

---

# **Guidance for Numerical Modelling in Wave and Tidal Energy**

---

*Mathew B. R. Topper*



**The University of Edinburgh**

*Revision: 729*

March 2010

---

# Abstract

---

Numerical modelling within the marine renewable energy community is in its infancy compared to more mature engineering disciplines such as aeronautics. Additionally, established software from other fields is often applied to marine energy challenges. These issues can manifest several difficulties such as conflicts in language and notation (semantics) and deficiencies in the approximations or applications of the models themselves. Numerical modelling has the potential to provide key cost reductions to the marine renewable energy community by replacing expensive experiments and prototypes with cheaper, reliable numerical simulations. Within this document, approximation, development and analysis of numerical models are detailed. Emphasis is placed on appropriateness of approximation, quality assurance of development and the rigour of the verification and validation process in order to ensure that results from numerical models, both developed and applied, are as efficient, reliable and transparent as possible.

---

# Acknowledgements

---

A number of my colleagues within the Institute for Energy Systems at The University of Edinburgh have helped me form the ideas and language contained within this document. Thanks go to my supervisor, Dr. David Ingram for being clear, yet flexible about the outcome of the work. I'd also like to thank Dr. Scott Couch, Dr. Ally Price and Dr. Gareth Gretton for their technical input alongside Mr. William Edwards. In addition, I would like to thank Dr. Greg Payne and Dr. David Forehand for reviewing the draft report. Finally, I'd like to thank Lucy Cradden for patiently reading and correcting the document throughout its development.

— \* —

“The purpose of words is to convey ideas. When the ideas are grasped, the words are forgotten.”  
(Chuang Tzu)

---

# Contents

---

<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Figures and Tables</b>	<b>vi</b>
<b>Nomenclature</b>	<b>vii</b>
<b>1 Numerical Modelling</b>	<b>1</b>
1.1 Standardisation . . . . .	1
1.2 Physical Insight . . . . .	2
1.3 Report Structure . . . . .	3
<b>I MODEL DESIGN</b>	<b>4</b>
<b>2 Data</b>	<b>5</b>
2.1 Introduction . . . . .	5
2.2 Parameterised and Field Data . . . . .	5
2.2.1 Parametric Data . . . . .	6
2.2.2 Field Data . . . . .	6
2.3 List of Variables . . . . .	7
2.3.1 Parametric Variables . . . . .	7
2.3.2 Field Variables . . . . .	9
<b>3 Approximation</b>	<b>11</b>
3.1 Introduction . . . . .	11
3.2 Classic Non-Dimensional Numbers . . . . .	12
3.3 Temporal Approximation . . . . .	13
3.4 Spatial Approximation . . . . .	15
3.5 Dynamic Approximation . . . . .	17
3.6 Free Surface Approximation . . . . .	18
3.7 Comparison of Methods . . . . .	18
<b>II IMPLEMENTATION</b>	<b>23</b>
<b>4 Development</b>	<b>24</b>

---

4.1	Introduction . . . . .	24
4.2	Revision Control . . . . .	25
4.3	Code Standards . . . . .	26
4.4	Boundary Conditions . . . . .	27
4.4.1	Characteristics of the Shallow Water Equations . . . . .	29
<b>5</b>	<b>Testing</b>	<b>32</b>
5.1	Introduction . . . . .	32
5.2	Verification . . . . .	32
5.2.1	Verifying the Order of Convergence . . . . .	33
5.3	Numerical Error Estimation . . . . .	35
5.3.1	Richardson's Extrapolation . . . . .	35
5.3.2	Grid Convergence Index . . . . .	36
5.4	Validation . . . . .	37
5.4.1	The Validation Experiment . . . . .	38
<b>6</b>	<b>Prediction</b>	<b>40</b>
6.1	Introduction . . . . .	40
6.2	Phenomena Identification and Ranking Tables . . . . .	41
6.3	Validation Metrics . . . . .	43
6.4	Calibration Under Uncertainty . . . . .	46
<b>7</b>	<b>Challenges and Further Work</b>	<b>47</b>
	<b>Glossary</b>	<b>49</b>
	<b>Bibliography</b>	<b>57</b>

---

# Figures and Tables

---

## Figures

1.1	Influence Diagram leading to Physical Insight . . . . .	2
3.1	Temporal Flowchart . . . . .	14
3.2	Dimensional Flowchart . . . . .	16
3.3	Dynamic Approximation Flowchart . . . . .	19
3.4	Free Surface Approximation Flowchart . . . . .	20
4.1	Sample FORTRAN95 code with documentation. . . . .	28
4.2	Characteristics of 1-d Shallow Water Equations in $x$ . . . . .	31
4.3	Characteristics of 1-d Shallow Water Equations in $y$ . . . . .	31
6.1	PIRT production process from Pilch <i>et al.</i> [15] . . . . .	42
6.2	Example PIRT from Bayarri <i>et al.</i> [5] . . . . .	42
6.3	Quantitative comparison without errors or uncertainty. . . . .	43
6.4	Quantitative comparison including acknowledged errors. . . . .	44
6.5	Direct statistical comparison incorporating error and uncertainty. . . . .	45

---

## Tables

3.1	Comparison of Numerical Methods . . . . .	22
-----	---	----

---

# Nomenclature

---

$A$	Reference area.
$A_1$	Actual fractional error.
$C_E$	Asymptotic range constant.
$C_L$	Lift coefficient for a wing.
$C_a$	The added mass coefficient for wave-body interaction.
$C_d$	Drag coefficient.
$D$	Spatial dimension of simulation.
$E$	Error.
$E_1$	Estimated fractional error.
$F$	Generic force.
$F_s$	Factor of safety.
$N$	Number of grid points.
$P$	Order of convergence.
$Q$	Method of manufactured solutions source term.
$S_b$	Damping stiffness coefficient matrix.
$V$	Reference velocity.
$\dot{x}$	Non-dimensional form of $x$ .
$\varepsilon$	Relative error.
$\eta$	Free surface elevation.
$\mu$	Dynamic viscosity.
$\nu$	Kinematic viscosity.
$\omega$	Characteristic frequency.
$\phi$	Velocity potential.
$\rho$	Fluid density.
$a$	Acceleration.
$b$	Damping coefficient.
$c$	Chord length of an aerofoil section.
$c_L$	Lift coefficient for a aerofoil section.
$d$	Water depth.
$f, j$	Generic functions.
$f_0$	Reference body force.
$f_{\text{exact}}$	Exact solution.
$g$	Gravity.

---

$h$	Grid spacing.
$m$	Mass.
$m_{\text{added}}$	Added mass coefficient for body in ideal fluid.
$p_0$	Reference pressure.
$r$	Grid refinement ratio.
$s$	Displacement.
$u, v, w, \mathbf{u}$	Velocity.
GCI	Grid Convergence Index.



---

# Numerical Modelling

---

## 1.1 Standardisation

This document presents a process for undertaking the numerical modelling of fluid phenomena relating to wave and tidal energy extraction. Numerical modelling of fluids is a mature scientific field yet it still remains a fiercely expanding discipline. Thus, the necessity to provide a ‘standard’ as to how to partake in numerical modelling for wave and tidal energy extraction may seem counter-intuitive. In fact, because such a wealth of knowledge exists for numerical modelling of fluids this can give rise to differences in approach across academic institutions and even within research institutes themselves. Such differences can highlight themselves in:

**Data:** It is known that (particularly in wave body interactions) the names of certain parameters infer different meanings dependant on the physics for which they were derived. This can cause confusion if two researchers were to discuss a problem using these identically named parameters but arrived at from different approaches. This document seeks to encourage the use of clear definitions when using such parameters.

**Approximations:** It may be the case that a researcher with a clear understanding of a particular numerical method may be biased towards it. The use of any method can be bogus, however, if the approximations that led to its inception do not represent the pertinent physics. This document seeks to encourage an ‘approximation first’ approach to model selection.

**Implementation:** SuperGen II’s ‘workstream one’ sought to initiate a database of numerical and experimental results that can be called upon to gain insight into physical phenomena using credible numerical models. Without a standardised methodology for providing the numerical results and related validation experiments to the database, confidence in its contents will be reduced.

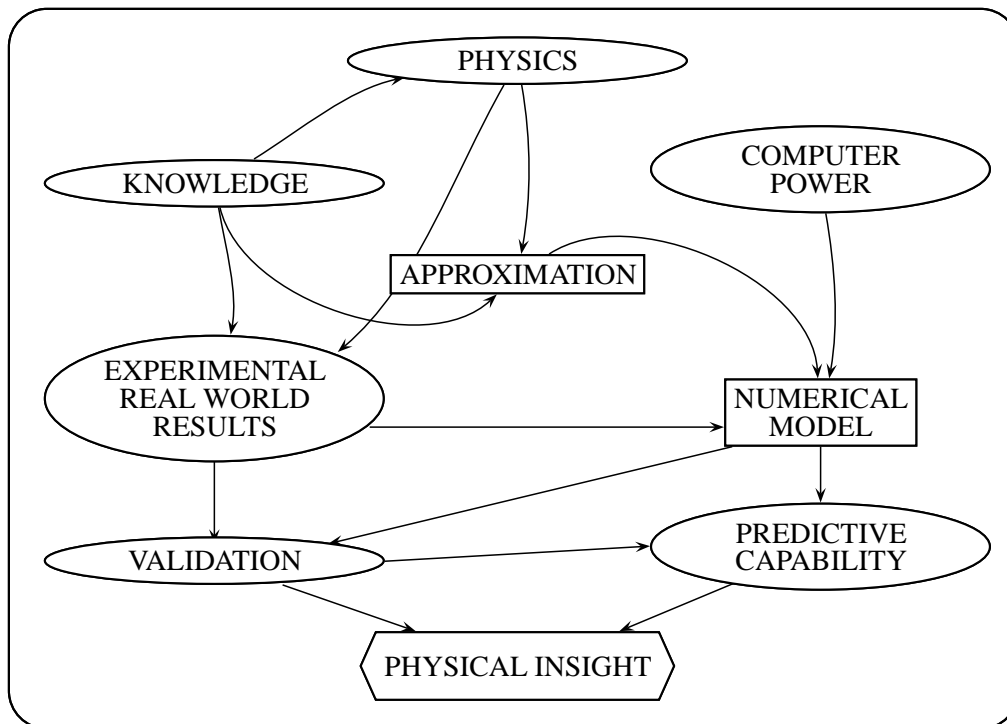
The document is structured using a ‘Why?’, ‘Which?’ and ‘How?’ approach. The next section will discuss the reasons for choosing to undertake numerical modelling of a physical phenomenon and describe the value of the results to those that undertake and commission it.

## 1.2 Physical Insight

Why undertake the numerical modelling of a physical phenomenon? Such endeavour may appear unnecessary when the physics encompassed in the design of a numerical model can be reproduced in a laboratory. In fact, although a strictly experimental approach may seem ideal it is often impractical or impossible to reproduce all of the phenomena embodied in a real life problem. In addition, the costs of such tests may be insurmountable, particularly for an infant technology such as tidal or wave energy conversion. Furthermore, a numerical simulation may provide (even in its failure) greater understanding of the mechanisms that generate the observed phenomena. To summarise, numerical modelling should be undertaken when:

- Understanding of the mechanisms of an observed physical phenomena is sought,
- Predicting physical phenomena that have yet to be observed,
- There is a desire to reduce risk.

From a scientific perspective the first two items ignite the most interest and clearly from a commercial, political and safety perspective reduction of risk is key. Reducing risk is interdependent on developing a solid understanding of all the physics that will effect a developers device or the energy resource and the subsequent ability to predict the impacts of varying the said physics. Thus, all three reasons are complimentary.



**Figure 1.1:** Influence Diagram leading to Physical Insight

Figure 1.1 shows an influence diagram that leads from the studied physics to the ability to infer insight about them. Key to the entire process is the current state of knowledge with regards to the real life problem and the encompassed fluid physics. This knowledge impacts upon evaluating which physics is the most significant, choosing an appropriate approximation, and the state of validation experiments. Once an approximation is decided upon, a numerical model can be designed taking into account any existing experimental results available and the disposable computing power. Verification and then validation of the model against the experimental results provides confidence that a model has the predictive capability required to gather physical insight. This insight may come from the validation process itself by identifying the key parameters that effect the fluid motion or it may be post-validation whence the model is confidently applied to a more complicated physical problem. This process may also lead to the discovery of new physical phenomena that can be investigated further.

An important feature to note about figure 1.1 is that the computing power available does not influence the approximation. This is key to assuring that the approximation chosen is representative of the physics being tested.

### 1.3 Report Structure

Having described the reasons for carrying out a numerical modelling approach, the remainder of this documents describes the best practise for designing and testing a numerical model to ensure that its outcomes can be utilised with a significant degree of confidence.

Part I of this document describes the various designations of data that are used to describe and quantify physical fluid phenomena. Subsequently, a decision criteria is outlined in order to aid the selection of an appropriate approximation to the full equations of motion. Part II overviews the best practice for developing and testing a numerical model for a chosen approximation. It then goes on to describe how the testing process informs the credibility and predictive capability of the developed numerical model.

For alternative reading regarding best practice in marine computational fluid dynamics and verification and validation see [11] and [3].

## **PART I**

### **MODEL DESIGN**

## 2.1 Introduction

Understanding the classes of data used to describe and quantify the results of numerical modelling of physical phenomena is critically important. This will aid the design of a numerical method and will, in turn, clarify the quantity of data that must be known beforehand in order to allow a robust description of the phenomena in question. The following chapter will provide a classification of data which serves as the common prerequisite to many numerical methods.

## 2.2 Parameterised and Field Data

An important consideration when designing a numerical method is the form in which the results will be presented. This choice is highly dependant on the phenomena in question but it can be biased by other factors. Often, there is some experimental data that a model should seek to match. Sometimes, however, experiential bias can produce output that may be of little value to describing the physics and which could implicitly lead to the use of an inappropriate numerical method. Starting with a clear understanding of the character of the data that is required aids in the efficient selection of numerical solution.

Two clear distinctions can be made when describing the data produced by numerical methods. The form of this data may be common *parametrised* outputs, for example, lift and drag coefficients or it may be in the form of *field data* which is useful, for example, when visualising the velocities of a tidal stream as it rounds a headland. For choosing which data form will provide the greatest insight, it helps to have a physical understanding of the quantities described by each data type.

### 2.2.1 Parametric Data

Parametric data describes one unique physical attribute. In wave-body interactions these could be the hydrodynamic or hydrostatic variables; in aeronautics these could be lift or drag coefficients. Parametric data is often well understood and allows tidy comparisons between different solutions. Parametrised data is often presented outside of the time domain and therefore, a steady, unchanging result is assumed over the range of variables for which it is defined. Examples of this are the coefficients of lift and drag over a range of angles of attack for an aerofoil or damping force on a floating body over a range of wave frequencies. It may be inappropriate, then, to describe time dependant phenomena in the form of parametrised data. For instance, once a field domain solution is produced, the necessity of producing hydrodynamic parameters to predict the dynamics are negated by the fact that the solution is naturally set in the time domain. It is, however, foolish to totally disregard parametrised data as it can provide excellent sanity tests for complicated numerical methods and may be required in order to compare to existing experimental data. Most importantly, parametrised data is often the best way to communicate the results to the wider community in an understandable format.

### 2.2.2 Field Data

Field data can be much more revealing than parametrised data but this often comes at greater expense. By definition, field data should be available over the entirety of a prescribed physical domain for a continuous period of time. This suggests a greater computational burden compared to parametric solutions and also necessitates the need for information about the boundaries of the physical domain and temporal information such as how a wave will be driven into the simulation. In addition, the solutions are often in two or three dimensions and this extra detail can be difficult to compare to experimental results or other numerical simulations. When parametric data is required to match experimental results it can be extracted from the field solution for this purpose. However, the wealth of additional information that a field solution can supply is undeniable.

Three dimensional temporal flow visualisation such as animated stream lines or stream tubes can provide detailed information as to how a fluid passes through a domain and past obstacles. This detail of information is achievable only with a field solution. The temporal nature of a field solution also allows the visualisation of the reaction of bodies to extreme non-linear events that may not be possible to model when seeking a solution of parametric data alone. Also, field solutions allow for the provision of both dynamic and deforming domains like the flow of blood through a beating heart, for instance.

With all these capabilities it is important to maintain focus on the desired output. Sometimes a field solution is the only method for obtaining information about an unsteady, highly non-linear phenomenon, but it still may be the case that a particular parameter or set of parameters is sought. In such a situation it is irrelevant to produce flow visualisation output and may slow the production of results as a consequence. In such situations the domain can be constrained to the area of interest

and thus the computational effort could be reduced at the expense of far field data. In contrast, even if a field solution is sought from the outset, it is highly recommended to output parametric quantities which can be used to validate a solution by comparing the output against experimental results.

## 2.3 List of Variables

This section provides a non-exhaustive list of some of the common parameters and field data. It is hoped that this list will provide some guide as to what data is useful for specific modelling situations and concurrently aid in the choice of a numerical method. The list is roughly divided into parametric quantities that are specific to subject areas that incorporate the fluid dynamics of wave and tidal energy converters, followed by a general collection of field data quantities.

### 2.3.1 Parametric Variables

#### Wave-Body Interactions

What follows is a list of terms commonly used to define parametric values of wave-body interactions. Note, often the names of these terms can be inter-changeable and real care must be taken to ensure that they are referenced to the particular form of governing equation from which they were derived. For example the properties of the *added mass coefficient* for a body in ideal fluid,  $m_{\text{added}}$  may be governed by the equation

$$F = (m + m_{\text{added}})a \quad (2.1)$$

where  $F$  is the force on the body,  $a$  is acceleration and  $m$  is the stationary mass of the body. When a free surface is included the governing equation takes the form

$$F = C_a a + bv. \quad (2.2)$$

where  $C_a$  is now the added mass coefficient,  $v$  is the body's velocity and  $b$  is the damping coefficient. Given the added mass is often assigned a predefined value, knowing from which governing equation it was derived is imperative. Further information can be found in Newman [13].

Consideration must also be given to the linear free surface assumption with which these parameters were developed. This limits the validity to waves of small steepness (amplitude over wavelength) and currents of small velocity. More in-depth and non-linear physics (described by the ensuing list of parameters) may also be derived by transforming the equations whose results are normally parameterised by wave frequency into the time domain. Exotic terms may now appear such as *memory terms* which require a summation across the entire time domain. It is academic at this moment as to whether some of the additional terms resulting from the transformations have genuine physical meaning [18, p. 601].

**Added Mass** When modelling the properties of a body in motion, the effect of the surrounding fluid can impact on the motions of the body. For instance, if the fluid is very dense then it will resist the bodies motion and the body will accelerate as if it had greater mass. The added mass coefficient seeks then to quantify the necessary gain in the bodies mass to account for the inertia of the surrounding fluid. In fact any physical process that affects the force on the body proportionally to its acceleration is included as added mass. In general it is represented by equation (2.1) however for wave-body interactions it is redefined by equation (2.2).

**Buoyancy** A hydrostatic force working opposite to the gravitational acceleration. The phenomena is caused by a pressure difference between the high and low sides of the object. It is often modelled as a function of displacement from a mean position such that

$$F_b = S_b s \quad (2.3)$$

where  $S_b$  is known as the stiffness coefficient matrix and  $s$  is the displacement.

**Damping Coefficient** The force on a body can have a component that is related to its velocity. The coefficient of this force is known as the damping coefficient as it often reduces or damps the amplitude of the motion of the body. It may be written as

$$F_d = b v \quad (2.4)$$

where  $b$  is the damping coefficient. In wave body interactions the damping coefficient is the result of waves radiated from the body as it moves with the incident wave. It appears as the second term in equation (2.2).

**Wave Excitation Force** The force generated on a body by incident waves. This can be split into the force from the incident wave and the waves diffracted by the body itself. For linear wave theory, it is considered to be unaffected by the motions of the body itself and is proportional to the incident wave amplitude.

### Lifting Bodies

The following parameters are the most common for lifting bodies and are often found in combination with each other and presented in tables with reference to angle of attack, such as within Abbott and Doenhoff [2]. Care must be taken to state the conditions under which the coefficients were derived as they may vary significantly over fundamental parameters such as Reynolds number. It is also likely that the presence of other bodies or a free surface could alter values of these coefficients.

**Drag Coefficient** The coefficient,  $C_d$ , in the drag force equation (2.5) represents the effect of many physical phenomena. Drag can result from skin friction, the shape of the object (or *form drag*) and drag resulting from the redirection of the airflow of an object to induce



lift also known as *induced drag*. The coefficient of drag is not a constant and can vary significantly for the same body over both Reynolds numbers and angles of attack.

**Drag Force** A force applied to a body that is strictly in the opposite direction to the oncoming flow. The equation for the drag force is given as

$$F_d = \frac{1}{2}\rho v^2 C_d A \quad (2.5)$$

where  $C_d$  is the coefficient of drag and  $\rho$  is the fluid density. Here  $A$  is a reference area and would be chosen as the square of the chord for a aerofoil section, nominally unity. It can be seen that this force is a percentage of the force induced if the flow were to come to a complete stop.

**Lift Coefficient** Non-dimensional quantity used in comparisons of different planforms of wing or wing sections. The coefficient of lift is of interest when relating it to the drag coefficient or the angle of attack of a particular aerofoil. Separate equations exist dependant on whether obtaining the coefficient of lift for the entire wing,  $C_L$ , where

$$C_L = \frac{2L}{\rho v^2 A} \quad (2.6)$$

where  $L$  is the lift generated by the wing,  $\rho$  is the fluid density and  $A$  is the total area of the wing planform. For an individual aerofoil section of the coefficient of lift,  $c_L$  is defined by

$$c_L = \frac{2L}{\rho v^2 c} \quad (2.7)$$

where  $L$  now represents the lift generated by the particular aerofoil section and  $c$  is the chord length of the section.

### 2.3.2 Field Variables

The field data variables mentioned here are but the most common. In fact, any varying quantity could be represented as field data so in a gas, for instance, density would also be an appropriate field variable. Normally, however, in wave and tidal hydrodynamics density is considered constant throughout the fluid.

It should be noted that these field variables often mean little in isolation, unlike a parametric variable that remains highly useful in isolation. Field variables are often best presented as a group as in a vector diagram or where this is impossible, parametric data should be extracted from the field data.

**Pressure** Given by force per unit area it's the pressure from surrounding fluid particles that drives fluid motion in hydrodynamics. Fluid pressure can often be split into two forms one being *hydrostatic pressure* generated by gravitational forces and the second being *hydrodynamic*

*pressure* being the pressure generated by the fluid motion itself. In certain circumstances the hydrostatic pressure is dominant and the hydrodynamic pressure can be ignored. This is known as a *hydrostatic assumption*. Importantly, body forces can be derived from the pressures on surfaces or boundaries and thus the motions of bodies within the fluid domain can be determined.

**Velocity** The most fundamental variable in a field solution is velocity and it can be represented in two forms. Firstly, the most common *Eulerian* form evaluates the velocity of a fluid at particular points in space and the velocities are written in the form

$$\begin{aligned}u &= u(x, y, z; t) \\v &= v(x, y, z; t) \\w &= w(x, y, z; t).\end{aligned}$$

Alternatively each particle's velocity can be tracked in relation to its own position at time  $t = 0$ . This is known as a *Lagrangian* flow description. In fixed grid applications the Eulerian description is applied; however, the Lagrangian description can be useful for the tracking of surface particles, for instance.

**Velocity Potential** When inviscid irrotational flows are being considered it is possible to define a velocity potential,  $\phi$  such that

$$\nabla\phi = \mathbf{u}. \tag{2.8}$$

A velocity potential is not a strict field variable as it is not normally available at all points in the domain. It is generally available on all surfaces and boundaries of the domain and common field variables can be determined analytically at any point within the domain. Such a potential reduces the spatial manifold by one and hence can reduce the computational burden.

---

# Approximation

---

## 3.1 Introduction

As seen in the previous chapter, by designating the classes of data that will be sought for the outcome of a numerical model an intrinsic understanding for the pertinent physics of the encompassed phenomena is demonstrated. This knowledge will allow the best representation of findings to peers, although understanding the form of the final outcome is insufficient for designing the model. This process will advance as a consequence of the modeller's understanding of the pertinent physics, which will instigate a process of decisions with the goal of reducing the complication of the full fluid equations. A modeller will choose to make these approximations in order to avoid a direct numerical simulation of all physical phenomena. Such a model for wave and tidal research is well beyond the limits of most computational facilities at this time. With careful consideration a number of the complications can be reduced and, provided that the chosen approximation maintains the most pertinent physics, the results generated should agree well with experimentation. If they don't, then it's likely that the approximation may have been chosen in error and this, in itself, is a useful insight.

This chapter illustrates the process for forming a numerical approximation to a physical problem in wave and tidal hydrodynamics. The first three sections relate to the three main areas of hydrodynamic modelling. These are the temporal approximation, the spatial approximation and the dynamic approximation, respectively. Finally, the concluding section of this chapter focuses on the specific problem of the free water surface. Flowcharts are included which present a hierarchy of decisions demonstrating how each approximation leads to a numerical method. The summary table 3.1 collates the outcomes in order to compare all identified numerical schemes<sup>1</sup>.

Within the flowcharts and in order to aid the selection of an appropriate approximation some parameters that define certain properties of the flow are useful. These are known as the classic non-dimensional numbers.

---

1. Numerical methods described by acronyms, such as Reynolds Averaged Navier-Stokes (RANS), are listed in the glossary of this document.

### 3.2 Classic Non-Dimensional Numbers

It is often crucial to determine the relative values of certain physical properties in a fluid flow. A great deal of information for the flow in question can be revealed by considering the importance of certain physical properties described by a series of non-dimensional parameters. These become apparent in the non-dimensionalised momentum equation for the incompressible flow of a Newtonian viscous fluid, given here in the  $x$ -direction [10]:

$$\begin{aligned} \left(\frac{\omega L}{\sqrt{V}}\right) \frac{\partial \tilde{u}}{\partial \tilde{t}} + \tilde{u} \frac{\partial \tilde{u}}{\partial \tilde{x}} + \tilde{v} \frac{\partial \tilde{u}}{\partial \tilde{y}} + \tilde{w} \frac{\partial \tilde{u}}{\partial \tilde{z}} \\ = \left(\frac{L f_0}{V^2}\right) \frac{\partial f_0}{\partial \tilde{x}} + \left(\frac{p_0}{\rho V^2}\right) \frac{\partial p_0}{\partial \tilde{x}} + \left(\frac{\mu}{\rho V L}\right) \left(\frac{\partial^2 \tilde{u}}{\partial \tilde{x}^2} + \frac{\partial^2 \tilde{u}}{\partial \tilde{y}^2} + \frac{\partial^2 \tilde{u}}{\partial \tilde{z}^2}\right). \end{aligned} \quad (3.1)$$

Here ‘ $\tilde{\cdot}$ ’ implies a non-dimensional variable,  $u, v$  and  $w$  are the velocities in the  $x, y$  and  $z$  directions respectively and  $L, V, f_0$  and  $p_0$  refer to reference length, velocity, body force and pressures respectively. Also given is  $\omega$ , the characteristic frequency, and  $\mu$ , the dynamic viscosity.

The physical importance of the quantities in parenthesis in the above equation are now described.

**Euler Number** Used to determine the importance of the reference pressure on the equations of motion and is given by

$$Eu = \frac{p_0}{\rho V^2}, \quad (3.2)$$

where  $p_0$  is the reference pressure.

**Froude Number** The Froude number describes the relationship between the body force and the inertial force in the equations of motion and is given by

$$Fr = \frac{V}{\sqrt{Lg}}, \quad (3.3)$$

where  $V$  is a reference velocity and  $g$  is the acceleration due to gravity. When choosing the body force as gravity a small Froude number would imply that the effect of gravity is of importance and thus free surface simulation should be important. It may be neglected, however, in the case of short length scales or large reference velocities.

**Reynolds Number** A highly important term in numerical modelling of fluids is given by

$$Re = \frac{VL}{\nu}, \quad (3.4)$$

where  $V$  and  $L$  are characteristic velocities and lengths and  $\nu$  is the *kinematic viscosity*. The Reynolds number allows a judgement on the importance of viscosity to the equations of motion. If the Reynolds number is high then the terms associated with viscosity may be neglected (away from surfaces) and if it is low then they may not. Another important point to note is that when scaling to examine viscosity-based effect it is important to maintain the

Reynolds number of the full scale test or the results may be invalid. Similarly in numerical analysis if a certain Reynolds number has been assumed then the test is only valid where the parameters simulated produce the said number.

**Strouhal Number** The Strouhal Number represents the importance of time dependant phenomena in the equations of motion. It is defined by

$$St = \frac{\omega L}{V} \quad (3.5)$$

where  $\omega$  is the frequency of the characteristic process such as vortex shedding. If the Strouhal number is low, say due to a low frequency of vortex shedding from a body, then it may be appropriate to neglect time dependant terms.

### 3.3 Temporal Approximation

Determining whether or not the problem to be solved has a time dependency will have a key effect on the selection of a numerical method. There are a number of clear examples where a *steady* and time independent solution can be sought. For instance, lifting solutions to aerofoils are often undertaken in the steady state as the wake can be assumed to have taken an unchanging form after a finite period of time has passed. Solutions to wave-body interactions are often considered to be steady-state as well. A single result can be extracted for a particular wave frequency. The real life interaction is not time-independent however and varies across the particular sinusoid associated with the chosen frequency. The term steady-state is used as no numerical time stepping is required to derive the motion of the body. For frequency domain results the wave steepness is often chosen to be small so that a linear approximation can be used. Some non-linear motion can also be represented by frequency domain results; however, the ability to perform such a steady state simulation is easily lost. Unrestricted body motions or strongly non-linear boundaries may well impart the use of an unsteady solution.

Flowchart 3.1 shows the decision process for forming approximation regarding the temporal nature of the simulated problem. The Strouhal number will provide an initial indication of whether a problem is steady or not, along with the observation of any obviously dynamic processes. Once an unsteady solution is sought, the next stage is to look for simplifications that can provide a pseudo steady solution as discussed above for wave body interaction. For instance, one must determine whether the wave steepness is small enough to be approximated by a linearisation and whether the motion exhibits some periodicity. Failing this, one must decide whether or not turbulence is important to the outcome of the simulation. The modeller must then choose whether to average the effect of this turbulence and, if so, it becomes important to understand the turbulent length scales in order to minimise the error from this approximation. If this *Reynolds averaging* process is not sufficient, perhaps because the turbulent motions dominate the flow, then direct

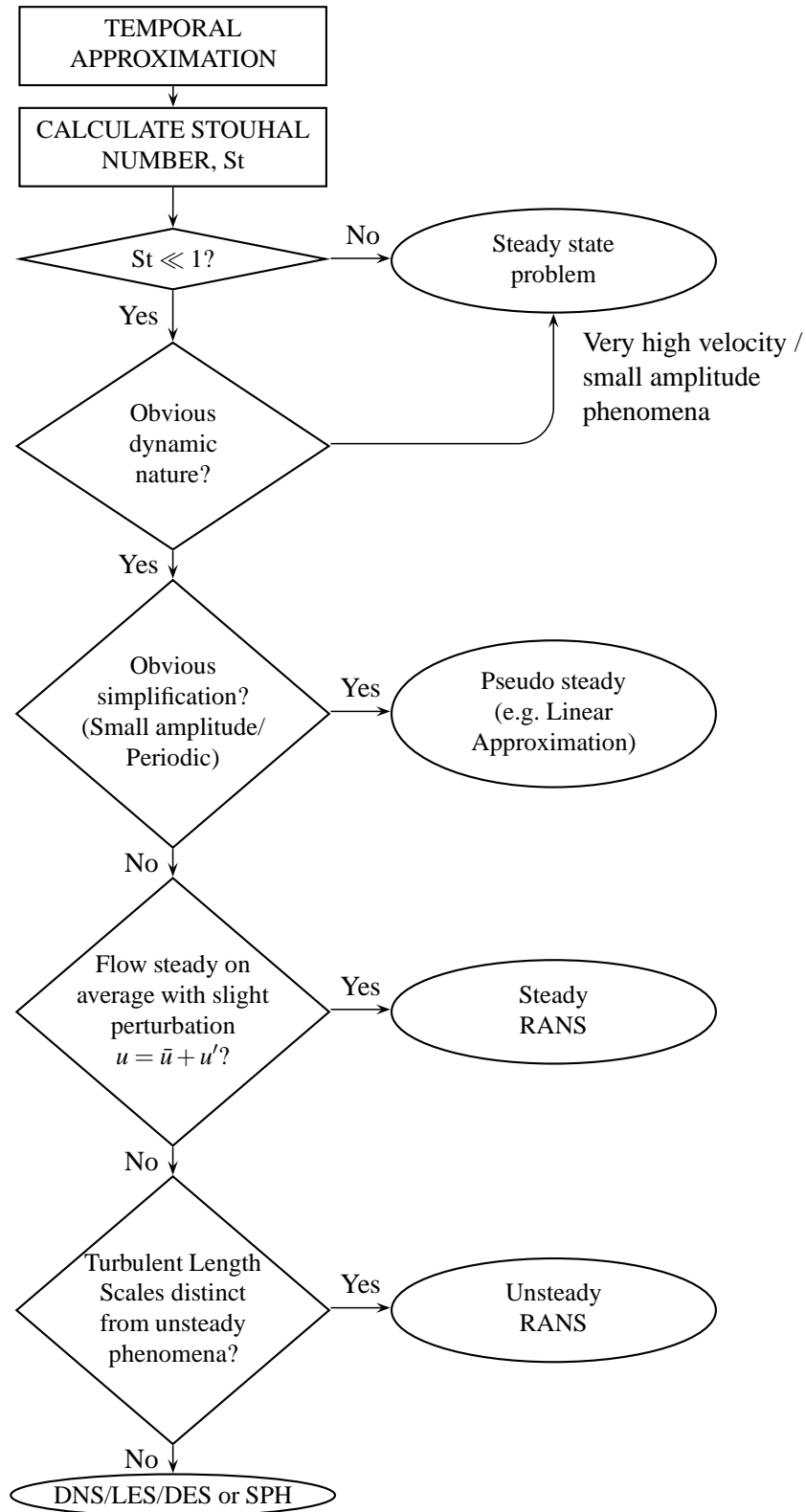


Figure 3.1: Temporal Flowchart

numerical simulation must be applied over the entire domain or for restricted regions of the domain. Alternatively, the contemporary Smoothed Particle Hydrodynamics (SPH) method can resolve turbulent structures at various scales [8].

### 3.4 Spatial Approximation

Reduction of the spatial domain has a clear advantage from a computational perspective, but under what circumstances is such an approximation viable? A number of approximations for flows with free surfaces have been designed over the years, most seeking to either restrict the vertical range of the computational domain or remove the vertical momentum equations. These approximations are viable under certain conditions, notably those of the linearised free surface boundary conditions and the shallow water approximation.

The *shallow water approximation* assumes that the characteristic length scale,  $L$ , is much greater than the depth,  $d$ . It is often used for tidal flows but care must be taken to ensure that the scales of the flow being examined represent this approximation. On the other hand, the linear approximation to the free surface does not require shallow water but does require small amplitude disturbances on the free surface. With this approximation the computational domain can be bounded above by undisturbed free surface.

A linear approximation is often used to describe the motion of a wave energy converter in “normal operation”. However, where large scale motions are concerned, the problem becomes non-linear and hence this approximation will be inaccurate. Also, the linear approximation assumes an inviscid, irrotational fluid and thus is invalid when a wave energy converter is subject to an oncoming flow as the device is likely to shed vortices that will require unsteady temporal simulation.

Approximations for the transverse direction to the oncoming flow are commonly seen in fluid dynamics. The cross flow equations are often neglected under the assumption that the flow is unchanging normally to the primary direction of motion. Strictly speaking, this is an averaging of an equation of motion and this is a more robust approach to neglecting a spatial dimension. Such an averaging should not strictly be possible for turbulent flows as turbulence is, in general, a three dimensional phenomenon.

Symmetry can also reduce the dimension of a problem. For instance, in the flow through an actuator disk the problem can be taken from three to two dimensions by assuming that the flow will be identical in all the planes through the centre of the disk that are tangent to the dominant flow direction. Hence, only a two dimensional simulation need be carried out and the three dimensional flow can be abstracted from those results. One should note that axisymmetric flows are somewhat idealised and that, once again, turbulent motions would invalidate this assumption.

Flowchart 3.2 defines the process for selecting a dimensional approximation. The vertical approximation is considered first, moving onto the horizontal approximation. If it is found that

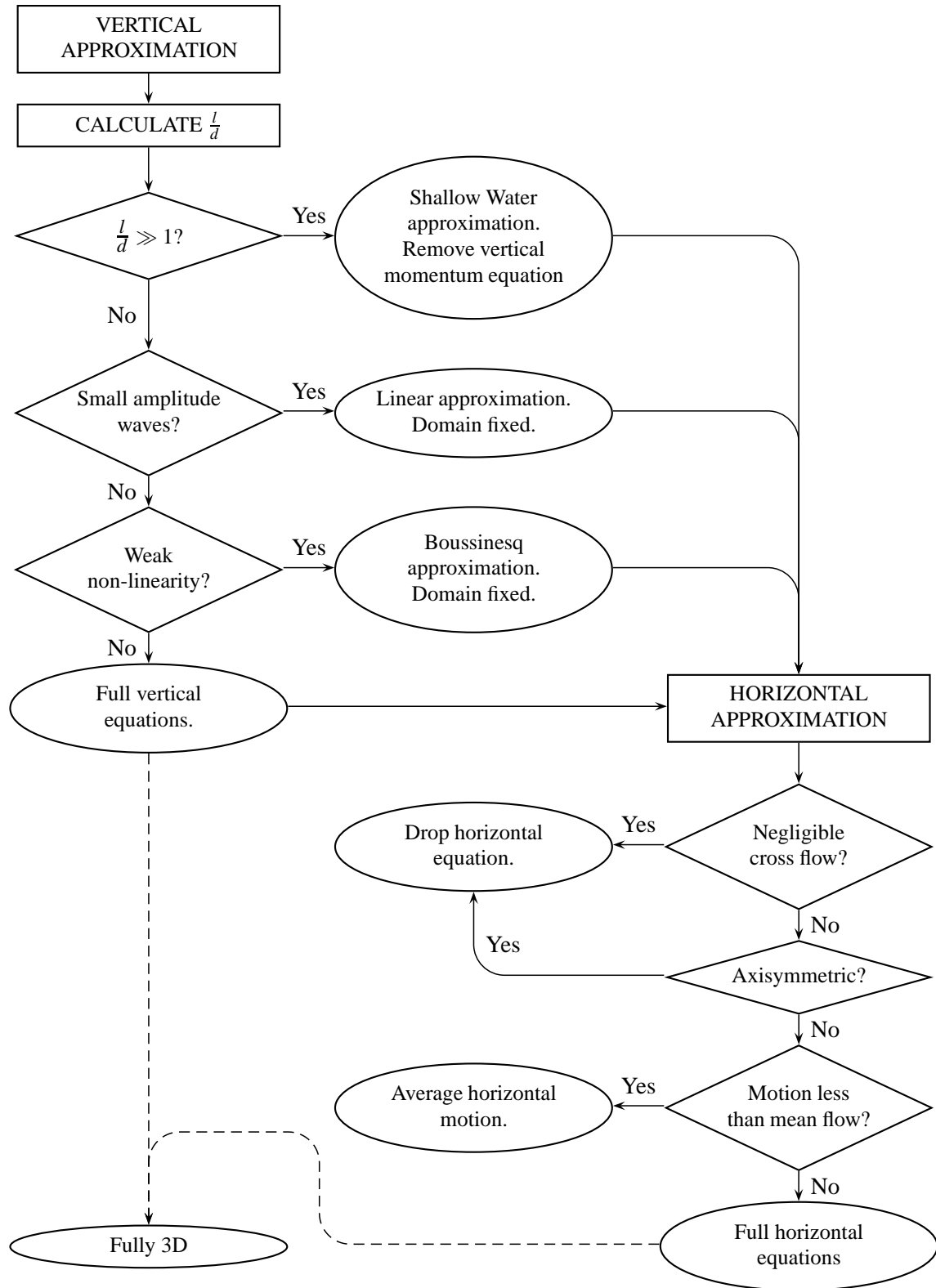


Figure 3.2: Dimensional Flowchart



no approximations can be made in either direction then full 3D equations must be solved by an appropriate method.

### 3.5 Dynamic Approximation

Making an approximation to the dynamic nature of the flow has the ability to make large reductions in the complication of the equations to be solved. The crucial quantity being approximated here is viscosity. It is important to note that the dynamic approximation could modify some of the previous approximations were it decided that viscosity plays an important role throughout the fluid, hence a modeller must not consider the approximations as “set in stone” until considered as a group. The Reynolds number,  $Re$ , is a key parameter for choosing a dynamic approximation. Its formulation gives an indication of the relative scale of viscous effects compared to inertial effects and hence their importance to the flow in question. At the extremes, if the Reynolds number is very small, then viscosity is dominant and a solution to the Stokes flow problem should be sought. If the Reynolds number is very high then viscosity can be neglected and the Euler or potential flow equations can be employed.

Unfortunately, the physics of fluids is complicated by the existence of turbulence in high Reynolds number flows. A concise definition for the nature of turbulence is challenging, but essentially it is seen as a statistical fluctuation of velocity, in all directions, about the mean. A grey area is the distinction between defined vortical structures (such as appears in the wake of a submerged object) and random statistical turbulence. Some authors like to consider turbulence by considering the motion of cascading eddies reducing in scale from larger into smaller and smaller disturbances, which includes phenomena at many length scales. Some prefer to make the distinction between the larger bathymetry-generated objects and the smaller scale eddies that form in the dissipation of that object. Whichever definition is used, the end result is a flow where turbulence can be of significance and is highly complicated to model. Turbulence is a highly important physical phenomena, as without it the large scale vortices could not dissipate their energy in high Reynolds number flows and chaotic motion would dominate all flows of interest.

To the advantage of numerical modellers, the work of Prandtl (see Schlichting [19]) showed that the impact of turbulence on a flow can be marginalised to regions of significant shear, such as at the boundary of a solid object. This is the classic boundary layer approximation that assumes outside a small boundary layer a turbulent flow may be modelled with an inviscid solution whilst a more complicated method is required within the boundary layer. This approach significantly reduces the challenge of modelling the flow away from boundaries such as in the wake of a marine current turbine. However, should there be persistent vortical structures in the fluid away from the boundaries, this approximation may not be valid. Also, the extent of the turbulent boundary layer generated by the flow over the bottom of a tidal channel may reach as far as the free surface. This

could cause severe hydrodynamic and mechanical problems for tidal devices if the turbulent length scales are similar to the characteristic length scales of the device.

Flowchart 3.3 demonstrates a decision process that can be used for forming a dynamic approximation.

### 3.6 Free Surface Approximation

The free surface does not always require a formal approximation, as this section would suggest. However, given the obvious importance of a free surface in wave and tidal hydrodynamics, an approximation to the free surface physics would be advantageous. We have already encountered some free surface approximations in section 3.4 and this section gives a more formal criteria for how those approximations are chosen with respect to the motion of the free surface.

Interestingly, an inviscid dynamic approximation can be used up to the most complicated forms of free surface approximation. However, once entrained air in a breaking wave is considered important, viscosity will become integral to the evolution of the flow and thus a more sophisticated solver is required.

Introduced here is the *boundary element method*. This is a branch of potential flow that requires the solution of its governing equations only on the domain boundaries. This effectively reduces a three dimensional problem to a two dimensional one. It also has the advantage of being Lagrangian in nature and thus surface tracking is more naturally undertaken. Unfortunately, given the potential flow assumption of inviscid flow, the range of dynamic approximation where this technique is valid is small.

Flowchart 3.4 shows the process for deciding upon a free surface approximation. Notably, the small scale transient and multi-valued non-breaking wave approximations depend on an inviscid dynamic approximation. Without this then a single phase fluid solver will be required.

### 3.7 Comparison of Methods

Given the interchangeability of a number of the approximations reached above, a table has been compiled to present the numerical methods identified with respect to each level of approximation. Table 3.1 restates the key temporal, spatial, dynamic and free surface approximations. Each numerical method is then checked to see if it is capable of representing the approximation and then whether it is appropriate to use it for that level of approximation. These are represented with ‘●’ and ‘√’ respectively. Where it is unclear if the method relates to a level of approximation there is a ‘?’.

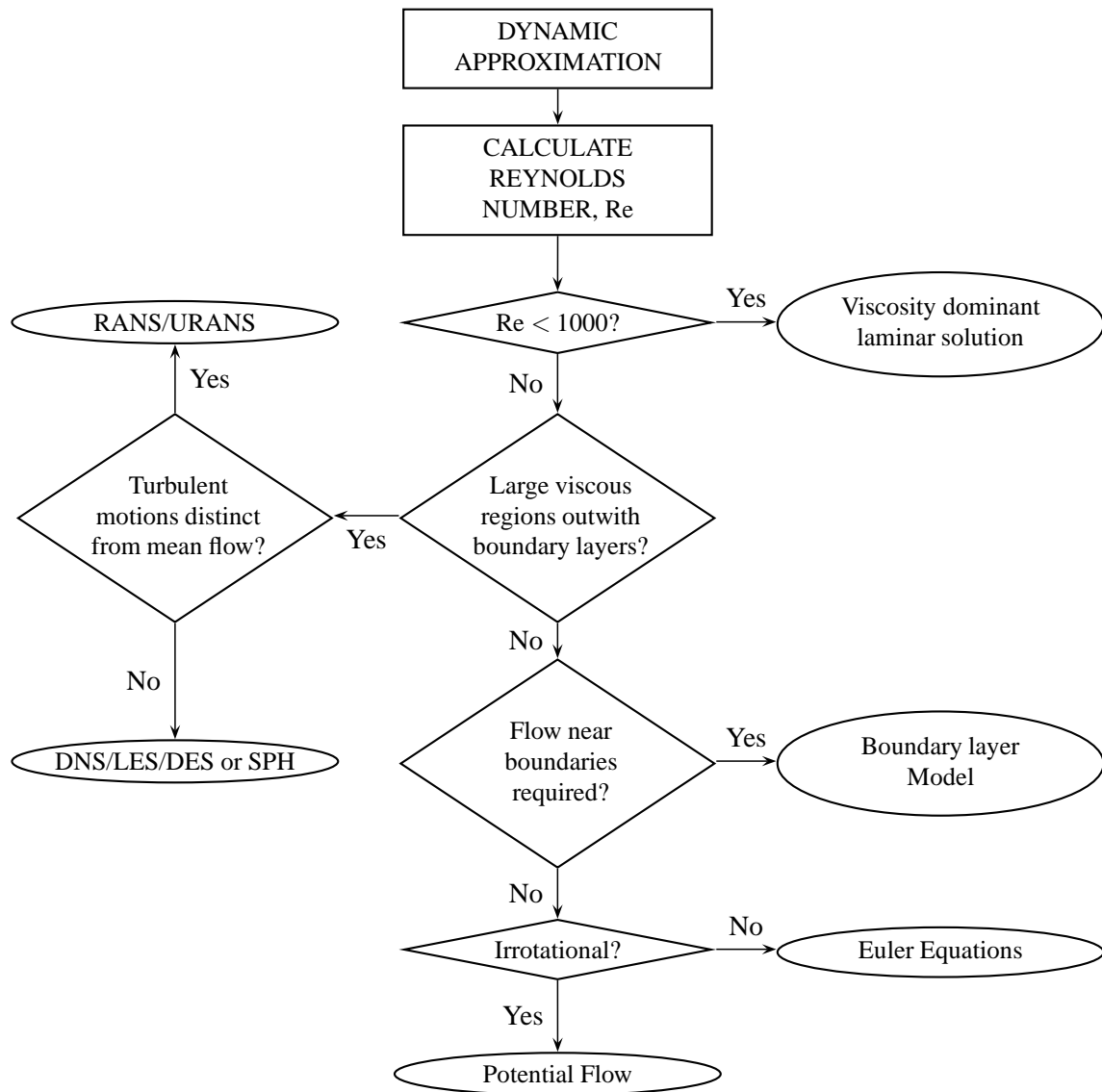
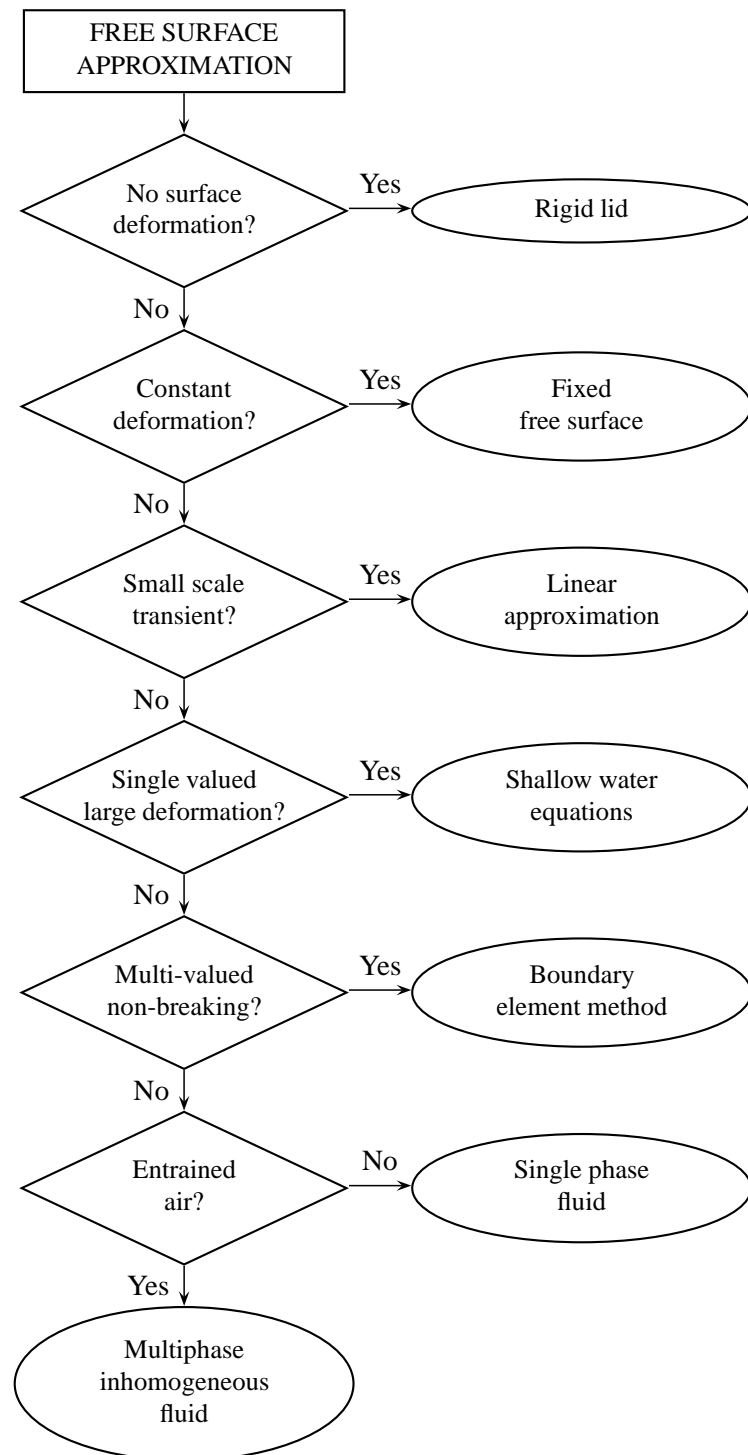


Figure 3.3: Dynamic Approximation Flowchart



**Figure 3.4:** Free Surface Approximation Flowchart

---

Note that the table indicates that the abilities of the higher level methods are more wide ranging than the more selective methods. This is certainly the case, however, with these capabilities comes a computational cost that is hard to justify if the physics can be represented by the less intensive method. Hence, it is best practise to select a method capable of representing the chosen approximation avoiding other methods with expensive latent capabilities.



## **PART II**

### **IMPLEMENTATION**

---

# Development

---

## 4.1 Introduction

Part II of this document examines the evidence gathering process required to give credibility to the results of numerical models. Ideally, the approximation chosen in part I will devise a numerical model that, together with a high level of numerical confidence, can demonstrate predictive capability. This facility is provided by matching related validation experiments to the outcomes of the numerical model. Successful validation is important to quantify predictive capability and, in contrast, failure to validate demonstrates that the approximation chosen to design the model is insufficient (assuming confidence in the accuracy of the numerical and experimental results). The process of determining whether a model is, “solving the equations right,” is known as *verification* and without taking this step little credence can be had in the integrity of a numerical model or the accuracy of the results it produces.

Before the process of verification can start, a model must be chosen that discretises the approximation developed in part I to represent the physical phenomenon to be examined. There may be commercial, open-source or in-house codes available that already achieve the desired level of approximation. A description of a number of available commercial packages for the marine environment is available in McCabe [12]. The reduction of development time when using commercial code is fortuitous, but this does not imply that the numerical error can be assumed to be small and studies should be carried out to quantify it. Should there be no existing model or should it fail to verify, then a new model must be developed - a protracted task in itself.

This chapter is concerned with highlighting elements of code *quality assurance*, (QA), that are important in the development stages of a numerical model. For reasons of clarity, the word *code* shall now be used to describe the scripts of programming language that make up the algorithms within the numerical model. Also, the numerical model shall now be referred to simply as the *model*.

It should be noted that a model is highly unlikely to undergo one development stage prior to being tested (see chapter 5), and then be assessed for predictive capability (chapter 6). It is part of the author’s experience that a model will be tested at a number of stages along the way to completion and this process will increase the body evidence for the models successful verification.



As a consequence of this non-linear and modular approach, the need for quality assurance from the developer is paramount.

## 4.2 Revision Control

To an inexperienced programmer the value of revision (or version) control may not be immediately clear. Indeed, it may be seen as an obstacle to the main aim of the programmer, that of finishing a working model. An experienced programmer, on the other hand, will know that revision control negates the risk of making a change that could cripple a model, as corrupting changes can be returned to a working version. Also, as the code becomes more complex, the management of different ideas can be confined to *branches* of the model and as more programmers join the development team, their work can be synchronised by *merging* their efforts into one central store or *repository*.

It is recommended that, at an early stage, any model being developed should be placed under revision control. At the very least, revision control will serve as some sort of backup, even if the code is not particularly ‘mature’. Certainly, once any form of testing or verification has taken place, it is important to *check-in* the successfully verified code so that (should the programmer be uncertain of the accuracy of a later version) it can be recovered to a state that is known to work with some confidence.

A consequence of maintaining a revision control system is the necessity to write logs with each check-in. This may seem like a hassle should the programmer have changed one line of code, for instance, but in a years time when it is needed to locate the particular version of the file prior to the single line change, the log explaining the differences will be invaluable. It’s not too strong to say that without the keeping proper logs, revision control will be ineffectual.

As code becomes more developed a programmer may wish to add new features. At the same time, however, they may not wish to destabilise or corrupt the accuracy of their existing model. In this case a ‘branch’ (simply a copy) of the code can be created in order to develop the new feature separately from the main (or *trunk*) code. When the feature branch is ready and tested it can be merged into the trunk and subsequently erased.

Another feature of revision control is a *snapshot* or *tag*. These are either fixed revisions of a model or a copy of the trunk at a milestone or release. These tags can be further debugged, but the expectation is that the trunk will continue to evolve whilst the tag’s feature set will be mainly unchanged. This is useful if a major feature upgrade is planned and a programmer still wishes to use the pre-feature code for problems that don’t require the new features.

Server based revision control systems, such as Subversion, have become very common as they expand the use of revision control to development teams that can *check-out* and develop a model independently and then merge their changes back into the common store. This can require some

management, should two programmers be working on the same section of code at the same time, but it offers an invaluable feature for open source projects where the development teams are somewhat disparate.

When a model has become more mature, the developer may wish to consider implementing *regression testing*. Regression testing requires a test suit that can be implemented after every revision to the numerical model. This will stop a coding error being transmitted through many revisions before (hopefully) being found. The difficulty lies in designing a test suit of a size that will not make the revision control process too time consuming.

### 4.3 Code Standards

In early central and personal computing, hardware memory was at a premium. Even with today's computers, researchers often find they are hampered by a lack of memory for ever more expansive simulations. In present times, however, the restrictions on memory no longer apply to source code. The established trend is for better conforming, understandable and portable code to be developed. In order to allow portability between users and systems, documents known as *coding standards* have become commonplace.

Coding standards normally are produced for a particular institution in relation to a particular programming language. The concept behind them is that any code submitted to the institution (NASA, MetOffice) will be produced in such a way that it could be distributed to any of their members who will be able to use and develop the code further. To achieve this, rules about style, documentation and other code specific features are either enforced or recommended. To illustrate these points, let us consider the style and in-situ documentation of a numerical model.

Coding style is an application of basic common sense. This includes giving variables names that relate to their usage, initialising variables in a clearly defined area of the code, and indenting code blocks such as loops or logical statements. Even using whitespace and blank lines to make code more readable may seem elementary, but for programmers of an older generation this may have been shunned. Often, tab characters are not recommended due to issues with porting between systems, however, a number of text-editors are capable of converting tabs to whitespace should this be necessary.

A lack of any in-code documentation should be seen as a gross oversight by the developer (unless the code has significant external documentation to describe its source). Even the developer themselves, when stepping away from their code for a period of time, may find it impossible to reform the ideas behind their algorithms should what face them be bare FORTRAN, say, with no prose to reference against it.

In-code documentation can also aid the revision control process by making a more 'local' identification of the changes to a file. This is particularly true of code that has many subroutines or

functions (smaller programming blocks) contained in one file. While the revision control log can explain the main purpose of the change, the log held within the in-code documentation can more precisely locate the change and describe it in more detail. It is good practice for code to contain a written log of changes, referencing who made them and at which time.

Most importantly, in-code documentation should explain the overall purpose as well as the inputs and outputs of the main code and/or any subroutines. Without this information it will be near impossible for any person other than the developer to operate the model. Describing the purpose of code sections can be achieved via referencing an external document or described explicitly at the top of the code segment. Each input and output variable should be described at introduction and, ideally, the purpose of all the local variables should be described too. Some details of how the algorithm is designed is also useful, but this may be more appropriately written ‘in-situ’ within the code. Figure 4.1 demonstrates some of these ideas using the FORTRAN95 programming language.

Following a code standard should negate a number of common errors that regularly occur in model development. For example, code standards will emphasise the need to declare global constants once and then reference that one declaration rather than reproducing it in each separated code block. This will stop an unsuspecting user altering one constant unknowing that they should have altered at other locations in order to achieve the result they desired.

This document is not defining or declaring a particular code standard to follow, but it is recommending that modellers attempt to follow a reasonable subset of the rules and recommendation contained within the standards for the particular programming language. Examples of such standards exist on the wikipedia “Coding conventions” page, however no FORTRAN standard is referenced there. For FORTRAN standards please see the standards published by the MetOffice [4] or the NOAA [6].

## 4.4 Boundary Conditions

Uniqueness and stability of a numerical model is often determined by the application of boundary and initial conditions. Too little information can lead to an infinite number of solutions and too much can lead to no solution at all. Determining the correct type and number of boundary conditions is an important step in the development of a numerical model.

In general, techniques exist to classify a set of partial differential equations (PDEs) (such as the Navier-Stokes equations) in relation to their required boundary conditions. These classifications are known as hyperbolic, parabolic, elliptic, or hybrid systems. Physically, hyperbolic systems represent the advection phenomena, parabolic systems admit solutions corresponding to damped waves and, finally, elliptic systems represent diffusion phenomena [9]. A solution containing a mixture of these phenomena is known as a hybrid.

```

PURE FUNCTION Source_Influence (collocation, input_panel, node_number, quadorder)
!
! Reference:
!   The basis for this algorithm has been constructed from H. Xu's PhD Thesis "Numerical
!   Study of Fully Nonlinear Water Waves in Three Dimensions" Massachusetts Institute of
!   Technology, February 1992. p63 equation 4.12
!
! Purpose:
!   To calculate the source influence associated with a collocation point
!   and the source distribution of a far field generic element.
!   The program takes in a collocation point a far field panel, a given order of quadrature
!   and returns the result of the source influence. Equation 4.12 is solved for each given
!   root depending on the level of quadrature and then a weighting is applied.
!
! Record of Revisions:
!   Date           Programmer           Description of change
!   ===           =====           =====
!   25/08/06       Mathew Topper           Original Code
!   28/08/06                                     Added and tested Nm, Nms and Nmt functions.
!                                       They should probably be in their own module!
!   29/08/06                                     Added Legendre subroutine but it needs fixing!
!   01/09/06                                     Changed the way the arrays run so vectors are stored
!                                       across the first subscript not the second.
!                                       Added Static_Data Module to hold static data (gravity etc)
!   08/09/06                                     Fixed Jacobian. Realised it's not dependant on s and t
!                                       and therefore should be calculated just once.
!   11/09/06                                     Realised Jacobian thingy was total nonsense! In fact it only works
!                                       that way for a constant Jacobian. Anyhoo, traced the symetry bug down
!                                       to problems in Nm function. Also discovered mistake in Nmt function.
!                                       So much for my testing. Oops! Need to test again an analytical solution
!                                       now.
!   6/11/06                                       Have made a series of replacements to encompass the new data structures.
!                                       I need to make it work for pointers still, however. It now works with
!                                       pointers.
!                                       Removed the outer node loop so that this works out the source influence
!                                       rather than the whole integral. I think.
!                                       Have deleted the outer loops and called this Source_Influence as it now
!                                       works out the influence parts of the integral
!   16/11/06                                     Put the gl_table_bounds_picker as an array is the Static_Data mod to stop
!                                       all the calls to it.
!                                       Now using fixed arrays for the results of nm, nms and nmt which need to be
!                                       set up at the beginning of the program.
!                                       Now recording values for J/r for quadrature of order 2 to save
!                                       recalculating for each of the nine nodes of one panel.
!                                       Brought gl_table bounds and coefficient part of gl_table local to save time.
!                                       Currently running at ~1.3 seconds for 5 million panels.
!   21/11/06                                     Function is now PURE. Also running faster due to parallel computation.
!   31/1/07                                       Changed the srtructure to upate the stored j's and r's at different times
!                                       as the stored j's on the panels only need resetting at the time step, but the
!                                       r's are recalculated for each collocation point.
!
IMPLICIT NONE

! Data dictionary: declare calling parameter types & definitions
TYPE(node), POINTER      :: collocation           ! Collocation point
TYPE(panel), POINTER    :: input_panel           ! Inflence over this panel on collocation point
INTEGER, INTENT(IN)     :: node_number           ! The node number of input_panel at which to calculate the influence
INTEGER, INTENT(IN)     :: quadorder            ! Order of quadrature to use
REAL                    :: Source_Influence      ! The result of the intgration

! Data dictionary: local variables
INTEGER                 :: iquadloop, jquadloop, inmloop ! loop dummy variables
REAL                   :: nmpart                 ! The Nm(s,t) part of integral equation 4.12
REAL                   :: xs, ys, zs, xt, yt, zt  ! x y and z derivatives in (s,t) space
REAL                   :: jpart, j1, j2, j3       ! The Jacobian of integral equation 4.12 and it's component parts
REAL                   :: rpart, r1, r2, r3       ! The distance, r(s,t) in equation 4.12 and it's component parts
REAL                   :: tempnm                 ! This is used so that call are only made to Nm functions once.
INTEGER                 :: arraynum              ! This is used to number the values for the joverr arrays in the panels
REAL, DIMENSION(4)     :: fastnm                ! This is the array that holds the values of nm for quadrature order 2

!Initialisation of variables
arraynum = 1

```

Figure 4.1: Sample FORTRAN95 code with documentation.

Classification of a set of governing equations is determined by calculating their characteristic normals. Once these are known, characteristic surfaces can be defined. These characteristic surfaces demonstrate how information travels within the domain of interest. This information then provides a visual interpretation of the necessary boundary conditions for solving the system of equations. Techniques for finding characteristic normals and forming characteristic surfaces can be found in Hirsch [9, chap. 3].

#### 4.4.1 Characteristics of the Shallow Water Equations

As an example, consider the characteristics for the shallow water equations in one spatial dimension, but still allowing velocity normal to the direction of the flow. This is a more concise reproduction of the analysis given in Topper [20].

The shallow water equations for inviscid, incompressible flow are given by

$$u_t + uu_x + g\eta_x = 0 \quad (4.1)$$

$$v_t + uv_x = 0 \quad (4.2)$$

$$\eta_t + (\eta + d)u_x + u\eta_x = 0 \quad (4.3)$$

where  $u$  is the velocity in the  $x$ -direction,  $v$  is the velocity in the  $y$ -direction,  $\eta$  is the free surface elevation,  $g$  is gravity and  $d$  is the undisturbed water depth for a channel with a flat bottom.

The characteristics of these equations are then given by

$$\lambda = u \pm \sqrt{g(\eta + d)} \quad (4.4)$$

$$\lambda = u. \quad (4.5)$$

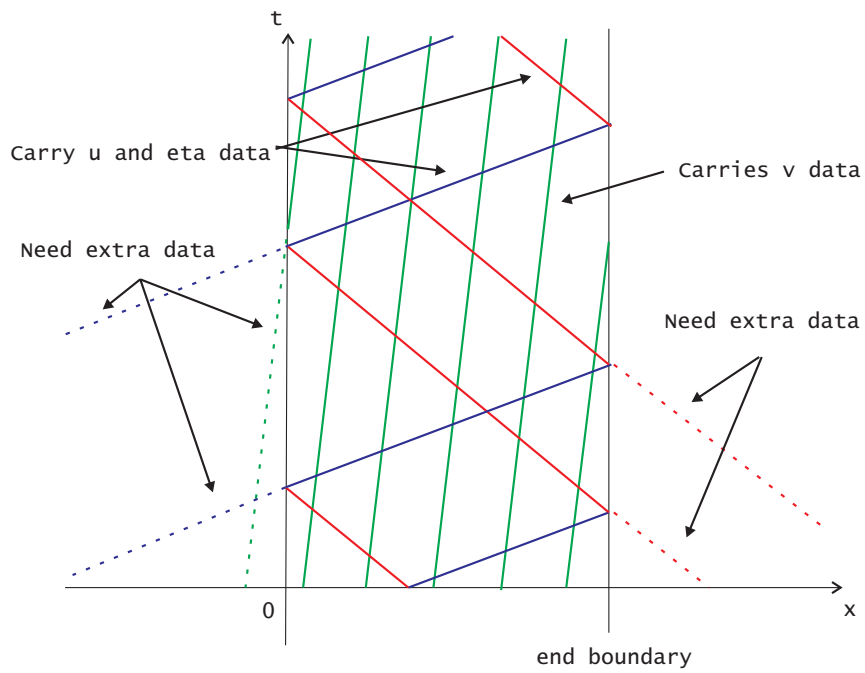
Interestingly, an upwind differentiation method would fail to correctly model these equations for this domain. For typical values, ( $u \approx 2, (\eta + H) \approx 30, g \approx 10$ ) equation (4.4) will have two characteristic curves, one being positive and the other being negative. It can be seen that information propagates forwards *and backwards* in space for the shallow water equations. An upwind method does not take into account the effects of information propagating in the direction opposite to the flow. Thus it is necessary to find a scheme that considers spatial derivatives in both directions, i.e. a central differencing scheme.

The required boundary conditions can be demonstrated by illustrating the characteristics. Figure 4.2 shows the characteristics in the  $(x, t)$  plain. The red and blue lines are the characteristics that carry the data for  $u$  and  $\eta$  and the green line is the characteristic that carries the data for  $v$ . Note that the gradients of the lines are the inverse of the value of each characteristic. As the red and blue lines enter the region, travelling in opposite directions, both the entry and exit boundaries are required to give a value for either  $u$  or  $\eta$ , but not both. Also, see that the green perpendicular

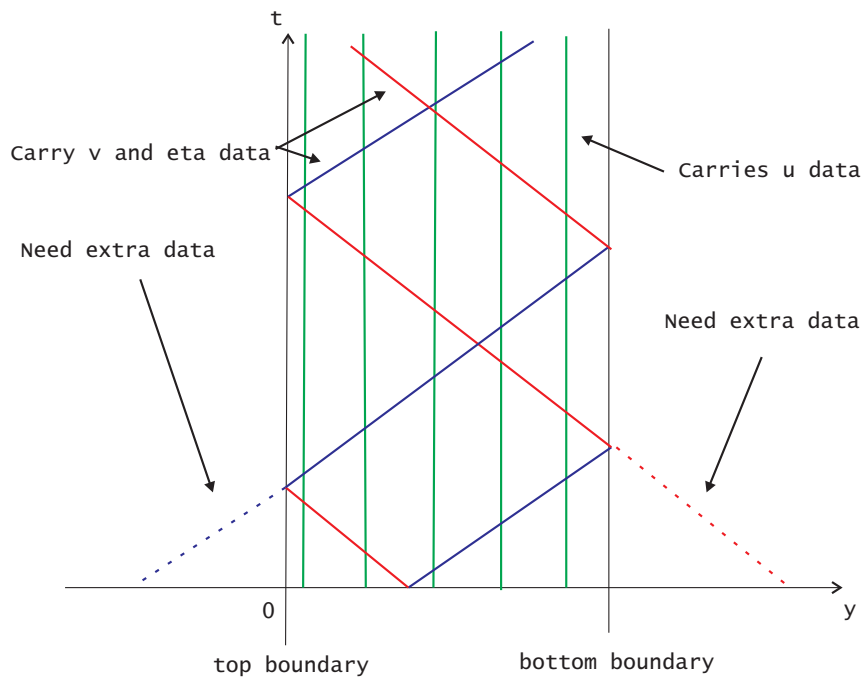
velocity line will always enter the region with positive direction. Thus data for  $v$  at  $x = 0$ , the entry boundary, is required.

Finally, consider the boundaries for the  $y$ -direction. These will be very similar to that of the  $x$ -direction but note that setting  $v = 0$  at the beginning  $x$ -boundary,  $x = 0$ , has a very positive effect. Figure 4.3 shows that there is no need to specify data for  $u$  on the  $y$ -boundaries as the characteristics don't intercept them. Only some data for  $v$  or  $\eta$  is required on the boundaries.

As is seen above, classification of a numerical method and analysis of its characteristics are crucial to applying the correct boundary conditions and can even (as shown above) indicate which differentiation schemes will be applicable. Importantly, in extreme cases it is quite possible that these classifications can change dynamically. For instance, in some transonic flows the classification can change as the flow transforms from sub- to super-sonic. This can cause difficulties in developing boundary conditions for such a problem. Transience will also change the characteristic normals of a set of PDEs. Thus it is important to always study the characteristics of the equations for the range of values they will be applied to.



**Figure 4.2:** Characteristics of 1-d Shallow Water Equations in  $x$ .



**Figure 4.3:** Characteristics of 1-d Shallow Water Equations in  $y$ .

## 5.1 Introduction

Without formal testing, the results of numerical methods are not credible. This is an important concept to grasp when modern post-processing can turn the most fallacious results into highly convincing visual simulations. Therefore, before any faith should be placed on the “pretty picture physics” of a numerical method, the user of the method must demonstrate that the model is producing results in agreement with the approximation it was based upon. This step has come to be known as *verification*. Note this is not testing if the model represents the physics it was designed to simulate. This process examines the models ability to reproduce the mathematics that it was designed to solve. The process of comparison to ‘real world’ results is known as *validation*. This examines if the approximation was chosen wisely.

The intermediate step between verification and validation is *numerical error estimation*. This process examines the error bounds of the numerical model for a simulation without an analytical result, using a particular grid resolution for a calculation that is to be compared to experimental benchmark data.

## 5.2 Verification

Verification is a critical process in the production and testing of a numerical model. A number of conclusions can be reached from the process of code verification. Transformations of the approximation equations, the *order* of the discretisation and the encoding of the discretisation and numerical method will be confirmed by the accumulation of evidence of code verification [17, 14]. One method to demonstrate verification is through a grid refinement study. However, it may be ‘over the top’ to consider a refinement study in the very early stages of the development of a model. Other *consistency checks* are worth considering first, such as checking continuity of mass or velocities in basic simulations. Once a developer is confident that the more obvious bugs have been eradicated then a formal grid refinement to verify the model will increase the evidence that the method is performing as expected. Note, should the numerical method be time stepping



then both a spatial and temporal study should be considered. As example, a spatial grid refinement study is presented.

In order to carry out a grid refinement study, a few prerequisites are required. As mentioned earlier, obvious consistency checks will avoid elementary errors being reproduced. Furthermore, assurance of *iterative convergence* is critical. Iterative convergence relates to schemes that iterate to a solution by reducing residuals. If these residuals are not within a confined limit, then the verification process will not be reliable. If iterative convergence is assured (or unnecessary) then to determine the order of grid convergence, a minimum of three simulations must be performed.

For comparison, an analytical solution or benchmark is required. It is important to contemplate the quality of an analytical solution before its application to code verification. It may be that one or many ‘real life’ analytical solutions exist for a set of equations. Difficulty arises if the chosen analytical solution does not exercise *all the terms* of the equations being used. Subsequently, the results of a grid refinement study will only verify the terms which are exercised by the analytic solution. Where no ‘real world’ analytic solution can be found to test all of terms the the equations, it can be possible to generate an ‘unrealistic’ manufactured solution that will achieve this. This process is known as *the method of manufactured solutions* and is introduced in Roache [17]

The method of manufactured solutions is elegant and powerful. The basic premise is to design a continuum solution that is analytic but still complicated enough to exercise all the terms of the equations being tested. Roache gives the hyperbolic tangent,  $\tanh$ , as an example. This solution is then entered into the governing equations to generate a source term known as  $Q(x, y, z; t)$  that will produce this outcome when included in the numerical model. The production of  $Q$  may not be trivial and Roache describes using code generated by symbolic computation to describe it<sup>1</sup>. Once  $Q$  is included in the numerical model, an analytical solution is provided that will verify *all of the equations* discretised.

### 5.2.1 Verifying the Order of Convergence

Once an appropriate analytical solution is selected, the error between the numeric and analytical results are analysed over successively finer grids. The error between the two solutions,  $E$ , is given by

$$E = f(h) - f_{\text{exact}} = C_E h^P + \text{Higher Order Terms} \quad (5.1)$$

where  $f$  is the approximate function value for a specific grid spacing,  $h$ ;  $f_{\text{exact}}$  is the exact solution to the function;  $C_E$  is a coefficient and  $P$  is the order of convergence. It follows that

$$C_E = \frac{E}{h^P}. \quad (5.2)$$

---

1. This process may inherently add another source of error from the symbolic computation that is not discussed in Roache.

If  $C_E$  remains constant over successively finer grids then the verification of code has been successful. Crucially, at this stage, there is insufficient information to solve (5.2) as the order of convergence,  $P$ , is not yet known. Formally, this is the theoretical order of accuracy for the discretisation at each grid point. Thus, for a second order method,  $P$  would be two. However, it is not always the case that the *observed order of convergence* matches the *theoretical order of convergence*. It may, therefore, be necessary to determine the observed order of convergence prior to the verification test. The order of convergence can then be determined for a fixed grid refinement ratio,  $r$  by

$$P = \ln \left( \frac{f_3 - f_2}{f_2 - f_1} \right) / \ln(r). \quad (5.3)$$

Routinely, each successive grid refinement has double the resolution of the last. Some other ratio may be chosen, but care must be taken to ensure that the ratio remains the same for each refinement. Also, the ratio must be high enough to differentiate the discretisation error from other computational errors. When results from simulations using the three grids have been calculated an order of convergence can be determined.

It is unlikely the numerical order of convergence will be precisely the same as the theoretical order of convergence. It is more likely that, due to other numerical factors and computational errors, it will be less expected. Some of this error may be due to problems categorising refinement on non-structured grids. It has even been reported that some commercial models produce orders of convergence better than analytical.

At early stages of grid refinement, using very coarse grids, the order of convergence is effected by non-discretisation errors. If the model is operating properly then as the grid is refined  $P$  will enter the asymptotic range of convergence, defined by [1] as:

The asymptotic range of convergence is obtained when the grid spacing is such that the various grid spacings  $h$  and errors  $E$  result in the constancy of  $C_E$ ,

$$C_E = E/h^P \quad (5.4)$$

Evidently, equations (5.4) and (5.2) are identical. Thus, achieving the asymptotic range is an equivalent code verification test.

The purpose of the above analysis is to rigorously demonstrate that the discretisation of a continuous equation has been implemented successfully. It may not be possible to apply the analysis to all forms of numerical method, but in any case, it must be demonstrated rigorously that any model developed is, as stated earlier, “solving the equations right”. Without assurance of this, any results from a model can not be regarded as accurate. It is possible to use other QA methods, such as static testing, to gather evidence of the verification of a model. Examples of such tests are described in Oberkampf *et al.* [14].

Once satisfied that a model has been verified for all “options” then the process of ‘verification of calculation’ allows us to estimate the level of error for a simulation at a particular grid resolution.

### 5.3 Numerical Error Estimation

Once verification of code has been completed, the next step is to assess the error for a calculation, in particular, one that hasn't an analytical solution. This is an important step for validation and, later, prediction, as it provides a bound on the numerical errors in relation to other sources.

#### 5.3.1 Richardson's Extrapolation

A technique for reducing the error of a numerical calculation is known as *Richardson's extrapolation*. It is used to provide an estimate to the continuum (fully spatially converged) solution,  $f_{\text{exact}}$ , using two grids within the asymptotic range. It is assumed that the solution,  $f$ , can be expanded in a Maclaurin series based on the grid spacing,  $h$ , such that

$$f = f_{\text{exact}} + j_1 h + j_2 h^2 + j_3 h^3 + \dots \quad (5.5)$$

where the functions  $j_1, j_2, \dots$  are independent of the grid spacing. In the classic example using a second order numerical method following Roache [17],  $j_1$  is zero and hence two solutions  $f_1$  and  $f_2$ , possessing grid spacings  $h_1$  and  $h_2$  respectively, can be combined to produce the equation for the continuum solution, discarding the higher order terms:

$$f_{\text{exact}} = f_1 + \frac{f_1 - f_2}{r^2 - 1}. \quad (5.6)$$

The estimate to  $f_{\text{exact}}$  in equation (5.6) is fourth order accurate with exclusive use of 2nd order central differencing. Otherwise, the persistence of third order terms in the expansions obligate a third order estimate.

The above extrapolation provides a (minimum) third order accurate estimate to the converged solution using just two second order grids. Whereas, encoding a third order method would incur a number of challenges. Beneficially, the extrapolation can be applied as a postprocessor, requiring no reference to the production code or grids. It is also applicable to derived parameters, again providing higher order estimates to their true value.

Conversely, there are a number of difficulties associated with this application of Richardson's extrapolation. Assumptions of grids being smooth and within the asymptotic range can be difficult to ensure, especially when postprocessing. Increasingly problematic, Richardson extrapolation has been shown to be non-conservative. Therefore, it is not recommended that Richardson extrapolation is used for the purpose of 'refining' a result.

It is more expedient to use Richardson's extrapolation to estimate the accuracy of a calculated numerical solution. The general form of Richardson's extrapolation for the limit of grid refinement of a  $P$ -th order method is given by

$$f_{\text{exact}} = f_1 + \frac{f_1 - f_2}{r^P - 1}. \quad (5.7)$$

In general, the results of equation (5.7) are order  $(P + 1)$  accurate although should the next odd derivative in series (5.5) be zero (such as in a central differencing scheme) then the scheme will be order  $(P + 2)$  accurate. Considering the second term of equation (5.7) to be an error estimator to the fine grid solution,  $f_1$ , an application of the binomial expansion produces:

$$A_1 = E_1 + \mathbf{O}(h^{P+1}, E_1^2). \quad (5.8)$$

$A_1$  is known as the *actual fractional error* and is defined by

$$A_1 = \frac{f_1 - f_{\text{exact}}}{f_{\text{exact}}}. \quad (5.9)$$

$E_1$  is the *estimated fractional error* and is defined as

$$E_1 = \frac{\varepsilon}{r^P - 1} \quad (5.10)$$

where  $\varepsilon$  is the relative error,

$$\varepsilon = \frac{f_2 - f_1}{f_1}. \quad (5.11)$$

The significance of equation (5.8) is that  $E_1$  provides an *ordered estimate* to the actual fractional error,  $A_1$ , for the fine grid solution. This is best illustrated in contrast to  $\varepsilon$ , defined in equation (5.11), that considers neither the refinement ratio,  $r$  or the order of convergence,  $P$ . Hence, the value of  $\varepsilon$  as an estimate for discretisation error is highly suspect.

### 5.3.2 Grid Convergence Index

In an effort to promote consistent reporting of the quantification of errors, Roache [17] developed, from the above analysis, a criteria known as the *Grid Convergence Index* (GCI). The GCI relates a grid refinement study to the expected results using grid doubling for a second order method. The GCI provides an error band indicating the divergence of a solution from the asymptotic value. A small GCI therefore indicates that the numerical solution is within the asymptotic range.

The GCI for a fine grid solution is

$$\text{GCI}_{\text{fine}} = \frac{F_s |\varepsilon|}{(r^P - 1)} \quad (5.12)$$

where  $\varepsilon$  is defined by equation (5.11).  $F_s$  is known as the *factor of safety*. The factor of safety acts as a probabilistic correction dependant on the needs of the user. For instance, when only two grids are available then  $F_s$  is set to 3.0. When a third grid can be examined, there is more confidence and  $F_s$  can be reduced to 1.25.

The GCI not only provides a useful metric for the error bounds for a numerical calculation, it can

also provide a required grid resolution. Assuming the grids are within the asymptotic range,

$$r_{\text{required}} = \left( \frac{\text{GCI}_{\text{required}}}{\text{GCI}_{23}} \right) \quad (5.13)$$

where ‘required’ indicates the desired accuracy and resolution and  $\text{GCI}_{23}$  indicates the GCI between a second and third grid.

The GCI can be used to provide an error bound from a combination of independent constituents. For example, considering the function  $f(x, y, z; t)$  then the GCI for  $f$  is

$$\text{GCI}_f = \text{GCI}_x + \text{GCI}_y + \text{GCI}_z + \text{GCI}_t \quad (5.14)$$

It is important when applying the GCI to a grid refinement study that the order of convergence is in the asymptotic range. Also, when using an unstructured grid, there is an alternative grid refinement ratio which can be applied,

$$r_{\text{effective}} = \left( \frac{N_1}{N_2} \right)^{\left(\frac{1}{D}\right)} \quad (5.15)$$

where  $N_1$  and  $N_2$  are the number of grid points for the solutions  $f_1$  and  $f_2$  and  $D$  is the dimension of the problem.

The above consideration of unstructured grids reveals some serious difficulties when applying the GCI to unstructured or unconventional grids/methods. For example, it may be possible to improve the accuracy of an unstructured grid without altering the number of nodes simply by redistributing them to areas of greater disturbance. In this case, using the GCI to inform a required resolution may be misleading.

Given the range of disciplines and methods that are available to numerically examine problems of waves and tides it would be foolish to assume that grid refinement or the GCI error estimators will be applicable to all. It should be the case, however, that where a discretisation of equations has taken place, some study must be undertaken to estimate the numerical accuracy of the calculation undertaken. Provided this is achieved with rigour, then the means is not necessarily important, but single grid error estimation is challenging in the absence of a grid refinement study.

## 5.4 Validation

The role of validation is to test the ability of an approximation to describe the physical system under investigation. This can *never* be achieved for a numerical model as a whole, it can only be achieved for a particular calculation with which there is experimental data to compare. More formally, validation will provide the frontier of acceptability for a particular approximation of a physical phenomena [14].

The ultimate goal of validation is to provide a quantitative error estimate for the quality of a prediction using a numerical method by comparing to a strictly monitored experiment. The challenges of achieving this are extensive but not insurmountable. As with verification, there must be an evidence gathering process that builds confidence that the errors in a validation experiment are strictly controlled and understood. It is unlikely that the experience and understanding required to achieve such a goal will be evident in the first experiment, thus often a hierarchy of validation is defined in order to facilitate various stages of the confidence building process.

Validation of a numerical calculation should not be attempted without first assessing the error in the numerical simulation that is to be compared. This is the reason why verification of code and quantification of errors in calculations are so important. If no error is given for a calculation then the results reported by the model could easily take on any conceivable value making the most accurate validation experiment worthless. The reader is urged to return to sections 5.2 and 5.3 if they are unfamiliar with these concepts.

A *validation hierarchy* seeks to reduce the complication or uncouple complicated physical problems into fundamental physical phenomena, that a number of numerical methods would be capable of tackling. The tiered system allows for a number of lab based validation experiments to be undertaken, increasing the confidence level that a numerical model may be capable of tackling some of the coupled problems where the difficulty of obtaining both experimental and numerical results is increased.

The validation hierarchy can highlight a number of different experiments that can be undertaken for a particular physical problem. The process of deciding which validation cases are the most important can be made by compiling a phenomena identification and ranking table (PIRT). Given the overarching nature of these tables, discussion about them is left to chapter 6. They are mentioned here just to emphasise that a formal decision making process is critical to making the best use of sparse resources. The next stage is to design the experiment itself.

#### **5.4.1 The Validation Experiment**

The *validation experiment* is a fledgling research problem [14]. In the past experiments have been used to improve fundamental understanding, improve theoretical models or improve reliability of existing systems. A validation experiment's goal is to determine the predictive accuracy of a numerical model. In other words a validation experiment seeks to quantify the ability of an approximation within a model to reproduce physical phenomena. With this goal in mind, the needs of the model must be placed before the needs of the experiment. It is appropriate, in this case, for the numerical model to lead the design of the experiment. It can be used to define expected results, but it can also reveal sensitivities that could lead to a re-evaluation of the importance of a particular parameter or physical process that hadn't been expected before. In any case, the experiment designed for the needs of the model is likely to produce a more useful experiment

than one without such intention. To achieve this goal Oberkamp *et al.* [14] recommends a set of guidelines for validation experiments listed as follows:

**Guideline 1:** A validation experiment should be jointly designed by experimentalists, model developers, code developers, and model users working closely together throughout the program, from inception to documentation, with complete candor about the strengths and weaknesses of each approach.

**Guideline 2:** A validation experiment should be designed to capture the essential physics of interest, including all relevant physical modeling data and initial and boundary conditions required by the model.

**Guideline 3:** A validation experiment should strive to emphasize the inherent synergism between computational and experimental approaches.

**Guideline 4:** Although the experimental design should be developed cooperatively, independence must be maintained in obtaining both the computational and experimental results.

**Guideline 5:** A hierarchy of experimental measurements of increasing computational difficulty and specificity should be made, for example, from globally integrated quantities to local measurements.

**Guideline 6:** The experimental design should be constructed to analyze and estimate the components of random (precision) and bias (systematic) experimental errors.

The above guidelines emphasise validation as a collaborative effort, producing results that will be meaningful to the predictive capabilities of the numerical model. It is also important to attempt to use fundamental measurements to examine the correlation between experimental and numerical results rather than secondary information derived by a mathematical process (in the language of Part I, using field data rather than parameterised data). This reduces the uncertainty in the accuracy of the mathematics used to derive the secondary data.

---

# Prediction

---

## 6.1 Introduction

The ultimate goal of the verification and validation process is to ascertain the predictive capability of a numerical model. Even in the case where a simple trend or insight is sought, given limited validation data, credibility of the numerical results is pivotal. Over the past twenty years, work to formalise the question of predictive capability of numerical models has advanced. A recent, ‘large’ project was undertaken by the US Department of Energy’s Accelerated Strategic Computing Initiative (ASCI) that was tasked with increasing the reliability of numerical software for defence and nuclear programs [16]. The reports and papers of this project represent the cutting edge of research into informing predictive capability of numerical models. Their approach is based upon the *gathering of evidence* to estimate, quantifiably, the error and uncertainty inherent with a calculation used in a predictive mode. Verification and validation, as discussed above, are an integral part of the evidence gathering process.

Numerical and experimental errors have been discussed in the preceding chapter. A further key concept in assessing predictive capability is *uncertainty*. Uncertainty can be subdivided into two groups:

**Aleatory Uncertainty** is the random uncertainty inherent in any physical system. The variability of this uncertainty is understood and quantifiable. It is also known as irreducible uncertainty.

**Epistemic Uncertainty** is uncertainty from a lack of knowledge. This can stem from insufficient knowledge of boundary conditions, for instance. It is also known as reducible uncertainty.

Both aleatory and epistemic uncertainty should be represented by statistical processes. Unfortunately, it is much more difficult to produce a probability distribution for an epistemic uncertainty such as an unknown boundary condition. For perfect predictability epistemic uncertainty must be minimised and this is achieved through the validation process. If epistemic uncertainty can be redefined as acknowledged error or aleatory uncertainty then predictive capability can be obtained in a more straightforward manner. Unfortunately, wave and tidal research is dominated by epistemic uncertainty. Numerical and quantifiable experimental errors, known as *acknowledged errors* must be considered together with aleatory and epistemic uncertainty. Even acknowledged errors, however, may have some epistemic uncertainty inherent in the methods with which these errors



were estimated or recorded.

The validation process involves evidence gathering to reduce the quantity of the epistemic uncertainty in a numerical model. The next two sections describe methods for structuring and reporting this evidence. However, in the case where only sparse data is available, such as in ocean modelling, validation is more challenging. Thus, the calibration process is discussed, with respect to current research being undertaken to reduce uncertainty, in section 6.4.

## 6.2 Phenomena Identification and Ranking Tables

The aforementioned evidence gathering process requires a strategic framework. To formalise and facilitate this strategy, the ASCI program recommended the use of a Phenomena Identification and Ranking Table (PIRT). As the name implies, the table is used to disassemble a physical system into the baseline phenomena and then rank their importance. Once this ranking has taken place, the current state of numerical models and validation experiments, to illustrate these phenomena, are identified. With the PIRT as a guideline, resources to improve upon the status quo can be prioritised effectively. The PIRT is similar to the validation hierarchy described in section 5.4 except, now, priority is given to the most important phenomena. The validation hierarchy remains a useful aid to the production of a PIRT. In fact, the process of producing a PIRT is as important a record as the table itself.

The production of a PIRT is an opportunity (possibly a necessity) to consult all levels of an investigative team. Numerical modellers, experimentalists, theorists and managers will shape the choices and importance of phenomena identified for a particular physical system. The PIRT is an living document, being revised in the wake of new insight discovered through the verification and validation process. Figure 6.1 shows the PIRT formation process represented as a flowchart in Pilch *et al.* [15]. This is an example of how the process for formal development of a PIRT should be undertaken.

Figure 6.1 uses the term ‘driver’ to describe the physical system chosen to be modelled. It is important that this driver not be too expansive. If, after investigation, it is found that the ranking of each phenomena is indistinguishable, it becomes impossible to prioritise a development, verification and validation strategy. In this situation, it may be indicative of an over reaching project goal. Nonetheless, identification of this problem at an early stage of a project through the PIRT will be beneficial as the project goals can be redefined.

Once a PIRT is drafted it will provide a snapshot of the requirements to ensure predictive capability of the numerical model(s) simulating the physical system. This includes a ranking of the phenomena characterising the system and the state of verification and validation of related model(s). An example of a PIRT (called a I/U map in this case) is given in Bayarri *et al.* [5] and is reproduced as figure 6.2.

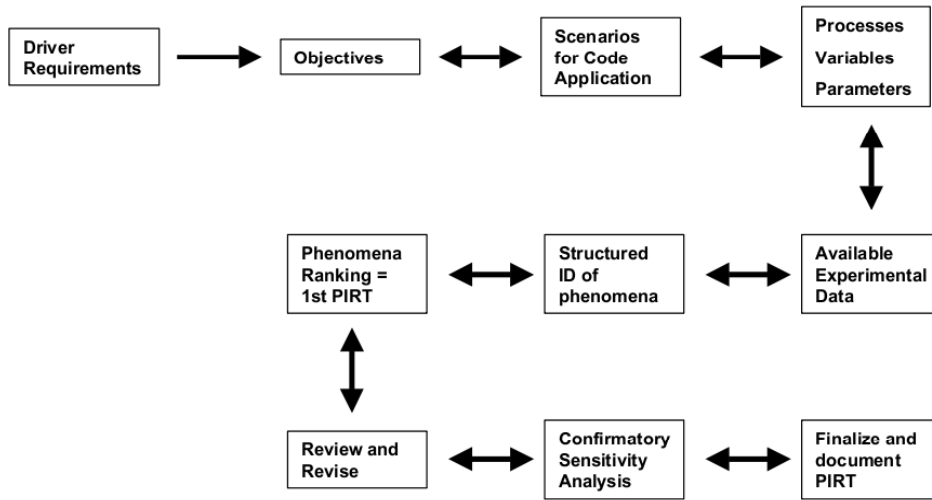


Figure 6.1: PIRT production process from Pilch *et al.* [15]

Input		Impact	Uncertainty	Current status
Geometry	Electrode symmetry-2d	3	Unspecified	Fixed
	Cooling channel Gauge	1 Unclear	Unspecified Unspecified	Fixed 1, 2 mm
Materials		Unclear	Aluminum (2 types ×2 surfaces)	Fixed
Stress/ strain	Piecewise linear	4	Unspecified (worse at high T)	Fixed
	$C_0, C_1, \sigma_s$	3	Unspecified	Fixed
Contact resistance	$1/\sigma = u \cdot f; f$ fixed	3	Unspecified	Fixed by modeler
	$u = 0$ for electrode/sheet $u =$ tuning for faying	5	Unspecified $u \in [0.8, 8.0]$	Tuned to data for 1 metal
Thermal conductivity $\kappa$		2	Unspecified	Fixed
Current		5	No uncertainty	Controllable
Load		5	No uncertainty	Controllable
Mass density ( $\rho$ )		1	Unspecified	Fixed
Specific heat ( $c$ )		1	Unspecified	Fixed
Numerical parameters	Mesh	1	Unspecified	Convergence/speed compromise
	M/E coupling time	1	Unspecified	
	Boundary conditions	1	Unspecified	Fixed
	Initial conditions	1	Unspecified	Fixed

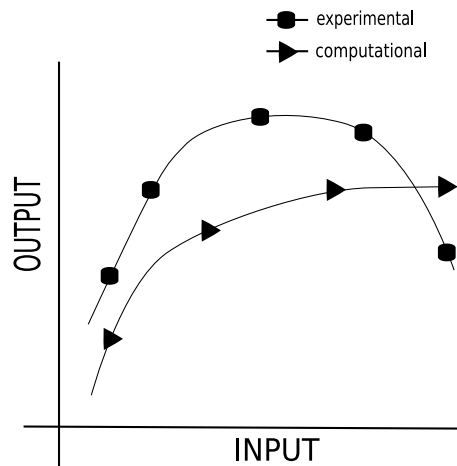
Figure 6.2: Example PIRT from Bayarri *et al.* [5]

An element which a PIRT should also identify that is not represented in figure 6.2 is the current state of validation metrics for the identified phenomena. The quality of the validation metrics will directly influence the predictive capability of a model.

### 6.3 Validation Metrics

Once a validation experiment is complete, the metrics used to quantify the comparison between the numerical and experimental results are of critical importance. This provides quantitative evidence for informing the predictive capability of a numerical model. A validation metric should seek to quantify as much of the error uncertainty as possible, for numerical and experimental results.

The development of validation metrics in computational science has mirrored the growing concern with predictive capability. The most basic metric is a simple qualitative comparison. Not unhelpful in some respects, but the resolution and accuracy of such a comparison, when no care or attention is given to either numerical or experimental error, is questionable. The next stage is a quantitative comparison, but again, if no attention is paid to the error inherent in the comparison then the comparison has little value. An example of this sort of metric is given in figure 6.3<sup>1</sup>.

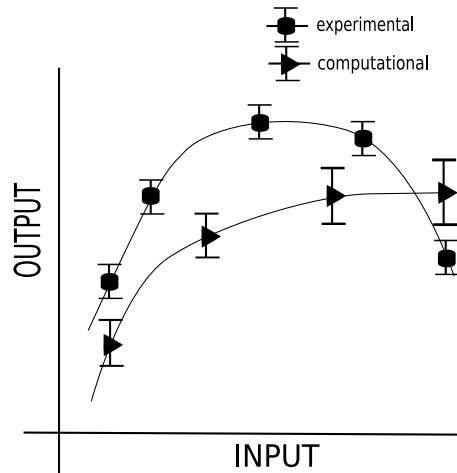


**Figure 6.3:** Quantitative comparison without errors or uncertainty. Adapted from Oberkamp *et al.* [14]

The next stage (and probably representative of the status quo in computational science) is to consider the numerical and quantifiable experimental errors, described in quantification of error and validation sections (sections 5.3 and 5.4 respectively). This metric has strong grounding in the discussion of chapter 5. The description of these errors demonstrate care in the production of both numerical and experimental results and, as discussed earlier, are an important step to defining predictive capability. However, although this form of metric demonstrates that understanding of

1. The example figures in this section are recreations of similar figures in Oberkamp *et al.* [14]

acknowledged errors in a numerical or experimental test, the metric does not account for either aleatory or epistemic uncertainty. An example of such a metric is given by figure 6.4.



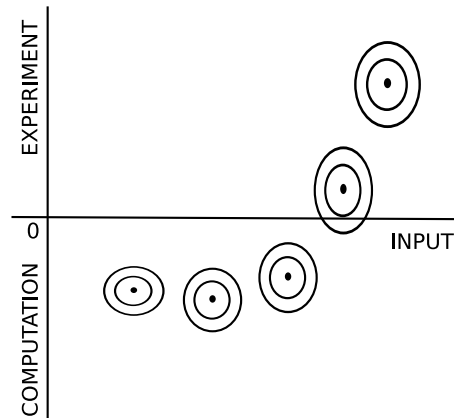
**Figure 6.4:** Quantitative comparison including acknowledged errors. Adapted from Oberkampf *et al.* [14]

Aleatory uncertainty can be modelled using statistical distributions for parameters where the variations are understood, i.e. turbulence. The epistemic uncertainty in a calculation or experiment results from those errors or uncertainties that for which there is no knowledge. This may be due to an unmeasured boundary condition or an unpredicted reflection in a wave tank or a coding error. The ability to estimate these uncertainties is the final building block in providing predictive capability. The process of undertaking the estimation of these uncertainties is invariably statistical, and likely laborious. For instance, repeated experiments in different experimental facilities will quantify some epistemic error in validation experiments. For numerical simulations, statistical analysis of uncertain parameters over ranges of values will quantify the effects of uncertain or random inputs upon a simulation.

When all of these contributors are considered it is possible to create a direct quantitative comparison using convolutions of probability distributions. This is represented in figure 6.5. The mean error for each input parameter is represented by a point and the circles surrounding each point are standard deviations of results away from this mean.

Finally, Oberkampf *et al.* [14] provides a guideline for the properties which a validation metric should ideally represent, thus:

1. A metric should incorporate an estimate of the numerical error in the computational simulation. This estimate may only be an estimate from solutions on under-resolved grids. However, we believe that representing the uncertain estimate of error as a probability, e.g., 1 out of 100 chance, is not defensible, since convergence of numerical error is more closely analogous to bias errors in experimental measurements.



**Figure 6.5:** Direct statistical comparison incorporating error and uncertainty. Adapted from Oberkampf *et al.* [14]

2. A metric should not exclude any modeling assumptions or approximations used in the computation of the simulation result. A metric should reflect all uncertainties and errors incurred in the modeling process.
3. A metric should incorporate estimates of the random errors in the experimental data that are the basis of comparison. In addition, we believe a metric should also include an estimate of the correlated bias errors in the experimental data, if possible.
4. A metric should depend on the number of experimental replications of a given measurement quantity. That is, a metric should reflect the level of confidence in the experimental mean that has been estimated, not just the variance or scatter in the data.
5. A metric should be able to incorporate computational uncertainty that is due to both random uncertainty in experimental parameters required for defining the calculations and any uncertainty that is due to lack of experimental measurement of needed computational quantities. Thus, a metric should use non-deterministic methods to propagate uncertainty through the subject computational model.

The challenge of quantifying epistemic uncertainty in a validation metric is one of great importance. However, it is also extremely challenging for numerical models that must operate with a vast amount of uncertain information. This is the case for climate modelling, ocean modelling and economic modelling. Reduction of uncertainty in these fields is normally attempted through calibration of uncertain parameters to produce a correlation with some known set of results. Achieving better predictive capability with models that require calibration is of great interest to researchers in the risk quantification field and is discussed in the next section.

## 6.4 Calibration Under Uncertainty

Calibration is distinct from validation. Validation provides a test to determine the ability of a model to represent a physical phenomena while calibration seeks to align a numerical model to known experimental data through altering uncertain input parameters. This, then, is therefore not a test of the applicability of an approximation to a physical system.

A calibration must estimate the values of unknown parameters to predict a set of known results from benchmark data. The problem inherent with this process is a lack of discussion of error and, as importantly, aleatory and epistemic uncertainty. With this in mind, the predictive capability of a model may be compromised. For example, where a set of parameters are found that allow a model to match some sparse data set (say through linear regression), it does not follow that away from those data points the choice of parameters are necessarily the correct ones. Similarly, without formal quantification of numerical error this process can also mask coding errors and unresolved grids.

There is a potential conflict in principles between the verification and validation community and the numerical modellers that are challenged with sparse data sets, such as climate, ocean and economic models. In fact, the focus of error and uncertainty quantification is beginning to be used by researchers in the aforementioned fields to actively improve the outcomes of their models, both in terms of unresolved grids [22] and uncertain input parameters. This process is sometimes known as inverse modelling [7].

The change to the calibration process to incorporate error and uncertainty can be known as *calibration under uncertainty* [21]. Commonly, regression schemes used in calibration consider uncertainty in the data calibrated against, but not error and uncertainty in the model itself. To incorporate the effects of model uncertainty in the calibration process, Bayesian statistics can be called upon. Bayesian statistics seeks to use evidence accumulated to better inform the probability distributions of uncertain parameters used in the calibration. This integrally requires that uncertainties are considered in a probabilistic sense. Once such implementation highlighted in Trucano *et al.* [21] is the *Kennedy and O'Hagan formulation*.

The difficulties with calibration under uncertainty arise from the requirement for greater numbers of simulations. With the inclusion of uncertainty in a calibration process, comes the need to represent uncertain inputs with statistical samples to examine the impact of these uncertainties. In fields where simulation times are often the limiting factor, this may preclude the development of this analysis in the short term. However, if models are to be produced that must be truly predictive in the face of sparse data sets, realisation and reduction of error and uncertainty must be taken into account. It seems likely therefore, that the expense of implementing the ideas above may be more economic than accurate gathering of the complete data fields for simulations on planetary or regional scales. As mentioned earlier, however, this remains a cutting edge and contentious topic.

---

## Challenges and Further Work

---

The purpose of this chapter is to highlight opportunities for further work that can be undertaken within the marine energy research community in order to increase the credibility and, hence, the predictive capability of the numerical models developed and tested.

**Strategic Planning:** Significant interest in technology roadmaps for marine energy and has produced a funding stream for the continued development of this technique. The PIRT table, discussed in section 6.2, provides a similar strategic overview for numerical modelling of wave and tidal phenomena. It defines the necessary model development and validation studies that should be carried out to improve the modelling capabilities of the project ensuring efficient allocation of resources. The application of such a methodology in the numerical model development could be extremely beneficial.

**Estimation of Error in Numerical Models:** With such importance being placed on the estimation of error from the use of discretisation in numerical models, error estimation must be taken into account at the design stage or when choosing an existing model. The ability to estimate numerical error in existing models must also be given serious consideration.

**Validation Experiments:** There is a unique opportunity to start the design of formal validation experiments as described in section 5.4. The years of experimental experience of wave device testing within the marine energy field facilitates an opportunity to demonstrate the capability to produce numerical model led experiments.

**Experimental Scaling for Free-Surface/Current Interaction:** It remains unclear whether a formal or informal understanding of the effects of scaling for validation experiments of free-surface/current interaction exists. This is an opportunity to definitively answer the questions surrounding the effects of scale on validation experiments where the design subject involves both free surface and current interaction.

Research into the four areas of interest highlighted above would produce a definitive increase in the credibility of the numerical modelling partaken in the SuperGen consortium and within the marine energy community as a whole. The challenges are far reaching, but not insurmountable, and if marine energy requires, for economic reasons or otherwise, a numerical model led design process, then some of the ideas presented here must surely be addressed. Overall, it is hoped that this document will aid those within the wave and tidal research community to take the first

steps in producing numerical models which the wider marine energy community can apply with confidence in the credibility of the results.



---

# Glossary

---

The following glossary contains technical terms that are used in the numerical modelling for wave and tidal research. They have been selected from the current document and that of McCabe [12]. The source of information for the descriptions is collected from numerous sources including personal experience, the internet and [www.wikiwaves.org](http://www.wikiwaves.org).

## **Advection**

The transportation of a conserved quantity by a moving fluid.

## **Algorithmic Efficiency**

The expense of a particular numerical technique in computing time and memory space. It is possible to predict the number of operations numerical methods may require and thus compare theoretical speeds. However, this does not account for the facilities on which the methods are being computed nor does it account for efficient programming of the methods.

## **Amplitudes of Motion**

The amount by which a body may move under a given force or moment over a specific period of time.

## **Angle of Attack**

Angle between the chord line of an aerofoil and the relative oncoming flow measured from the trailing edge of an aerofoil.

## **Body Force**

A force applied to the whole volume of fluid within a domain.

## **Boundary Condition**

A boundary condition is a mathematical formula that expresses a known physical restriction at a certain spatial coordinate. For instance, an appropriate boundary condition for a wall would be that no flow may pass through it.

## **Boundary Element Method (Panel Method)**

A simulation using potential flow that allows a reduction in spatial dimension by calculating the results solely on the surfaces or lines bounding the volume or area.

## **Boundary Layer**

The thin layer near a boundary where a sharp reduction in velocity can be observed due to the viscous effects of the boundary.

## **Boundary Layer Approximation**

The division of a high Reynolds number flow into an inviscid far field region and a viscous boundary layer region near to walls.

## **Boussinesq Approximation**

Approximation to the vertical component of the Euler equations for water waves that are

weakly non-linear and have fairly long wavelength.

**Cell**

A cell is the region defined within the boundaries of the mesh.

**Central Differencing**

Approximation to a differential operator using points on either side of the point to be differentiated.

**Chord Line**

Straight line between the leading edge (forward most part) and trailing edge (backwards most part) of an aerofoil. It's length is often chosen to be one for convenience.

**Compressibility**

The change of volume of an amount of fluid in response to pressure.

**Continuity Equation**

An equation that describes the transportation of conserved quantity. In fluid dynamics the continuity equation is often used to describe conservation of mass.

**Credibility**

The confidence with which a numerical model can be applied to predict the physics it was designed to simulate.

**Database**

A structured collection of records stored within a computer system.

**Degrees of Freedom**

Descriptions of the set of independent displacements or rotations that completely specify the displaced or deformed position or orientation of a system. In three dimensions the six degrees of freedom are often described with a nautical reference:

1. **Heave:** Moving up and down.
2. **Surge:** Moving back and forward.
3. **Sway:** Moving port to starboard (left to right).
4. **Pitch:** Rotation about the short horizontal axis as if the bow and stern were rising and falling.
5. **Yaw:** Rotation about the vertical axis as if the ship were changing bearing.
6. **Roll:** Rotation about the long horizontal axis as if the ship were listing from side to side.

**Density**

The amount of mass in a given volume.

**Detached Eddy Simulation (DES)**

A RANS simulation that employs a LES solver in fine regions of the grid. This is a method of circumventing the problem of averaging out short time scale non-turbulent motions using RANS.

**Diffraction**

The alteration to the form of a wave pattern from the presence of an obstacle in the path of the wave. Diffraction can be observed at point breaks where the wave fronts will bend around the point.

**Diffusion**

The movement of regions of high concentration in a fluid to regions of low concentration.

**Direct Numerical Simulation (DNS)**

Numerical simulation of the Navier-Stokes equations without the use of a turbulence model.

**Discretisation**

The process by which continuous regions or fields are divided into discrete points, panels or volumes. This facilitates the definition of a linear system which can then be inverted to produce a solution.

**Domain**

The region is space, time or any other parameter that a mathematical function is defined for. Thus, tests may be conducted over a series of frequencies and the bounds of those frequencies define the domain. For field solutions the bounds of space and time in which the field is defined represent the domain.

**Dynamic Boundary Condition**

Equation to enforce a constant pressure on a surface over time.

**Energy Equation**

Equation that enforces conservation of energy. This equation is rarely used in incompressible fluids but may be of interest where heat transfer is occurring.

**Euler Equations**

The Euler equations represent the equations of motion for inviscid flow.

**Field**

A region in which a physical quantity can be defined for each point in space and time.

**Force**

A quantity that denotes the ability to push, pull, twist or accelerate a body.

**Free Surface**

A surface at which the kinematic and dynamic boundary conditions are enforced. Often, such a surface is used to describe the interface between the sea and the atmosphere.

**Frequency Domain**

The change in a prescribed quantity with respect to frequency rather than to time. It can be used to give the response of a wave device to different frequencies of simple harmonic waves where a steady state solution can be produced for each case.

**Grid**

A set of spatial points within a domain that are chosen to compute field solutions. Grids

may be more dense where there is a need for greater resolution of the flow and less dense where little is happening.

**Grid Resolution**

A space, such as a volume, surface or line, must be approximated in a numerical model by discretising the space into small parts. The size of these parts is known as the grid resolution. The difference between the solution on this grid and the exact (or continuum) solution is known as the discretisation error.

**Hydrodynamics / Fluid Dynamics**

The study of fluids in motion. Marine hydrodynamics is the subset of hydrodynamics in which the fluid can be assumed to be incompressible.

**Hydrostatics**

The science of fluids at rest or under stable equilibrium. Thus if a fluid has a free surface then the gravitational force is balanced by the pressure of the water below the free surface. For this to be true the water must not be undergoing any vertical acceleration and in wave theory this assumption is known as the hydrostatic assumption.

**Incompressible**

A fluid that's volume does not change in response to pressure.

**Initial Condition**

An initial condition is similar to a boundary condition except that it describes a restriction in time rather than in space. For instance, if we were to choose slack water as time zero then it would be prudent to set the velocities within the domain to zero at that time.

**Inviscid**

A fluid that has no viscosity. High Reynolds number flows define such a fluid, determining the viscous terms in the equations of motion to be small and, thus, allowing the Euler equations to be used for the fluid under consideration.

**Irrotational**

A fluid in which the total vorticity is zero.

**Kinematic Boundary Condition**

Equation to ensure that the same particles remain on a given surface over time.

**Laminar Flow**

A fluid that travels smoothly and in regular paths or in parallel layers. A laminar flow velocity profile may be more rounded than that of a turbulent flow. Laminar flow is likely to be present at Reynolds number  $< 500$ .

**Large Eddy Simulation (LES)**

Numerical simulation of the Navier-Stokes equations with a turbulence model applied only at the sub-grid scale.

**Linear Approximation (Linear Wave Theory)**

An approximation to a general function using a linear function. Such an approximation can be used for the kinematic and dynamic boundary conditions for waves of small amplitudes.

**Linear Differential Equation**

The properties of such an equation allow two solutions for differing boundary conditions to be added to give the combined solution.

**Manifold**

A space modelled on some subset of euclidean space.

**Mean Drift Forces**

The mean forces exerted by ambient waves.

**Memory Equation**

A summation of a time dependant phenomena decoupled from the object of interest. For instance, in wave body interactions the radiated waves from the body will have a cumulative effect that increases over time as they extend away from the body.

**Mesh**

A mesh is formed by connecting the points within a grid to form cells.

**Moment**

A force applied away from the centre of mass of an object, thus rotating or bending the object.

**Momentum Equations**

Equations that enforce conservation of momentum.

**Navier-Stokes Equations**

The set of momentum equations used to describe the motion of viscous fluids. Together with continuity and energy equations, they make up the complete equations to describe the motions of fluids.

**Non-dimensional**

A quantity which has no physical unit often constructed by cancelling out the units of other combined physical quantities.

**Nonlinear**

A set of equations where linear approximations have not, can not or should not be made.

**Numerical Model**

A computer model that attempts to simulate a physical phenomenon or system of phenomena.

**Order of Approximation**

Corresponds to the index of the approximation function. Thus a constant approximation would be zeroth or 'low order' and a linear approximation would be first order. Usually any approximation greater than zeroth order is considered to be a 'higher order' approximation.

**Parameter**

Used to describe quantities that are generated by manipulating elementary field data. Also known as functionals.

**Phase (Fluid)**

A fluid phase is a distinction between two immiscible fluids, most commonly air and water. However other such phases may include water and oil, for instance.

**Physical Phenomenon**

Physical phenomenon, or a physical system, is the fundamental fluid property, or collection of properties, whose dynamics are not necessarily well understood.

**Planform**

A plan view or horizontal projection of a surface. In aeronautics this would refer to the shape of the wing and fuselage of an aircraft when viewed from above.

**Potential Flow**

An inviscid irrotational flow can be represented by a velocity potential,  $\phi$  that satisfies Laplace's equation:

$$\nabla^2\phi = 0.$$

**Predictive Capability**

The quantitative level of accuracy with which a numerical model can be used to predict physical phenomena for which no experimental comparison exists.

**Radiation**

Waves generated by a body in motion immersed in a fluid with a free surface.

**Residual**

The deviation of an approximate solution from the exact solution.

**Response Amplitude Operators (RAOs)**

Statistics used to predict the motion of a body to a given set of sea states and device parameters such as mass. RAOs are commonly used in naval design.

**Restoring Force**

If a body or a system is in equilibrium, then a restoring force describes the force that attempts to return the system to equilibrium. Obviously a restoring force can represent many forces, such as the damping force, for instance.

**Reynolds Averaged Navier-Stokes Equations (RANS)**

Time averaged steady approximation to the full Navier-Stokes Equations that can be applied to compressible viscous flow of general domains. The final term, known as the Reynold's Stresses, represents the effects of turbulence.

**Shear Stress**

A force applied tangentially to a surface.

**Singular**

The value of a quantity or parameter that at given points in a field or otherwise has an

unbounded or infinite value.

**Smoothed Particle Hydrodynamics (SPH)**

Instead of considering fluid as a continuum, Smoothed Particle Hydrodynamics is concerned with calculating the movement of particles that conform to the Navier-Stokes equations.

**Steady**

A quantity or quantities that are invariant over time.

**Streamline**

Paths that are tangential to the instantaneous velocity vectors within a flow.

**SuperGen Marine**

Research consortium tasked with the reduction of risk and uncertainty for wave and tidal energy development.

**Time Domain**

Record of the change in a prescribed quantity over time. Solutions of fields must be undertaken in the time domain.

**Time Step**

A numerical simulation in the time domain must divide the time into a number of small steps. The amount of time between these steps is known as the time step. The stability of numerical simulations is often sensitive to the choice of time step.

**Transport Equation**

A general partial differential equation used to describe conservation laws.

**Turbulence**

Statistically modelled fluctuation of velocity about the mean over many length and time scales. Turbulence is brought about by the presence of viscosity and rotational motion within a fluid and is dissipated at the molecular level.

**Turbulent Flow**

A fluid that travels with chaotic motion. Pressure and velocity can vary rapidly in space and time. Vortices of many scales can be formed and their effect on hydrodynamic bodies is particularly sensitive to this scale. A turbulent velocity profile may be flatter than a laminar one.

**Uncertainty**

An uncertainty is an error that is either unknown (epistemic) or that is non-deterministic (aleatory).

**Unsteady**

A quantity or quantities that change with time.

**Unsteady Reynolds Averaged Navier-Stokes Equations (URANS)**

Similar to RANS except that the time averaging is on a scale such that the non-turbulent unsteady motions of the fluid are preserved.

**Upwind Method**

Approximation to a differential operator using the point to be differentiated and points upstream of it.

**Validation**

The testing of a numerical model to determine how well it simulates a physical phenomenon.

**Variable**

Programming artifact that can alter its value during the execution of the program.

**Velocity Potential**

A scalar quantity that's derivative returns the velocity at any point in a defined field.

**Verification**

The confidence in a numerical model representing the approximation it was designed to simulate.

**Viscosity**

The resistance of a fluid to a shear stress.

**Vorticity**

Describes the spin of one or a group of particles in a fluid.

**Wave Exciting Forces and Moments**

Forces and moments generated by the body in motion that will have the effect of generating waves on the free surface.



---

# Bibliography

---

- [1] NPARC Computational Fluid Dynamics (CFD) Verification and Validation Web Site. URL <http://www.grc.nasa.gov/WWW/wind/valid/homepage.html>.
- [2] I. H. Abbott and A. E. V. Doenhoff. *Theory of Wing Sections*. Courier Dover Publications, 1959.
- [3] *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*. AIAA, 1998.
- [4] P. Andrews, G. Cats, D. Dent, M. Gertz, and J. L. Ricard. European standards for writing and documenting exchangeable fortran 90 code. Technical report, The MetOffice, 2009. URL [http://www.metoffice.gov.uk/research/nwp/numerical/fortran90/f90\\_standards.html](http://www.metoffice.gov.uk/research/nwp/numerical/fortran90/f90_standards.html).
- [5] M. J. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu. A framework for validation of computer models. *Technometrics*, 49(2):138–154, 2007.
- [6] S.-A. Boukabara and P. V. Delst. Standards, guidelines and recommendations for writing fortran 95 code. Technical report, National Oceanic and Atmospheric Administration, 2007.
- [7] B. S. Chua and A. F. Bennett. An inverse ocean modeling system. *Ocean Modelling*, 3(3-4): 137 – 165, 2001.
- [8] R. A. Dalrymple and B. D. Rogers. Numerical modeling of water waves with the SPH method. *Coastal Engineering*, 53:141–147, 2006.
- [9] C. Hirsch. *Numerical Computation of Internal and External Flows*. Butterworth-Heinemann, 2007.
- [10] J. Katz and A. Plotkin. *Low-Speed Aerodynamics, Second Edition*. Cambridge University Press, 2001.
- [11] *Best Practice Guidelines for Marine Applications of Computational Fluid Dynamics*. MARNET-CFD.
- [12] A. P. McCabe. An appraisal of a range of fluid modelling software. Technical report, Lancaster University, 2004.
- [13] J. N. Newman. *Marine Hydrodynamics*. The MIT Press, 1977.

- 
- [14] W. L. Oberkampf, T. G. Trucano, and C. Hirsch. Verification, validation, and predictive capability in computational engineering and physics. Technical Report SAND2003-3769, Sandia National Laboratories, 2003.
- [15] M. Pilch, T. Trucano, J. Moya, G. Froehlich, A. Hodges, and D. Peercy. Guidelines for sandia asci verification and validation plans – content and format: Version 2.0. Technical Report SAND2000-3101, Sandia National Laboratories, 2001.
- [16] D. E. Post and L. G. Votta. Computational science demands a new paradigm. *Physics Today*, 58(1):35–41, 2005.
- [17] P. J. Roache. *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, 1998.
- [18] T. Sarpkaya and M. Issacson. *Mechanics of Wave Forces on Offshore Structures*. Van Nostrand Reinhold Company Inc., 1981.
- [19] H. Schlichting. *Boundary-Layer Theory*. McGraw Hill Book Company, 1960.
- [20] M. B. R. Topper. The efficient conversion of tidal energy to electrical energy. Master’s thesis, The University of Southampton, 2003.
- [21] T. Trucano, L. Swiler, T. Igusa, W. Oberkampf, and M. Pilch. Calibration, validation, and sensitivity analysis: What’s what. *Reliability Engineering & System Safety*, 91(10-11):1331 – 1357, 2006.
- [22] E. Zaron and G. Egbert. Verification studies for a z-coordinate primitive-equation model: Tidal conversion at a mid-ocean ridge. *Ocean Modelling*, 14(3-4):257 – 278, 2006.