energy converter and by dissipation of energy due to turbulence around the device. Mathematical expressions for the source strength of each of these mechanisms are produced. The method for solving the source terms is described, and an example for a wave energy converter is presented.

1.4 WG1 WP2 D2 acceptance criteria

The acceptance criteria for this deliverable and WG1 WP2 D1 were defined together, and are as follows:

1. Report contains full description of theory and underlying assumptions in model to enable a full review by a third party.
2. Report contains sufficient detail for implementation of model in software code to be understood by a third party together with specification of test cases.
3. Model covers full range of WEC FDCs as agreed in WG0

The first criterion applies to the Representation deliverable (WG1 WP2 D1). The second criterion applies to this deliverable, and the final criterion is applicable to both deliverables. In order to satisfy the second criterion with this deliverable, the software tools and models that will be used to implement wave energy converters in a spectral wave model are described in detail, and the choice of those particular tools/models are justified. The implementation plan is broken down into four core elements, and for each element a description of the current relevant subroutines, variables, and new subroutines to be produced is provided. The way in which the functionality of each core element will be verified is specified in section 4 of this document. The implementation plan described is designed to be flexible, as outlined in the development strategy, and therefore allows the inclusion of the full range of WEC FDCs as agreed in WG0.

2 Overview of development strategy

2.1 Objectives of development strategy

The objective of WP1 WG2 in the PerAWaT project is to extend the functionality of an existing third generation spectral wave model to allow the representation of wave energy converters (WECs) and for this to be used to calculate the wave farm energy yield. The structure of the representation must be sufficiently flexible to allow any WEC to be represented and allow the layout of the wave farm and WEC control parameters to be modified to permit optimisation of the wave farm energy yield. The development strategy must support these objectives.

The three key desirable and necessary characteristics of the SpecWec tool are as follows:

2.1.1 The SpecWec tool must work reliably.

Here, a reliable software code is defined as one which runs consistently without crashing and is able to deal with a number of different user input scenarios. This requirement for SpecWec will be addressed during the development process as follows. The modifications/additions that need to be made to the spectral wave model have been divided into sections, here called core elements, which are defined in Section 4 of this document. Each of these core elements will be developed independently, and functionality testing (also outlined in Section 4) will be performed on each core element before they are combined together into the SpecWec tool. Finally, validation testing will be carried out on the completed SpecWec tool to confirm its reliability.

The modular development of the SpecWec tool will be aided by the use of a source code management (or versioning) software. This will allow the core elements of the model to be developed in different branches which can then be merged together once the functionality and reliability of the code has been confirmed. Source code management software allows multiple users to develop simultaneously with the ability to merge their work at any time. Additionally, source code management software tracks all the changes that are made as a code is developed, which allows the developer to compare current and previous codes side by side if an issue develops.

2.1.2 The SpecWec tool must be user-friendly.

The SpecWec tool which results from this project will be used by device developers to design array configurations of their specific wave energy devices. There are two kinds of potential users: those who wish to use SpecWec, and those who wish to develop the source code of SpecWec to represent a novel wave energy converter or class of wave energy converters. For both user categories, a clearly documented users' manual will be produced which will include a description of the representation of wave energy converters as well as explanation of all the possible user inputs and outputs which will be needed to run the model successfully. In addition, simple test cases including results files will be provided for users so that they can verify the SpecWec tool works on their system. For developer users of SpecWec, a document containing descriptions of the relevant subroutines will be produced. Also, all additions/modifications to the base spectral wave model source code will be supported by clear comments that include variable definitions, summaries of subroutines, and the author of the changes.

2.1.3 The SpecWec tool must be flexible.

As described in the acceptance criteria for this deliverable, the final software product designed for modelling arrays of wave energy converters will need to be able to allow for several fundamental design concepts. As such, it seems quite likely that the parameters needed to represent a wave energy converter will vary by device. Therefore efforts will be made to ensure that the model development is sufficiently flexible. Hardwiring of variables and parameters will be avoided, and defaults will be assigned for all parameters to ensure that each one doesn't need to be specified for each device. The functionality of SpecWec will incorporate the ability to represent each distinct fundamental design concept with its own subroutine that can be used to calculate the response of the device to the ambient wave field.

2.2 Overview of the development process

2.2.1 Description of the development process

The modification of a spectral wave model into the SpecWec tool will be broken down into four core elements: the input of wave energy converter parameters, specification of wave energy converter location, representation of wave energy converter performance, and the output of wave energy converter performance. These core elements are described in detail in Section 4 of this document. Software development will occur on each core element independently, including stability and functionality testing, Figure 1. Once all the core elements have been tested, they will be merged into a single tool which will then also be tested for functionality and stability.

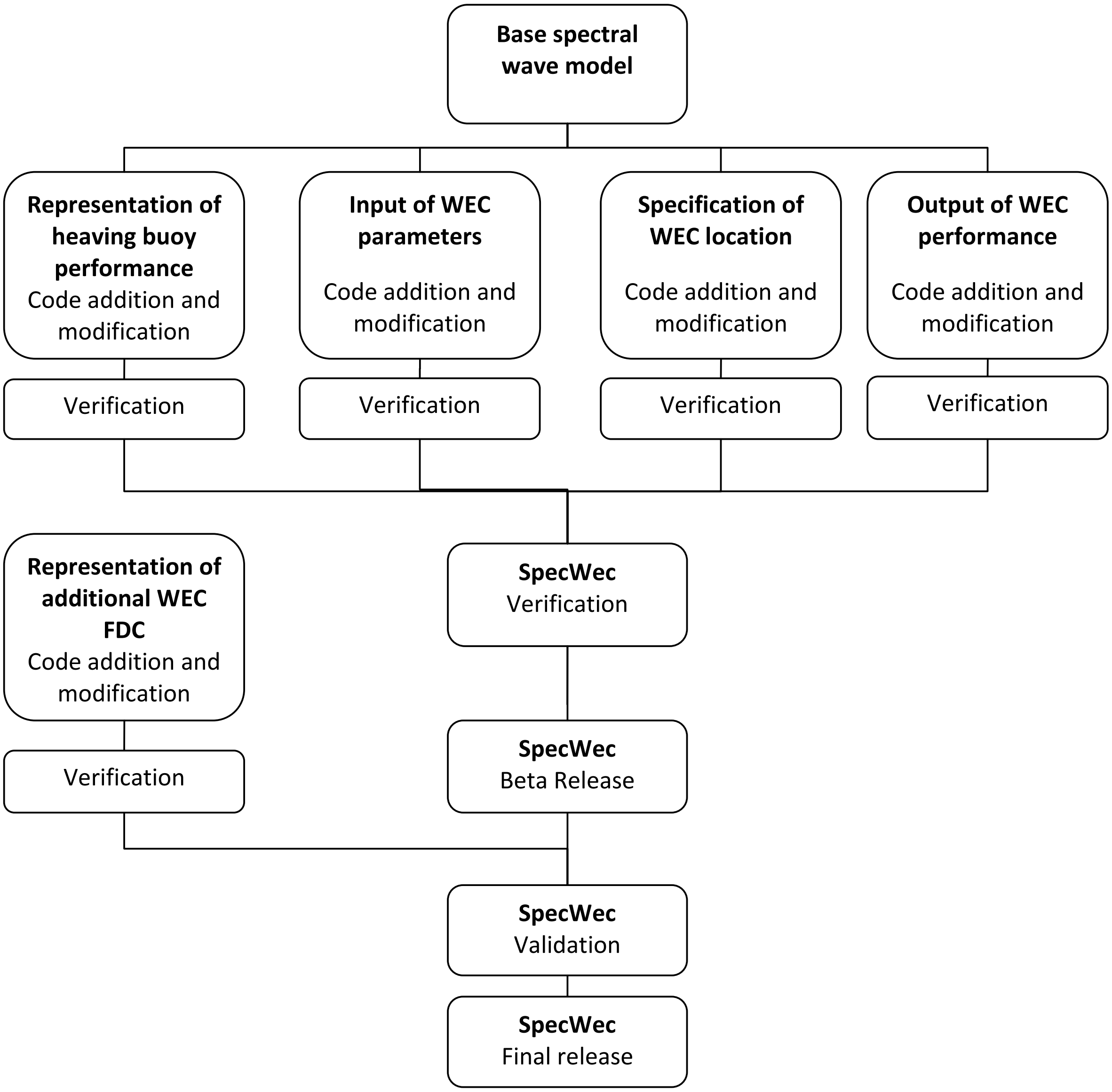

Figure 1: Schematic showing the development process for SpecWec.

2.2.2 Choice of development tools

The structure of third generation spectral wave models is generally highly complex, with multiple sub-routines and modules, contained within a number of different files and folders. To support this effectively it is beneficial to use an integrated development environment, which provides a graphical user interface for dealing with the code. There are several advantages to using such a development environment, including debugging support and code compilation support.

The debugging features of development environments are extremely useful for identifying issues with software code. Breakpoints can be set at various lines in the code, and can include a variable dependence (i.e. the breakpoint will not be triggered until a variable reaches a certain user-defined value). Once the model run is paused at a breakpoint, the simulation can be stepped forward one line or subroutine at a time. During the code stepping, the value of each current variable can be tracked, and the list of routines which have been called are displayed.

Development environments can also track which subroutines have been changed and when compiling and linking only compile the changed routines, which can make the build process much more efficient. Also, they have an extensive search tool which can look through an entire project at once, instead of having to go through each subroutine individually.

The development tool that will be used in this project is Microsoft Visual Studio 2008, which accompanies the Intel Visual Fortran compiler. This development environment has all the features described above, and has a long history of use, and therefore support and documentation. Microsoft Visual Studio includes a code editor which recognizes keywords for the Fortran programming language, and highlights them for easy recognition during software development. It also performs a background compilation as the code is being written in order to identify potential issues during development. This allows the user to save time by correcting problems during the writing of code, instead of trying to troubleshoot when the code doesn’t compile.

2.2.3 Management of code modification

In addition to using a development environment it is beneficial to employ a dedicated source code management tool to track changes to the code using a versioning system, which works as follows. The initial source code is “checked out” (or copied) to a user, who then makes modifications. When they have reached a certain point, for example completion of the coding of a subroutine, they can then save a version of the code. More than one user can “check out” a copy of the source code, which creates two or more branches in the code revision history. These branches can be merged together once the development of a certain section of code has completed. Source code management software tools allow a side by side comparison of code from subsequent versions, which makes it easier to spot errors.

The source code management system which will be used for this project is the open source program Git. Git was initially developed for Linux machines by Junio Hamano and Linus Torvalds (the developer of Linux). Git has now expanded to the Windows platform and is fully compatible with Microsoft Visual Studio. Git is known for its efficiency when working with large projects; the Git website describes it as

> "… commonly an order of magnitude faster than most other version control systems, and several orders of magnitude faster on some operations. It also uses an extremely efficient packed format for long-term revision storage that currently tops any other open source version control system."

http://git-scm.com/about

3 Selection of spectral wave model

3.1 Preliminary analysis of spectral wave models

WG1 WP2 D1 describes in detail the reasons that third generation spectral wave models were chosen for the task of numerically representing wave farms. The following section describes the third generation spectral wave models which were identified as potential base packages that could be modified for this task. Although there are currently several third generation spectral wave models in existence, two models were chosen for close evaluation. These particular models were chosen because they are widely used and accepted, well documented, and are open source which means that modifications can be made to the structure of the code itself. More specifically, both models are available under the GNU General Public License (for details see http://www.gnu.org). The first was the SWAN model, which was chosen because it has been used extensively for many years. The second model which was chosen was the TOMAWAC model, which was chosen because it was developed by one of the PerAWaT participants, EDF. In the rest of the section, the models are evaluated and compared and the final choice is documented.

3.2 Detailed review of SWAN

3.2.1 Background information

The Simulating WAves Nearshore (SWAN) model is a third generation spectral wave model developed at the Delft University of Technology in the Netherlands. Like all third generation spectral wave models, SWAN solves for the evolution of the wave action on a four dimensional grid with two horizontal spatial dimensions, frequency, and time. SWAN is an extension of deep-water third generation spectral wave models (such as WAM) to coastal regions, and is capable of accounting for coastal wind-wave processes associated with shallow water depth and background currents. According to the SWAN Technical Manual, "*the following wave propagation processes are represented in SWAN:*

1. propagation through geographic space,

2. refraction due to spatial variations in bottom and current,
3. diffraction,
4. shoaling due to spatial variations in bottom and current,
5. blocking and reflections by opposing currents and
6. transmission through, blockage by or reflection against obstacles.

And, the following wave generation and dissipation processes are represented in SWAN:

1. generation by wind,
2. dissipation by whitecapping,
3. dissipation by depth-induced wave breaking,
4. dissipation by bottom friction and
5. wave-wave interactions in both deep and shallow water.”

SWAN has been freely available and open source for the duration of its existence, which has resulted in extensive academic usage, including work in the field of marine renewable energy. For example, wave energy resources have been estimated using SWAN off the coast of Spain (Iglesias, 2009) and the Swedish west coast (Waters, 2009). Also, a description of the effect of an array of wave energy converters (represented as an obstacle) on the shoreline wave climate off the coast of Cornwall at Wave Hub utilized SWAN (Millar, 2006).

3.2.2 SWAN structure and code

SWAN is written in the Fortran 90 programming language, and is structured as a set of subroutines which are called by a main program. These subroutines are grouped together in various FORTRAN files, with more than one subroutine in each file. This slightly complicates managing modifications to the FORTRAN code, because the whole file must be “checked out” for modification, rather than just the subroutine requiring modification. The model comes with Microsoft batch files and Perl scripts which allow it to be run from a command line environment. A SWAN model run begins with initialization during which the user input is processed and prepares for the run by defining the necessary variables. Next SWAN computes the main time loop, and finally prints any output requested by the user, see Figure 2.

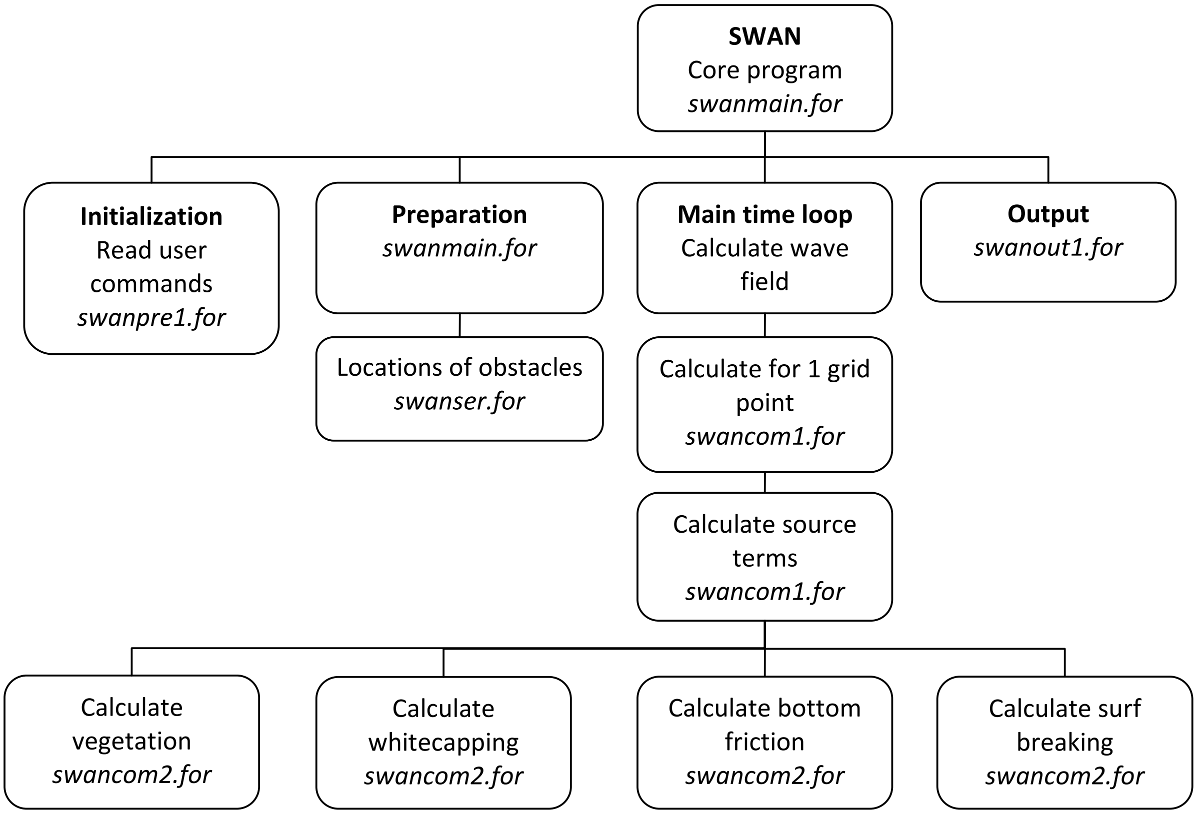

Figure 2: Schematic showing the structure of the computation of the wave action equation in the SWAN model. The name of the relevant Fortran subroutine is italicized.

3.2.3 SWAN numerical solver

All third generation spectral wave models solve the wave action equation (shown below), which consists of a series of convection terms (which act to move wave action around in frequency and horizontal space) and source/sink terms such as whitecapping, bottom dissipation, and wind forcing.

$$\frac{\partial N}{\partial t} + \underbrace{\nabla_x \cdot \left[\left(\vec{c_g} + \vec{U}\right)N\right] + \frac{\partial c_\sigma N}{\partial \sigma} + \frac{\partial c_\theta N}{\partial \theta}}_{Convection\ terms} = \underbrace{\frac{S}{\sigma}}_{Source\ terms}$$

Third generation spectral wave models tend to contain nearly identical parameterizations of source terms; the essential differences between these models lie in the numerical solver applied to the convection terms. SWAN uses a finite difference approximation of the convection terms, which means for example that spatial derivatives can be represented by subtracting the value at one location (x_2) from another (x_1) and then dividing by the distance between the two.

$$\frac{\partial c_x N}{\partial x} \cong \frac{\Delta c_x N}{\Delta x} = \frac{c_x N(x_2) - c_x N(x_1)}{x_2 - x_1}$$

This example is the simplest (first-order) representation of the derivative. However, an advantage of SWAN is that it provides the user with the ability to implement first, second, or third order finite difference representations of derivatives depending on their needs. Although accuracy of the approximation increases with the order of the representation, the use of higher order derivatives also increases the computational time. While traditionally finite difference numerical methods require a regular computation grid, SWAN includes options for implementation of either a regular or flexible grid. This capability makes it much easier to capture complicated model geometries such as a realistic coastline or island boundary.

3.3 Detailed review of TOMAWAC

3.3.1 Background information

The TELEMAC-based Operational Model Addressing Wave Action Computation model (TOMAWAC) was developed as a commercial spectral wave model by the EDF R&D’s Laboratoire National d'Hydraulique et Environnement. TOMAWAC is part of a system of fluid dynamical models (TELEMAC) which include two- and three-dimensional flow solvers, a sediment transport solver, and linear wave dynamics. As a third generation spectral wave model, TOMAWAC solves the same equation as SWAN, the wave action density equation, and is capable of representing the same propagation, wave generation, and wave dissipation processes. The TOMAWAC user manual suggests that “*TOMAWAC can be used for three types of applications:*

1) Wave climate forecasting a few days ahead, from wind field forecasts. This real time type of application is rather directed to weather-forecasting institutes such as Météo France, whose one mission consists of predicting continuously the weather development and, as the case may be, publishing storm warnings.

2) Hindcasting of exceptional events having severely damaged maritime structures and for which field records are either incomplete or unavailable.

3) Study of wave climatology and maritime or coastal site features, through the application of various, medium or extreme, weather conditions in order to obtain the conditions necessary to carry out projects and studies (harbour constructions, morphodynamic coastal evolutions, ...).”

The TELEMAC system was released as open source in June of 2010; because of this there is less academic precedent for the use of the models, including TOMAWAC. Published examples of TOMAWAC usage include the development of a wave atlas for the coasts of France (Benoit, 2004), estimation of sediment transport by waves in a complex coastal region (Brown and Davies, 2009), and erosion of a jetty by wave-induced currents (Abadie et al., 2008).

3.3.2 TOMAWAC structure and code

Like SWAN, TOMAWAC was written in the Fortran 90 programming language, and consists of a main routine which calls in various subroutines, all of which are defined in separate files. TOMAWAC also has associated batch files and perl scripts which allow it to be run from the command line environment. The structure of TOMAWAC is slightly more complex than SWAN because TOMAWAC is implemented within the TELEMAC base structure, but the increase in complexity is not significant. The file input/output subroutines which TOMAWAC uses are shared throughout the TELEMAC model system, and therefore are compiled separately as a library which is read in by the TOMAWAC executable. Within TOMAWAC the code is well structured, with a single file being used for each subroutine, which significantly simplifies the management of modifications. The basic structure of the model is shown in Figure 3, and can be seen to consist of the same four procedures as SWAN: initialization, preparation, main calculation, and output.

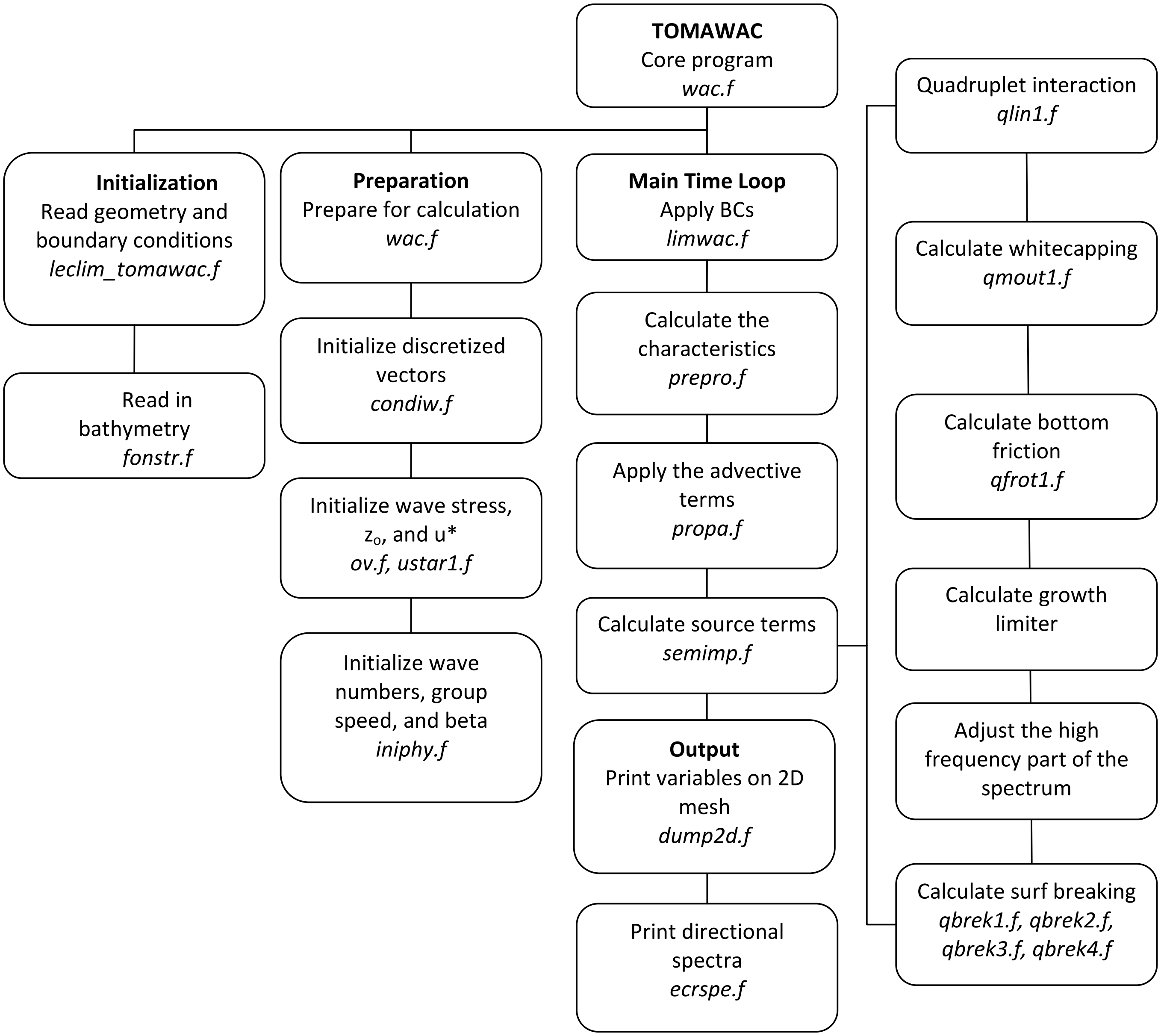

Figure 3: Schematic showing the structure of the computation of the wave action equation in the TOMAWAC model. The name of the relevant Fortran subroutine is italicized.

3.3.3 TOMAWAC numerical solver

As mentioned above, the major difference between the TOMAWAC and SWAN models is the numerical solver applied to the convection terms. Unlike SWAN, TOMAWAC solves the wave action equation in two steps by first finding an interim solution of the wave action equation with no source terms, and then applying the source terms to this interim solution. For the first step, TOMAWAC uses the method of characteristics, which is a technique for transforming the partial differential equation (PDE) which must be solved (the homogeneous wave action equation) into a system of ordinary differential equations (ODEs). This is done by mathematically defining functions in space and time (characteristics) upon which the PDE collapses to ODEs. The full derivation of this method can be found in (Esposito, 1981). The advantage of this method is that in the absence of a background current which changes in time, the characteristics calculation only needs to be performed once, at the very beginning of the model run. During the subsequent time steps, the only operation that is performed is interpolation of the characteristics to the mesh grid, which is substantially less computationally expensive than the characteristic calculation itself.

3.4 Comparison of SWAN and TOMAWAC

3.4.1 Representation of physical processes

As mentioned above, SWAN and TOMAWAC contain equivalent representations of the source/sink terms in the wave action equation. These include: bottom friction dissipation, whitecapping, triad and quadruplet non-linear interactions, wind input, and wave breaking dissipation. Because SWAN and TOMAWAC are phase-averaged models, they do not explicitly resolve processes such as diffraction and refraction. However, the SWAN model does provide a parameterization of the diffraction term which can be switched on and off as needed. Although a similar tool is not included in the current open source release of TOMAWAC, personal communication with the development team indicates that such a tool has been implemented, and will be provided for use in PerAWaT (it is also planned for this to be released in a future TOMAWAC release in 2011 or 2012). The real difference between the two models lies in the treatment of the convection terms, or how the models propagate wave action in horizontal, frequency, and directional space. In order to investigate this difference, a simple test case consisting of waves propagating up a linearly sloping bottom was implemented with all source terms deactivated (Figure 4). The same flexible grid was used for both models, and was 15 kilometres in the along-slope direction, 5 kilometres in the cross-slope direction, and had a horizontal grid spacing of approximately 125 meters and a bottom slope of approximately 1:100, with the depth ranging from 5 to 50 meters. The directional grid spacing was 12 degrees, and there were 26 frequencies in the frequency grid. At the deep water boundary, a Jonswap wave spectrum with 4.0 meter significant wave height and 0.1 Hz peak frequency was introduced. The numerical solver used for the SWAN model was the default second order representation. Both models were run for 120 time steps of 120 seconds in order to reach a stationary state.

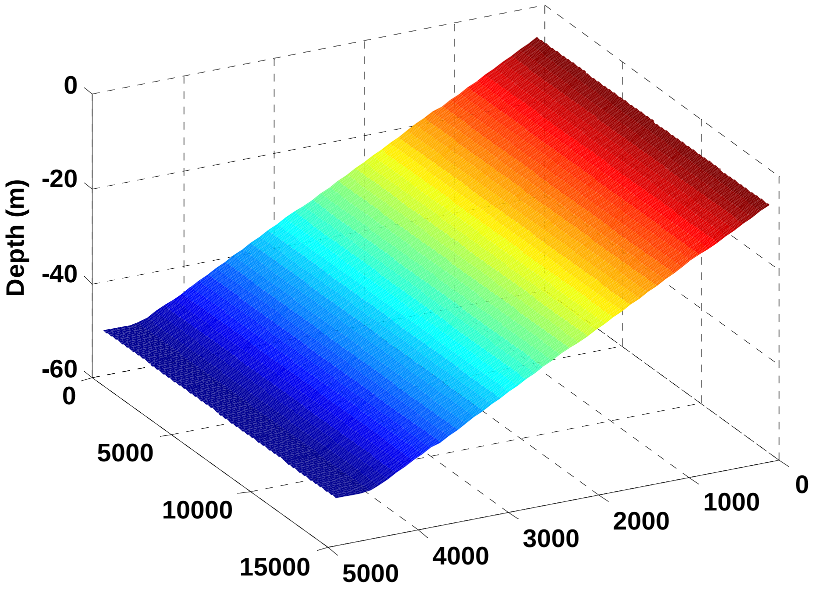

Figure 4: Depth in meters for the test case used to compare SWAN and TOMAWAC.

This test case allows direct comparison of the two models and their ability to represent the changes in wave action due to changes in bottom depth. As can be seen through comparison of the final significant wave height, the models produce nearly identical estimates for the changes in wave height as the energy propagates up the slope (Figure 5).

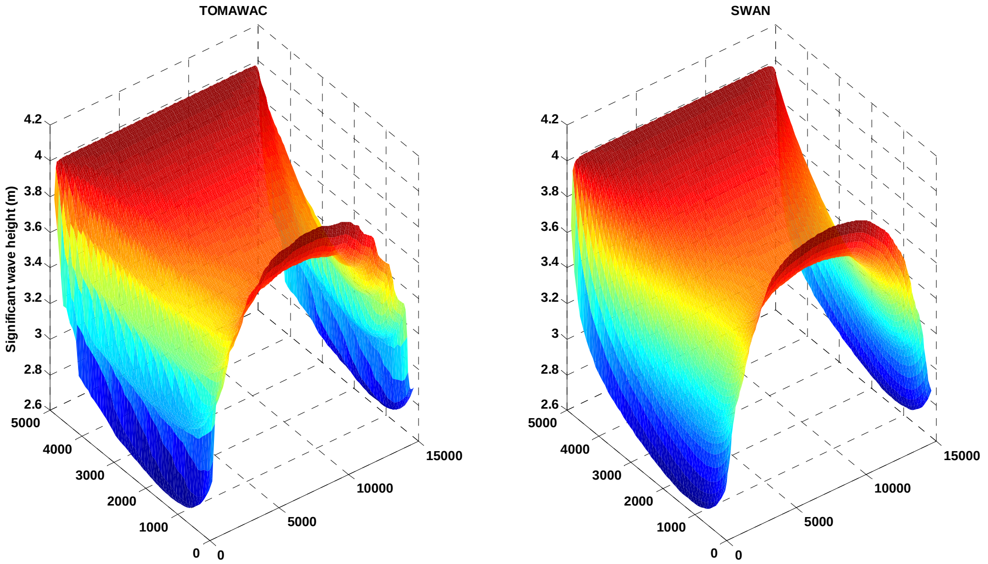

Figure 5: Significant wave height in meters for TOMAWAC (left) and SWAN (right).

As described in detail in WG1 WP2 D1, the representation of a wave energy converter in a spectral wave model will be implemented as a sub-grid element, specifically as a source/sink term in the wave action equation. Representing a WEC as a sub-grid element allows for easy inclusion of the frequency, directional, and sea state dependence. In order to understand both the process of including a source term into the models and the model response to a point source, a second test case was implemented in both SWAN and TOMAWAC in which an artificial source term was introduced. This test case introduced a point source term at the centre of the domain which adds a constant, positive amount of wave action into the system at each time step. As in the previous test case, the meshes for both models were identical and the systems were time stepped for 60 time steps of 120 seconds. The computational grid was the same as the previously described test case, except for the fact that a constant bottom depth was used for simplicity. Modification of both models to include a simple point source was straightforward. The changes to SWAN involved only two subroutines, and the addition of approximately ten lines of code. TOMAWAC modification involved the addition of three lines of code to one subroutine.

The solution to the artificial source term test case quantitatively should consist of concentric circles with their centre at the source point. The significant wave height should decrease as the distance from the source term increases. Both the SWAN and TOMAWAC solutions capture this behaviour (Figure 6). However, with both models, the expected pattern of concentric circles is interrupted by discrete rays. The TOMAWAC solution has more rays than the SWAN solution, which on the whole looks a bit smoother than the TOMAWAC solution. This smoothness may seem to be an indicator of a “better” solution; however the fact that the SWAN model produces smoother results is likely due to its solver being more numerically diffusive than TOMAWAC, rather than a fundamental difference in the solutions. Both of the model results have a starburst shape in which energy is propagated in preferred directions, and this effect is more pronounced in the TOMAWAC results. This is a well-known issue with third generation spectral wave models known as the Garden Sprinkler Effect. Because ocean waves (particularly swell) are dispersive, they continually spread out across the ocean, covering a large area. Spectral wave models use a discrete number of frequencies and directions, which limits this dispersion to distinct directions and frequencies, as can be seen in the model results.

There is a fair amount of established literature which has been produced to address the Garden Sprinkler Effect issue. The simplest way to deal with it is to increase the directional and/or frequency resolution of the model; however this increases the computational load. If it is not possible to eliminate the issue entirely through increased resolution, there are two potential methods for dealing with GSE (Tolman, 2002). The first, which is currently available as an option in SWAN, is the addition of an artificial diffusion term into the model which smoothes out the wave action field. This method can be useful but is computationally intensive. Tolman 2002 suggests that a better method for dealing with GSE is a technique in which the spectrum is averaged in horizontal space to smooth out the results. The advantage of this method is that it is less computationally intensive than the diffusion method. TOMAWAC does not currently include either of these methods, but there is no reason why they could not be implemented.

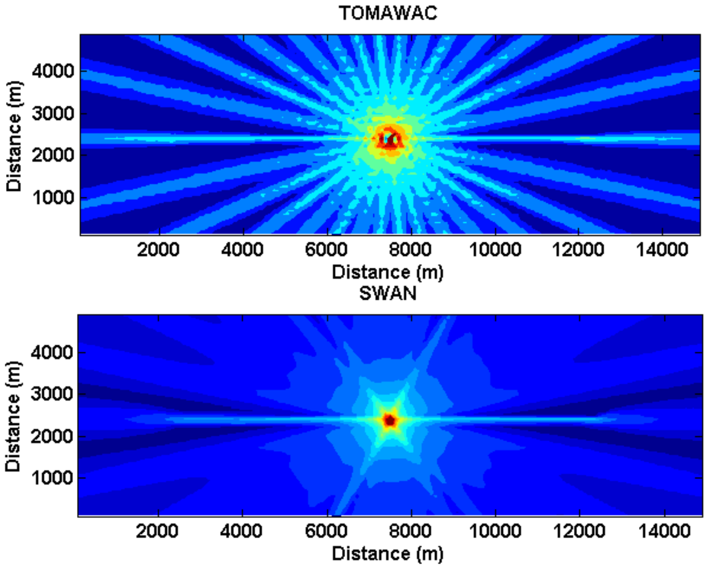

Figure 6: Significant wave height for the TOMAWAC (top) and SWAN (bottom) models.

3.4.2 Ease of use and modification

Both SWAN and TOMAWAC consist of a series of FORTRAN subroutines; however multiple SWAN subroutines are lumped together in a single text file, while TOMAWAC subroutines are each contained in a separate file. This is an advantage of TOMAWAC for source code modification over SWAN, but this issue can be alleviated through the use of an interactive development environment such as Microsoft Visual Studio. On the other hand, the TOMAWAC codes are all commented in French, which is not spoken by the primary code developer, and therefore acts as a disadvantage for code modification. Development teams for both the models have released a document in English containing Programmers Rules, which define clear rules for model modification. EDF is a participant in the PerAWaT program, and therefore the expertise of the development team of TOMAWAC is easily accessible for use in this project, as is a user's forum on the TELEMAC website. SWAN has been open source for many years, and therefore there are numerous users which participate in an active user's forum which can be found on the SWAN website.

For the test cases which are detailed above, the TOMAWAC model tended to have a shorter computation time than the SWAN model. This is because of the use of the characteristics method for solving the convection terms. However, in more sophisticated cases involving background currents which are not steady in time, the TOMAWAC model would become more computationally intensive. Despite the difference, both the model run times were well within an acceptable range (both runs took less than five minutes for the 15km x 5km test case described).

3.4.3 Final selection of software tool

As is detailed above, a close comparison of SWAN and TOMAWAC was undertaken in order to choose which software was most appropriate for representation of wave energy converters. The models were compared in several different aspects, including their representation of physical processes and ease of use and modification. In order to investigate the model differences, two test cases were implemented, depicting wave propagation up a sloping bottom and the introduction of an artificial source term. The discussion which follows addresses the differences found between the two models and details the final choice of which model is more suitable for use in the development of a wave energy converter representation.

SWAN and TOMAWAC, like all third generation spectral wave models, solve the wave action density equation which captures both the propagation of wave energy through a domain as well as the generation and dissipation of waves due to processes like wind and bottom friction. As is mentioned above, the source terms in the wave action equation which represent wave generation and dissipation are identical in SWAN and TOMAWAC, and therefore do not preclude either model from use in PerAWaT. The real difference in the representation of physical processes between the two models lies in the propagation of wave energy, through the solving of the convection terms in the wave action equation. SWAN uses a finite difference scheme, and TOMAWAC uses the method of characteristics. The consequences of these different schemes were investigated with the first test case that was described in the previous section, demonstrating the propagation of wave energy up a sloping bottom. Despite the different solutions of the propagation of energy, the two models produced nearly identical results, suggesting that neither model can be eliminated based on the representation of physical processes.

The other category used for comparison of the two models was ease of use and modification. The two models are structured similarly, and are written in the same programming language. One advantage to TOMAWAC is the fact that subroutines are each contained in separate files, unlike SWAN, which makes it easier to navigate through the code when developing. This advantage is offset by the fact that the TOMAWAC code is commented in French, which is not a language known to the primary code developer, and adds to the time required to modify the code. The second test case which was presented in the previous section involved the introduction of an artificial source term into both models. This modification was simple for both models, and therefore again does not eliminate either model for use in PerAWaT. Although the SWAN results for this test case were smoother, this is simply a reflection of the greater numerical diffusion of the model and not an indication that SWAN is more accurate than TOMAWAC. On balance it is felt that TOMAWAC would be slightly easier to modify than SWAN, although the difference is marginal.

The conclusion drawn from the above close comparison of the two models is that they are quite similar and that either one would be suitable for the purposes of this project. Therefore, the decision of which model to use was primarily based on the availability of support. The fact that EDF is associated with PerAWaT means that close interaction with the development team of TOMAWAC will be available throughout the course of SpecWec development, and this provides an advantage over the SWAN model. This advantage should not be under-estimated; modification of complex software code without support from the original developers is extremely difficult since it is often difficult to determine why a particular code structure has been used simply from the comments. However, that both models are suitable for the task means that if code development in the TOMAWAC model is significantly complicated for an unforeseen reason, than SWAN could be substituted. Therefore, care will be taken during the initial stages of SpecWec development to ensure that the code developed is as portable as possible.

4 Core elements of software tool

4.1 Overview of core elements

Modification of the TOMAWAC model will be structured around four core elements of the SpecWec tool. These core elements have been identified as

- the input of wave energy converter parameters,
- specification of wave energy converter location,
- representation of wave energy converter performance, and
- output of wave energy converter performance.

Because it is desirable to maintain the portability of the code in the event that it would be necessary to switch from TOMAWAC to SWAN, the representation of wave energy converter performance will be developed first. This module is less model-specific than those involving either the computational grid (locating the wave energy converters), or input/output of wave energy converter data. The order of core element development will therefore be:

1. Representation of wave energy converter performance
2. Input of wave energy converter parameters
3. Specification of wave energy converter location
4. Output of wave energy converter performance

Descriptions of how the core elements will be implemented, including key subroutines and variables in the TOMAWAC model, are provided below.

4.2 Representation of wave energy converter performance

As described in WG1 WP2 D1, wave energy converters will be represented by an additional source term. This source term will be associated with the computational node closest to the position of the wave energy converter specified in the steering file. This additional source term will be implemented inside a new subroutine which takes the current wave action density field as input. The new wave energy converter source subroutine will be called by the *semimp.f* subroutine, which is in turn called by the main routine (*wac.f*). The *semimp.f* subroutine contains all of the source term calculation, and therefore is a logical place for the new subroutine.

4.2.1 Key existing subroutines requiring modification

semimp.f is a subroutine called by the main routine *wac.f* which contains all the source term calculations. The code carries out seven main steps, as defined by the TOMAWAC developers:

1. *Allocate the temporal information*
2. *Update the wind*
3. *Calculate average quantities of the directional spectra*
4. *Assess the source terms: generation, quadruplet interaction, and white-capping*
5. *Apply the source terms*
6. *Special high-frequency spectrum treatment*
7. *Apply the wave breaking term*

Modification of *semimp.f* will consist of the addition of the call to a new subroutine, *wecsource.f*, which will contain the wave energy converter source term calculation and occur during step 4.

4.2.2 Key existing variables

F: contains the fully directional frequency spectrum at each computational node, and therefore has three dimensions: the number of nodes, the number of frequencies, and the number of directions.

This variable will be updated in the new subroutine to take into account the presence of wave energy converters.

4.2.3 New subroutines requiring production

wecsource.f: subroutine which will be called by *semimp.f* that will apply the wave energy converter source term to each of the user-specified positions. The structure of *wecsource.f* is dependent on the WEC source term strength derivation presented in WG1 WP2 D1, and will be as follows:

1) Solution of non-linear WEC hydrodynamics and dynamics
2) Calculation of power absorbed by the WEC
3) Calculation of radiated and diffracted wave by WEC
4) Calculation of change in energy flux for wave components
5) Determination of WEC source strength

4.2.4 Verification of code functionality

Verification of the functionality of the representation core element will be used before the core elements are integrated to become the SpecWec tool and will demonstrate that the newly added subroutine, *wecsource.f* performs as desired. In order to achieve this, a shell script will be written to run the subroutine independently. The shell script will define parameters for the subroutine, call the subroutine, and write the source term strength to file. This will be compared with a Matlab version of the same equations to ensure the correct calculation of the source term strength. An analytical solution can be derived for the absorbed power and the radiated power of a linear system (Evans 1980; Falnes 2002); this solution will also be used to confirm the correctness of the SpecWec solution. The shell script will be used for a series of test cases which cover a range of wave energy converter parameters.

4.3 Input of wave energy converter parameters

In order to set up a simulation in TOMAWAC it is necessary to produce a text file, called the steering or case file, which contains all the relevant commands for running the simulation. Because this system is already in place, it is straightforward to modify it in order to include parameters which give information about the wave energy converters in the model domain. All of the possible variables which can be fed into the model using the steering file are defined in a text file known as the dictionary file. The dictionary file contains the name of the variable as well as its default value. Any wave energy converter parameters which are to be included will need to be added to the dictionary file with their default values. Because of the presence of the default values, parameters which are added do not necessarily need to be implemented in each model run.

The parameters which are read in from the steering file are assigned to the proper variable name in a subroutine which is called during the preparation for calculation steps in TOMAWAC. The variables are sorted into four different types: integer, real, logical, and character. This subroutine will need to be modified in order to account for any new parameters which have been added to the dictionary file.

4.3.1 Key existing subroutines requiring modification

tomav6_p0.lib: dictionary file: The dictionary file is not actually a subroutine, but an input file that TOMAWAC uses. It is hardwired to contain all the possible input parameters, including their variable names and default values as well. This file will need to be modified by the addition of any new user inputs, and their default values.

lecdon_tomawac.f90: subroutine which assigns input parameters to the proper variables. The structure of *lecdon_tomawac.f90* is as follows:

1) Open the dictionary and steering files
2) Read in the dictionary and steering files
3) Assign the input parameters to the appropriate model variable
4) Check for any incompatible keywords

This subroutine will need to be modified in conjunction with the dictionary file. The new input parameters which are added to the dictionary file will need to be assigned model variable names, and associated with those names in step 3 of *lecdon_tomawac.f90*.

4.3.2 Key existing variables

ADRESS in the lecdon_tomawac.f90 subroutine – this variable holds the value of all of the input parameters until they are assigned to their individual names. Because TOMAWAC is part of the TELEMAC system, it shares a generalized set of subroutines which are used to read in steering files of all the models in the system. Therefore, when the steering file is read in, the value of each input parameter is stored in a generic variable, *ADRESS*. Then, when the *lecdon_tomawac.f90* routine (which is unique to TOMAWAC) is called, the values in *ADRESS* are transferred into individual variables. This process will need to be updated to include any new input parameters which are defined.

4.3.3 Verification of code functionality

In order to be certain that new wave energy parameters introduced into the TOMAWAC steering file are being read in properly, a comparison between the steering file input and the corresponding variable value directly prior to the main calculation body of the model will be carried out. This can be done with the debugging function of the Microsoft Visual Studio development environment and by including additional lines in the source code which will print the variable values out to a file. The latter method will produce a record which, in combination with the steering file, can then be used to document the correct functionality of this core element.

4.4 Specification of wave energy converter location

The (x,y) coordinates of wave energy converter locations will be designated in a separate ASCII input file which is read into the model during the initialization stages. The name of the location input file will be designated in the steering file. Because each wave energy converter will be represented with a source term at an individual node, there must be a new subroutine added which will identify the nearest computational node to each (x,y) wave energy converter coordinate. It would be possible to expand this subroutine to be able to deal with spherical coordinates as well, however this will not be part of the initial SpecWec tool release, as it is not very likely that the domain encompassing a wave farm will be large enough to require the use of spherical coordinates. It will be important to make sure that the distance between the wave energy converter and the nearest node is within a user-defined acceptable limit; this functionality will be included in the added subroutine. If the user chooses, they can design their computational mesh in such a way that the wave energy converter locations are located directly on a node. This is just a special case of the method in which the distance of a wave energy converter to the nearest node will be zero.

4.4.1 Key existing subroutines requiring modification

wac.f: main TOMAWAC routine which includes initialization procedures. The current structure of *wac.f* is as follows:

1) Initialization of local variables
2) Initialization of discretization vectors for the current, the wind, and the variance spectrum:
3) Preparatory calculations for non-linear interactions
4) Wave stress calculation and calculation of Zo and U*
5) Initial boundary conditions
6) Calculation of wave numbers, group speed, and spectrum factor B.
7) Output graphics for the initial state (Optional)
8) Prepare for propagation
9) Main Time Loop
    a. Increment the time step

  - b. Apply the boundary conditions
  - c. Update the bathymetry and the current
  - d. Prepare for propagation:
  - e. Propagation (interpolation along the characteristics)
  - f. Integrate the source terms
  - g. Pass the absolute frequency (Optional)
  - h. Print output files

The modification of *wac.f* will consist of the addition of a call to the new subroutine *wecloc.f90* which will read in the wave energy converter locations and locate the nearest node point. This new subroutine will be incorporated between steps 2 and 3.

4.4.2 Key existing variables

MESH%XEL%R: contains the x positions of the computational nodes of the mesh

MESH%YEL%R: contains the y positions of the computational nodes of the mesh

The MESH keyword denotes a structure which contains several different variables concerned with the computational grid. The percentage sign is used to designate the individual members of the structure. These variables will be used as input for the new subroutine, *wecloc.f90*, which will locate the nearest computational node to the each wave energy converter location.

4.4.3 New subroutines requiring production

wecloc.f90: subroutine which will read in the wave energy converter positions and find the nearest computational nodes. The subroutine will be structured as follows:

1) Open and read in the wave energy converter location file
2) Loop through each wave energy converter
   - a. Calculate the distance between the wave energy converter and all of the computational nodes
   - b. Find the minimum value of the distance
   - c. Store the node number

This new subroutine will be called during the initialization stages of the main TOMAWAC routine *wac.f*.

4.4.4 Verification of code functionality

The item that needs to be verified for this core element of the software tool is that the computational node associated with each wave energy converter by the subroutine *wecloc.*f is indeed the closest one to it. In order to show this, a shell script will be produced which defines input for the *wecloc.f* routine, calls the subroutine, and then prints a file which contains both the wave energy converter locations and the nearest node positions for comparison. Several test cases will be run with this shell in order to verify the functionality of this core element.

4.5 Output of wave energy converter performance

The TOMAWAC core model has the capability to produce two different kinds of output files: a two-dimensional results file, and a spectral results file. The two-dimensional results file consists of a variable chosen by the user (such as significant wave height or mean direction of the wave field) which is output on the two dimensional spatial grid at a time interval set by the user. The spectral results file contains the directional spectra of wave action at specific spatial positions which have been designated by the user, along with the time interval for output. For the purposes of PerAWaT, a new WEC output file will be incorporated. This file will contain the reference number of each wave energy converter and their power output at a time interval chosen by the user.

4.5.1 Key existing subroutines requiring modification

wac.f: the main FORTRAN routine of TOMAWAC. The structure of *wac.f* is defined in section 4.4.1 of this document. The modification of this subroutine will include a call to the new subroutine *wecwrite.f*, which will produce the WEC output file. This change will be incorporated into step 9h of the *wac.f* routine.

4.5.2 New subroutines requiring production

wecwrite.f90: subroutine which will print out the WEC power output at a user-defined time interval. The structure of the routine will be as follows:

1) Open the wave energy converter output file
2) Write the information, including the reference number for the wave energy converter and the power output for that wave energy converter.
3) Close the wave energy converter output file

4.5.3 Verification of code stability and functionality

The core element's ability to correctly output new parameters associated with wave energy converters will be verified by confirming that the wave energy converter power output files can be read in by the post-processing tools (MATLAB and Blue Kenue), and that the new variable values are consistent with those inside the model (which can be verified through the debugging feature of Microsoft Visual Studio).

5 Validation of the SpecWec tool

Verification and validation are fundamental elements of any software tool development. Verification of the SpecWec tool was dealt with in the relevant parts of Section 4, which describes the core elements of the SpecWec tool. This Section deals with validation.

Validation of the SpecWec tool is required to provide confidence that output generated by SpecWec is a sufficiently accurate representation of a wave farm to produce a reasonable estimate of average energy yield. For the case of SpecWec the critical output is the estimation of the power generated by the wave farm and its sensitivity to design parameters such as the spatial configuration of the wave farm and the control parameters of individual devices. Ideally, the output should be compared to a large commercial wave farm since this is what the SpecWec tool is designed to model. Unfortunately, no commercial wave farms currently exist and it is not expected that they will be developed in an appropriate time scale for this project. Consequently, it is necessary to use alternative representations of wave farms to provide some validation of the SpecWec tool. Technically this is not validation and to emphasise the difference this will be termed cross-validation.

The SpecWec tool shall be cross-validated against two alternative representations of wave farms; a numerical model and a physical model. For the numerical model, the SpecWec tool will be compared to the output of the WaveFarmer time-domain tool being developed by Garrad Hassan in WP1 WG1. The WaveFarmer time-domain model is used because the assumptions inherent in spectral wave models mean that they can only be used to model irregular waves so that comparison with the WaveFarmer frequency-domain model is not possible. For the physical model, the SpecWec tool will be compared to the results of wave-tank testing performed by QUB at their Portaferry wave basin in WG2 WP2 and by Garrad Hassan in WG2 WP1.

In addition to providing two independent sources of cross-validation data, the numerical and physical models provide data with distinct characteristics, which can be usefully exploited. The WaveFarmer time-domain tool requires an explicit mathematical representation of the performance of a wave energy converter, which can also be implemented explicitly using the SpecWec tool (although the actual representation may differ). Thus, differences between the output of WaveFarmer time-domain tool and SpecWec tool must be associated with the intrinsic differences between the two numerical models. In contrast, the wave-tank modelling requires a model of the dynamics and hydrodynamics of the wave energy converters to be developed, which can then be incorporated in SpecWec. Thus, differences between the output of the wave-tank and SpecWec modelling may be associated with intrinsic and extrinsic differences. By comparing the outputs of the three models it should be possible to isolate the intrinsic and extrinsic differences, and therefore apparent validity, of the models.

A key aspect of validation is defining the extent of validity. The extent of validity is the defined by the range of parameters for which SpecWec is valid. There are very few models that are valid in all circumstances and it is likely that SpecWec is no different in this respect. Moreover, it is not always clear the most appropriate model parameters by which the limits of validity should be defined and it is not always practical (due to time constraints) to test all the extents to get a comprehensive picture of validity. Notwithstanding the difficulty in defining extents, for the SpecWec tool it is suggested that two key parameters that should definitely be investigated are the WEC deployment density and WEC type/characteristics. The cross-validation cases described in the following sections are designed to provide both validation and the extents of validity.

If necessary, the SpecWec tool will continue to be refined in response to validation exercises so that both the intrinsic and extrinsic differences are reduced to acceptable levels or are limited by fundamental intrinsic assumptions. Acceptable levels of uncertainty in wave farm productivity depend on the state of the industry but a working level of 10% uncertainty in wave farm productivity will be used initially. There is an inherent assumption that extrinsic differences can be effectively eliminated by production of suitable models of the wave energy converter; if this is not possible then the uncertainty of wave farm productivity should be increased to account for the uncertainty in the extrinsic WEC model.

5.1 Cross-validation against the WaveFarmer time-domain tool

Comparison of output from the SpecWec tool and the WaveFarmer time-domain tool will be structured by increasing levels of complexity in the representation of the individual wave energy converters. This will be used to define the extent of validity of the SpecWec tool in terms of the WEC type/characteristics. Subsequently, comparison of outputs from the two tools with an increasing WEC deployment density will be investigated to determine the extent of validity of this parameter. For these comparisons all the other source term components will be turned off in the spectral wave model so that assumptions inherent in the WaveFarmer time-domain tool are replicated. However, for reference, SpecWec will also be run with the other source term components turned on to provide an indication of the significance the other source term components may have on the wave farm productivity.

Initial comparison will be made for an isolated single degree-of-freedom heaving buoy, with linear hydrodynamics, moorings and power-take-off. Complexity will be increased by adding non-linear elements to the system and increasing the number of devices in the array; recognising that the number of devices will be limited by the run-time of the WaveFarmer time-domain model. Comparisons will be made for a range of different sea-states including narrow-banded, broad-banded & bimodal spectra, sea-states with both low & high directional dispersions and sea-states with a range of steepness & significant wave heights.

The extent of validity for deployment density will be determined by using a regular grid layout of devices and varying the row and column spacing to identify where the output from the SpecWec tool diverges from that produced by the WaveFarmer time-domain tool. As previously, comparisons will be made for a range of different sea-states including narrow-banded, broad-banded & bimodal spectra, sea-states with both low & high directional dispersions and sea-states with a range of steepness & significant wave heights.

Comparison will also be made for the more complex wave energy converter of a floating articulated attenuator. Again the model will initially include only linear elements, and subsequently adding non-linear elements and increasing the number of devices in the array to increase the complexity of the model. From a modelling perspective it is expected that most wave energy converters will be less complicated than the articulated attenuator. Thus, cross-validation of the SpecWec tool for an articulated attenuator provides evidence that it will be valid for most types of wave energy converter. It is expected that the cross-validation of the SpecWec tool for an articulated attenuator will be made after the release of the beta version of the software, but will be completed before the final release.

5.2 Cross-validation against wave-tank models

Cross-validation of the SpecWec tool against wave-tank models requires implementation of a two-stage process. The first stage involves developing a spectral-domain model of an isolated wave energy converter being tested in the wave-tank, whilst the second stage involves implementation of this spectral-domain model in the SpecWec tool to allow it to be modelled in an array. Wave-tank testing also means that it is possible to determine the effect that a non-homogeneous incident wave field may have on wave farm productivity and whether the output of the SpecWec tool is valid in these conditions.

The first stage of cross-validation requires testing a single device in the wave tank and developing of a spectral-domain model of the device that provides a good estimate of its response and power capture. This has already been done successfully for a fixed oscillating water column, which is one of the fundamental device concepts - a point absorber, as described in Deliverable 1 of this workgroup (WP1 WG2) and with sufficient time and ingenuity it could be expected that this could be done for any device. However, the effort required to produce an accurate spectral-domain model may be significant, especially for multiple degree-of-freedom devices such as an articulated attenuator, but this is common for the development of all non-linear models of wave energy converters whether it is in the time-domain or spectral-domain (frequency-domain models are fundamentally linear and so relatively easy to develop, although they generally have limited accuracy).

To support the production of a spectral-domain model a software shell for the wave energy converter source term will be developed. This shell will allow the input of the wave spectrum at the isolated device together with the device's control parameters and the response and power capture calculated. This could be done using a spectral wave model, but this is clumsy because it requires solution of the wave action density propagation, when all that is of interest is the incident wave field at the location of the wave energy converter. This will streamline the production of a suitable wave energy converter source term saving both time and effort.

The second stage of cross-validation involves testing an array of devices in a range of different configurations and sea-states as specified in the previous sub-section, except that the other source-term components should be turned on and at least some of the testing will be done in a non-homogeneous wave field to determine whether the SpecWec tool remains valid in these circumstances. Because of the increased time, effort and expense associated with wave-tank testing it will not be possible to test in as extensive range of conditions as specified for cross-validation against the WaveFarmer time-domain model. However, the objective of identifying the extent of validity remains the same.

Wave-tank testing is planned for both single degree-of-freedom heaving buoys (WP2 WG1 & WP2 WG2) and articulated attenuators (WP2 WG1). The heaving buoys will be tested in both a (nominally) homogeneous wave field and well as a non-homogeneous wave field, which will be achieved by using a variable bathymetry. The articulated attenuators will be tested in a homogeneous wave field only. As discussed above, in each case an isolated device will be tested first to develop the spectral-domain model, followed by testing in array configurations. The isolated devices and arrays will be tested in a range of different sea-states including narrow-banded, broad-banded & bimodal spectra, sea-states with both low & high directional dispersions and sea-states with a range of steepness & significant wave heights. A range of different array configurations will also be tested, including those with a high deployment density to determine the validity of the SpecWec tool for closely spaced arrays.

This sub-section provides an outline of the planned wave-tank testing. More detailed testing programmes, which specify the exact range of tests, are contained in other deliverables. Specifically, Deliverable WG2 WP2 D1 for the testing of a large array of heaving buoys and Deliverable WG2 WP1 D1 for the testing of small arrays of heaving buoys and articulated attenuators.

6 References and bibliography

Abadie, S.M., C. Brière, J. Dubranna, P. Maron, D. Rihouey [2008] "Erosion Generated by Wave-Induced Currents in the Vicinity of a Jetty: Case Study of the Relationship between the Adour River Mouth and Anglet Beach, France*", Journal of Coastal Research*, Vol. 24, pp. 59 – 69.

Babarit A., Borgarino B., Ferrant P. and Clément A. [2009] Assessment of the influence of the distance between two wave energy converters on the energy production. In: Proceedings of 8th European Wave and Tidal Energy Conference, Uppsala, Sweden

Backer G. D., Vantorre M., Beels C., Rouck J. D. and Frigaard P. [2009] Performance of closely spaced point absorbers with constrained floater motion. In: Proceedings of 8th European Wave and Tidal Energy Conference, Uppsala, Sweden

Beels C., Troch P., De Visch K., De Backer G., De Rouck J. and Kofoed J. [2009] Numerical simulation of wave effects in the lee of a farm of Wave Dragon wave energy converters. In: Proceedings of 8th European Wave and Tidal Energy Conference, Uppsala, Sweden

Bellew S., Stallard T. and Stansby P. K. [2009] Optimisation of a Heterogeneous Array of Heaving Bodies. In: Proceedings of 8th European Wave and Tidal Energy Conference, Uppsala, Sweden

Benoit, M., and F. Lafon [2004] "A nearshore wave atlas along the coasts of France based on the numerical modelling of wave climate over 25 years", Coastal Engineering: *Proceedings of the 29th International Conference,* pp. 714-726.

Booij, N., Holthuijsen, L.H. and R.C. Ris, [1996] The SWAN wave model for shallow water, Proc. 25th Int. Conf. Coastal Engineering., Orlando, USA, Vol. 1, pp. 668-676.

Brown, J.M., and A.G. Davies [2009]. "Methods for medium-term prediction of the net sediment transport by waves and currents in complex coastal regions", *Continental Shelf Research*, Vol. 29, pp. 1502-1514.

Collins, J.I. [1972] Prediction of shallow water spectra, J. Geophys. Res., 77, No. 15, pp.2693-2707

Cruz J., Sykes R., Siddorn P. and Eatock Taylor R. [2009] Wave Farm Design: Preliminary Studies on the influences of Wave Climate, Array Layout and Farm Control, In: Proceedings of 8th European Wave and Tidal Energy Conference, Uppsala, Sweden

Demirbilek Z. and Vincent L. [2002] Water waves mechanics. In: Coastal Engineering Manual, Part II, Hydrodynamics, Chapter II-1, Washington, DC

EDF R&D [2010] "TOMAWAC: Software for sea state modelling on unstructured grids over oceans and coastal seas".

Esposito P. [1981] "Résolution bidimensionnelle des équations de transport par la méthode des caractéristiques". *Note EDF-DER-LNH HE-41/81.16.*

Eldeberky, Y. and J.A. Battjes. [1996]: Spectral modelling of wave breaking: Application to Boussinesq equations, J. Geophys. Res., 101, No. C1, pp. 1253-1264

Evans, D.V., [1980] Some analytical results for two and three dimensional wave-energy absorbers, in Power from Sea Waves, B. Count, Editor: Edinburgh, UK. p. 213-249.

Falnes, J., [2002] Ocean Waves and Oscillating Systems. Cambridge University Press.

Fitzgerald C. and Thomas G.[2007] A preliminary study on the optimal formation of an array of wave power devices. In: Proceedings of 7th European Wave and Tidal Energy Conference, Porto, Portugal

Fitzgerald C. [2007] Optimal configurations of arrays of wave-power devices. MSc Thesis. University College Cork

Folley, M. and Whittaker, T. [2009]. The effect of sub-optimal control and the spectral wave climate on the performance of wave energy converter arrays, Applied Ocean Research 31:260-266

Folley, M. [2010] Issues with physical modelling wave energy converter arrays, Internal report QUB MF 101112-01a

Folley, M and T. Whittaker, [2010]. "Spectral modelling of wave energy converters", *Coastal Engineering*, Vol. 57, pp 892-897.

Folley M. and Whittaker T [2011] Validating a spectral-domain model of an OWC using physical model data, *Abstract submitted to 9th* European Wave and Tidal Energy Conference, Southampton, UK

Hasselmann, K. and Collins, J.I., [1968] Spectral dissipation of finite-depth gravity waves due to turbulent bottom friction. Journal of Marine Research, 26(1): 1-12.

Hasselman, K [1974]. "On the spectral dissipation of ocean waves due to white capping", *Boundary Layer Meteorology*, Vol. 6, pp 107-127.

Hasselmann, S., K. Hasselmann, J.H. Allender and T.P. Barnett, 1985: Computations and parameterizations of the nonlinear energy transfer in a gravity wave spectrum. Part II: Parameterizations of the nonlinear transfer for application in wave models, J. Phys. Oceanogr., Vol. 15,pp. 1378-1391

Holthuijsen, L.H., Herman, A. and Booij, N. [2003] Phase-decoupled refraction/diffraction for spectral wave models, Coastal Engineering, Vol. 49, pp. 291-305

Holthuijsen, L.H., [2007] Waves in Oceanic and Coastal Waters. Cambridge University Press, Cambridge.

Iglesias, G., M. López, R. Carballo, A. Castro, J.A. Fraguela and P. Frigaard [2009] "Wave energy potential in Galicia (NW Spain)", *Renewable Energy,* Vol. 34, pp. 2323-2333

Komen, G.J. et al., [1994] Dynamics and Modelling of Ocean Waves. Cambridge Univ. Press, Cambridge, UK.

McIver P. [1994] Some hydrodynamic aspects of arrays of wave-energy devices. Applied Ocean Research 16(2):61-69

Madsen, O.S., Y.-K. Poon and H.C. Graber [1988] Spectral wave attenuation by bottom friction: Theory, Proc. 21th Int. Conf. Coastal Engineering, ASCE, 492-504

Mavrakos S. A. and McIver P. [1997] Comparison of methods for computing hydrodynamic characteristics of arrays of wave power devices. Applied Ocean Research 19(5-6):283-291

Michel W. H. [1999] Sea spectra revisited. Marine Technology 36(4):211-227

Miles, J.W. [1957] On the generation of surface waves by shear flows. J. Fluid Mech., Vol. 3, pp. 185 –204

Millar, D.L., H.C.M. Smith, and D.E. Reeve, [2006] Modelling analysis of the sensitivity of shoreline change to a wave farm, Ocean Engineering, Vol. 34, pp.884-901.

Newman, J.N., [1977] Marine Hydrodynamics. MIT Press, Cambridge, Massachusetts, USA

Phillips, O.M., [1957] On the generation of waves by turbulent wind. J. Fluid Mech., Vol. 2, pp. 417–445

Simon M. J. [1982] Multiples scattering in arrays of axisymmetric wave-energy devices - A matrix method using a plane-wave approximation. Journal of Fluid Mechanics 120:1-25

SWAMP Group: J.H. Allender, T.P. Barnett, L. Bertotti, J. Bruinsma, V.J. Cardone, L. Cavaleri, J. Ephraums, B. Golding, A. Greenwood, J. Guddal, H. Günther, K. Hasselmann, S. Hasselmann, P. Joseph, S. Kawai, G.J. Komen, L. Lawson, H. Linne, R.B. Long, M. Lybanon, E. Maeland, W. Rosenthal, Y. Toba, T. Uji and W.J.P. de Voogt, [1985] "Sea Wave Modelling Project (SWAMP), An inter-comparison study of wind wave prediction models, Part 1: Principal results and conclusions, in Ocean Wave Modelling". Plenum Press, 256p.

Thomas G. and Evans D. V. [1981] Arrays of three-dimensional wave-energy absorbers. Journal of Fluid Mechanics 108:67-88

Tolman, H.L. [2002] "Alleviating the Garden Sprinkler Effect in wind wave models", *Ocean Modelling*, Vol.4, pp. 269-289.

Tucker, M.J. and Pitt, E.G., [2001] Waves in Ocean Engineering, Elsevier, Oxford, UK

Venugopal V. and Smith G., [2009] Wave climate investigation for an array of wave power devices, In: 7th European Wave and Tidal Energy Conference, Aporto, Portugal.

WAMDI Group, [1988] The WAM model - a third generation ocean wave prediction model. Journal of Physical Oceanography, 18: 1775-1810.

Waters, R., J. Engström, J. Isberg and M. Leijon [2009] "Wave climate off the Swedish west coast", *Renewable Energy,* Vol. 34, pp. 1600-1606